A Multi-Module Anomaly Detection Scheme based on System Call Prediction

Zhenghua Xu [1], Xinghuo Yu [1], Yong Feng [1,2], Jiankun Hu [3], Zahir Tari [1], Fengling Han [1]

[1]*RMIT University, Melbourne, VIC 3001, Australia*
{zhenghua.xu, x.yu, zahir.tari, fengling.han}@rmit.edu.au
[2]*Harbin Institute of Technology, Harbin, 150001, China*
yong.feng.5601@gmail.com
[3]*University of New South Wales, ADFA, Canberra, ACT 2600, Australia*
J.Hu@adfa.edu.au

***Abstract*—Due to the rapid and continuous increase of network intrusion, the need of protecting our systems becomes more and more compelling. In many situations, there exists a weak anomaly signal detection problem: due to the little number of anomalous system calls, the anomalous patterns of some intrusions may not be enough to distinguish themselves from normal activities so the existing anomaly detection systems can not detect this kind of sequences accurately. Motivated by this, we propose a multi-module anomaly detection scheme to solve this problem through utilizing system call prediction to enlarge the patterns of weak anomaly signal sequences and make them more distinguishable. Besides this, a variation of the Viterbi algorithm (called VV algorithm) is developed to predict the most probable future system calls more efficiently and a Markov-based intrusion detection method is adopted for the pattern value calculation and anomaly detection. The results of our experimental study conclude the followings: (i) the proposed scheme can greatly improve the intrusion detection accuracy of this Markov-based intrusion detection method in terms of hit rates under small false alarm rate bounds; (ii) the performance of the proposed scheme depends on the prediction accuracy of the adopted prediction technique; (iii) the developed VV algorithm is exponentially more efficient than a baseline method.**

I. Introduction

In recent years, computer network has become an essential component of the modern society. There are rapidly increasing usages of computer network in many domains ranging from enterprise applications in financial, military and energy sectors to individual computer users. Therefore, the computer network quickly becomes the target of criminals and its security has become a critical problem to the society. As reported, the economic losses caused by cyber-crime are around $67 billion annually in US [1] and up to $1 trillion globally in 2009 [2].

Many intrusion detection techniques have been proposed to identify intrusions that have come into the computers and network systems and take actions to minimize the damages. According to the different kinds of analyses carried out, these intrusion detection techniques can be classified into the following two categories: *Signature Detection Systems* (or called *Misuse Detection Systems*) and *Anomaly Detection Systems* [3].

Specifically, signature detection systems [4], [5], [6] utilize pattern recognition techniques to find pre-characterized patterns of known attacks within the analyzed data. Therefore, they can detect well-known attacks effectively and efficiently but not be able to detect novel, unknown attacks. Anomaly detection systems [7], [8], [9] construct models (called *Norm Profiles*) to characterize the normal activities and any activity whose pattern deviates far away from the norm profile and exceeds a predefined threshold is treated as an anomaly (attack). Anomaly detection systems are ideal for detecting novel and unknown attacks. Not surprisingly the most common computer crimes and main sources of damages are from virus, worm or Trojan infections whose signatures can be changed easily and many new types of such attacks can be produced which can easily penetrate commercial firewall systems. Therefore, in this work, we focus on anomaly detection systems.

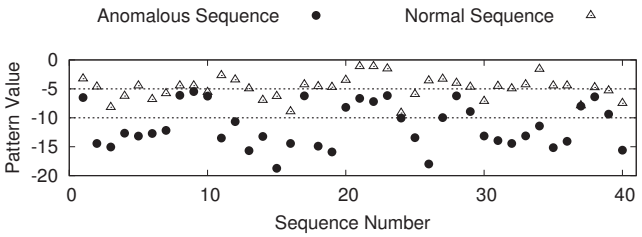

Fig. 1. Motivation Example

System Call is a program signal for requesting a service from the system's kernel. The existing work [10], [11] has demonstrated that short sequences of system call traces generated by program executions are stable and consistent during programs' normal activities so that they can be used to distinguish the abnormal operations from normal activities. Therefore, system call has been widely applied into intrusion detection work such as, [12], [13], [14], [15], [16], [17].

In many situations, due to the little number of anomalous system calls, the anomalous patterns of some intrusions may not be enough to distinguish themselves from normal activities (i.e., the pattern values of these sequences fall into the value range of normal activities). An example is shown in Fig. 1 where 40 pre-labeled anomalous short sequences and normal short sequences are processed by an intrusion detection system and their pattern values are calculated and plotted. We can observe that the pattern value range of normal sequences is $[0, -10]$ while that of anomalous sequences is $[-5, -20]$. Since the existing anomaly detection systems activate intrusion

978-1-4673-6322-8/13/$31.00

alarms whenever the deviations between given sequences and the normal activities exceed a predefined threshold, it is difficult for these systems to accurately distinguish the sequences whose pattern values fall into $[-5, -10]$. We call this kind of sequences *Weak Anomaly Signal Sequence* and this problem *Weak Anomaly Signal Detection Problem*.

Motivated by this, we propose a *Multi-module Anomaly Detection Scheme* to solve this problem and improve the intrusion detection accuracy of existing intrusion detection methods through utilizing a Markov-based prediction method to predict the most probable future system call sequences of weak anomaly signal sequences to extend these sequences, enlarge their patterns and make them more distinguishable. The most probable future sequence is predicted by finding the future sequence that matches a chosen *Prediction Model* best among all possible sequences. To improve the runtime efficiency, we further develop a **V**ariation of **V**iterbi algorithm (named *VV Algorithm*) to make the prediction of most probable future system call sequences much more efficiently, which uses a dynamic programming matrix to avoid enumerating all possible future sequences and calculating the same probability values repeatedly.

Moreover, in this proposed scheme, the intrusion detection methods adopted for the pattern value calculation and anomaly detection can be various according to different requirements. To keep it simple, we adopt a Markov-based intrusion detection method proposed by [8] to obtain the *Sequence Probability* as the pattern value. Extensive experimental studies are conducted on the benchmark *sendmail* dataset and the results conclude the followings: (i) the proposed scheme can greatly improve the intrusion detection accuracy of this Markov-based intrusion detection method in terms of hit rates under small false alarm rate bounds; (ii) the performance of the proposed scheme depends on the prediction accuracy of the adopted prediction technique; (iii) the developed VV algorithm is exponentially more efficient than a baseline method.

The rest of this paper is organized as follows. We first describe the benchmark *sendmail* dataset in Section II. Then, we present the multi-module anomaly detection scheme and the VV algorithm in detail in Section III and Section IV, respectively. Section V describes the experimental study and reports its results. We conclude the paper and discuss the future work in Section VI.

II. Dataset Description

In this paper, we use the benchmark *sendmail* system call traces collected by the Computer Science department of University of New Mexico. For detail procedures of generating and collecting these traces in this dataset, readers are refereed to [10], [11] and [14]. Briefly, normal and abnormal system call traces exist in this dataset are summarized as follows:

- Normal traces: traces of normal activities of the sendmail program include a trace of the sendmail daemon and several traces generated from invocations of the sendmail program in this dataset.
- Abnormal traces: traces generated and collected when the program runs with abnormal activities consist of two traces of the *syslog-local* intrusions, two traces of the *syslog-remote* intrusions, two trace of the *decode* intrusions, three traces of the *sunsendmailcp (sscp)* intrusions and two traces of unsuccessful intrusions (*sm5x* and *sm565a*).

There are two column data in each system call trace file, the data in the first column are the process IDs and the data in the second column are the corresponding system call code where each system call code stands for a specific system operation. For example, as shown in Table I, 2668 and 1391 in first column are process IDs while 3, 5 and 6 in second column represent system operations "open", "read" and "close", respectively. We can look up these mappings in a mapping file.

TABLE I
FRAGMENT OF SYSTEM CALL TRACE FILE

Process ID	Code
...	...
2668	5
2668	3
...	...
1391	6
...	...

III. Multi-Module Anomaly Detection Scheme

The proposed multi-module anomaly detection scheme consists of three modules: *Training Module*, *Detection Module* and *Weak Anomaly Signal Processing Module*.

Overall, the detection module online monitors the system call trace log and uses a sliding window to continuously extract the short system call sequences. For each short sequence, a pattern value is calculated. If this pattern value is higher than a predefined maximum threshold (denoted as T_{max}) or lower than a predefined minimum threshold (denoted as T_{min}), it means this sequence has a distinct normal or abnormal pattern so we can directly identify this short system call sequence as a normal activity sequence or an anomaly, respectively. Otherwise, this short sequence is a weak anomaly signal sequence and it will sent to the weak anomaly signal processing module for further processing.

Taking Fig. 1 as an example, if $T_{max} = -5$ and $T_{min} = -10$, the sequences with pattern values higher than -5 (e.g., the first normal sequence) or lower than -10 (e.g., the second anomalous sequence) are directly identified as normal activity sequences or anomalies, respectively. The sequences whose pattern values fall into $[-5, -10]$ (e.g., the first anomalous sequence) are weak anomaly signal sequences and will be sent to the weak anomaly signal processing module.

In the weak anomaly signal processing module, we enlarge the pattern of the weak anomaly signal sequences to make them more distinguishable by predicting their most probable future system call sequences based on either the normal or abnormal activity model generated in the training module. Finally, we obtain the most probable extended system call

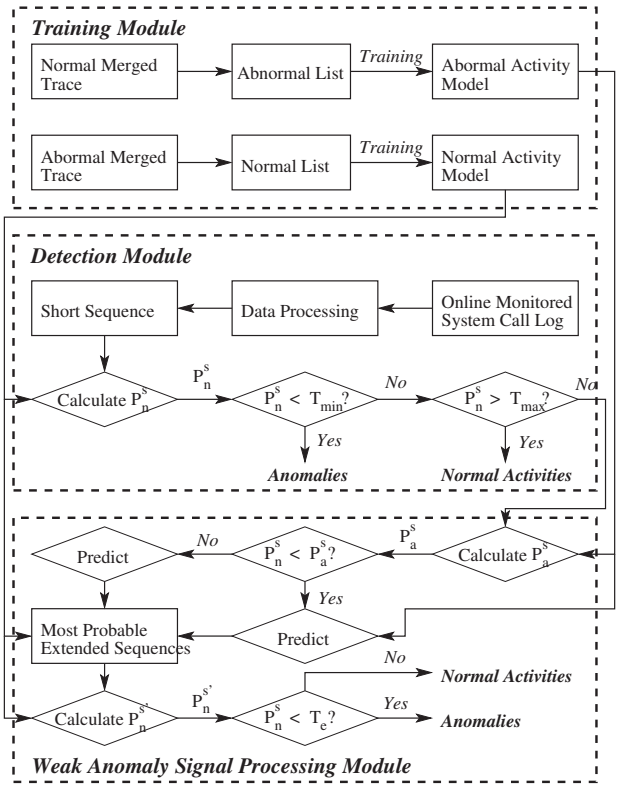

Fig. 2. Multi-Module Anomaly Detection Scheme

sequences through combining these predicted future sequences with the given weak anomaly signal sequences. To detect the anomaly signal based on these extended sequence, for each extended sequence, a new pattern value is calculated and compared with a new extended threshold.

The diagram of this multi-module anomaly detection scheme is shown in Fig. 2 and some details of each module will be presented in the following subsections.

A. Training Module

In training module, we aim to train two Markov models called *Normal Activity Model* and *Abnormal Activity Model* to characterize the normal and abnormal system behaviors by using normal and abnormal historic system calls.

A Markov Model is a model with the special Markov assumption: the probability distribution of the state at time $t+1$ depends on the state at time t, and does not depend on the previous states leading to the state at time t. Formally,

$$Pr\left(s_{t+1}=i_{t+1}|s_t=i_t\right)=Pr\left(s_{t+1}=j|s_t=i\right)=p_{i,j}, \quad (1)$$

where $p_{i,j}$ is the probability of being in a state j given its previous state is i and called *Transition Probability*. Therefore, given a state set Q, we can define a Markov model by a *Transition Probability Matrix* M and a *Initial Probability Vector* V as shown in Eq. (2) and Eq. (3), respectively.

$$M=\begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,m} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,m} \end{pmatrix} \quad (2)$$

$$V=\left[p_1,\ldots,p_i,\ldots,p_m\right], \quad (3)$$

where $p_{i,j}$ is the transition probability between states s_i and s_j, p_i is the probability of the system being in state i at time $t=0$ (called *Initial Probability*), m is the cardinality of Q.

Since the normal system call traces consist of only normal system operations but the abnormal system call traces consist of both normal and abnormal system operations, directly training the abnormal activity model by abnormal traces can not characterize the anomalies correctly. To overcome this problem, we divide the traces into a list of short sequence by the following steps: (i) We sequentially merge all pre-labeled historic normal traces in training dataset together to generate a normal merged trace and also generate an abnormal merged trace by the same way. (ii) The normal merged trace is scanned by a sliding window to create a list of unique short system call sequences, each of which is associated with a quantity value indicating how many times this short sequence appears in the normal merged trace. The resulted list is called *Normal List*. (iii) We further scan the abnormal merged trace by the same sliding window to generate an *Abnormal List*. Differently, for each resulted short system call sequence, we first look it up in the normal list; if we find a match in the normal list, we discard this sequence; otherwise, we store it in the abnormal list. Table II shows an example of normal list generated by a sliding window with window size $w=3$.

TABLE II
EXAMPLE OF NORMAL (OR ABNORMAL) LIST

Short Sequence	93, 94, 5	$\cdots$	4, 50, 27	$\cdots$
Quantity Value	357	$\cdots$	22	$\cdots$

Based on the normal and abnormal lists, we are able to train the normal and abnormal activity Markov models through treating each system call as a state s_i and the system call set as Q, the transition probability and the initial probability can be obtained by Eq. (4) and Eq. (5), respectively.

$$Pr\left(s_{t+1}=j|s_t=i\right)=p_{i,j}=\frac{N_{i,j}}{N_i}, \quad i,j\in Q, \quad (4)$$

$$Pr\left(s_0=i\right)=p_i=\frac{N_i}{N}, \quad i\in Q, \quad (5)$$

where $N_{i,j}$ is the number of system call i in the list followed by a system call j; N_i is the number of system call i in the list and N is the total number of system calls in the list.

Bayes parameter estimation can also be used to estimate the transition probabilities and initial probabilities from historic data. However, because of high computational cost, it is not adopted in this work. Given enough historic data, the estimation through Eq. (4) and Eq. (5) can be quite stable.

B. Detection Module

In the detection module, many existing intrusion detection methods [18], [19], [20] can be used to calculate various pattern values according to different requirements. To keep it simple, a Markov-based intrusion detection method proposed

by [8], which uses the *Sequence Probability* as the pattern value, is adopted in this work. The sequence probability indicates how probable this sequence occurs in the given Markov model and is defined as follows:

$$P^s(s_1, \ldots, s_l) = p_{s_1} \cdot \prod_{i=2}^{l} p_{s_{i-1}, s_i}. \quad (6)$$

where p_{s_1} is the initial probability and p_{s_{i-1}, s_i} is the transition probability.

In [8], given a normal activity Markov model, a sequence probability value (denoted as P_n^s) is calculated for each short sequence and compared with an unique threshold. If P_n^s is lower than the threshold, this sequence is identified as an anomaly; otherwise, it is a sequence of normal activities. However, this method encounters the weak anomaly signal detection problem: the sequence probability values of some anomalous sequences fall into the value range of normal sequences so using a unique threshold is hard to accurately distinguish the weak anomaly signal sequences.

In order to solve this weak anomaly signal detection problem, in our scheme, we pre-define two thresholds: the maximum threshold (denoted as T_{max}) and the minimum threshold (denoted as T_{min}). The value of T_{max} is defined so large that, if the resulted P_n^s of a short sequence is larger than T_{max}, this sequence is a sequence of normal activities with extremely high certainty. Similarly, The value of T_{min} is so small that, if the resulted P_n^s of a short sequence is smaller than T_{min}, this sequence is an anomaly with extremely high certainty. Therefore, given a short sequence, it will be directly identified as a normal activity sequence or an anomaly if its P_n^s is larger than T_{max} or smaller than T_{min}, respectively. Otherwise, this short sequence is a weak anomaly signal sequence and will be sent to weak anomaly signal processing module for further processing.

C. Weak Anomaly Signal Processing Module

The weak anomaly signal processing module enlarges the pattern of the weak anomaly signal sequence to make it more distinguishable by predicting its most probable future system call sequence based on a chosen *Prediction Model*.

The prediction model is either the normal or abnormal activity model generated in the training module. For a given weak anomaly signal sequence, its corresponding prediction model is the one which matches this given sequence better. Therefore, we calculate one more sequence probability value of the given weak anomaly signal sequence based on the abnormal activity model (denoted as P_a^s) and compare it with the previous resulted P_n^s. If P_a^s is larger than P_n^s, it means this short sequence matches the abnormal activity model better than the abnormal activity model so we should use abnormal activity model as its prediction model. Otherwise, the normal activity model is chosen as the prediction model.

Now, we can utilize the selected prediction model to predict the most probable future system call sequence. The most probable future sequence is defined as the sequence that matches the prediction model best (i.e., the one with the maximum sequence probability among all possible sequences). Therefore, given a weak anomaly signal sequence, a baseline solution is to enumerate all possible future system call sequences with length n and iteratively calculate the sequence probability values of these sequences according to Eq. (6) where s_1 is the last system call of the given weak anomaly signal sequence. The future sequence with the highest value is the most probable future system call sequence. However, given a finite set of distinct kinds of system calls Q, there are m^n possible future sequences, where $m = |Q|$. Therefore, the computational complexity of this solution is $O(m^n)$ which is exponential to the future sequence length n and hence intractable. To solve this problem, we further propose a variation of Viterbi algorithm (named VV algorithm) to obtain the most probable future system call sequence more efficiently in $O(m^2 n)$ times.

Finally, a most probable extended system call sequence with the enlarged pattern is obtained through combining the predicted most probable future sequence with the given weak anomaly signal sequence. To detect the anomaly signal based on these extended sequence, for each sequence, we calculate a new sequence probability value ($P_n^{s'}$) based on the normal activity model and compared it with a new extended threshold (T_e) to identify this weak anomaly signal sequence as a normal activity sequence (if $P_n^{s'} \geq T_e$) or an anomaly (if $P_n^{s'} < T_e$).

IV. Variation of Viterbi Algorithm

Apart from intrusion detection accuracy, runtime efficiency is also very important for the proposed scheme. The detection module online monitors the system call log to generate real-time short sequences continuously which requires the weak anomaly signal processing module to be able to predict the future system call sequence instantaneously. Otherwise, we can not stop the attacker's hazardous operations on time and the proposed anomaly detection scheme becomes meaningless.

Since the computational complexity of the baseline solution is exponential and intractable, we further propose a variation of Viterbi (VV) algorithm which can obtain the most probable future system call sequences with length n (denoted as $S_p(n)$) more efficiently with $O(m^2 n)$ time complexity.

Given a prediction Markov model defined by a set of distinct kinds of system calls, Q, a transition probability matrix, M, and an initial probability vector, V, we first define $PV(n, z)$ to be the sequence probability value of the most probable future system call sequence ending at system call v with length n. Therefore, $S_p(n)$ is the sequence with the value: $\max_{z=\{1,\ldots m\}} PV(n, z)$ where m is the cardinality of Q.

Since the n^{th} system call is fixed to be z, $PV(n, z)$ is a maximization over the first $n-1$ future system calls and can be represented as Eq. (7).

$$PV(n, z) = \max_{y_1, \ldots, y_{n-1} = \{1, \ldots m\}} \left[\prod_{i=1}^{n-1} p_{y_{i-1}, y_i} \cdot p_{y_{n-1}, z} \right], \quad (7)$$

where y_i is the variable of i^{th} future system call and p_{y_{i-1}, y_i} is the transition probability between variables y_{i-1} and y_i. Since

VV Algorithm

Input: Markov model defined by Q, M and V;
s_{last} : the last monitored system call;
m : cardinality of Q;
n : sequence length.
Output: $S_p(n)$: most probable future sequence.

(i) Initial:
$PV(0, s_{last}) = 1$; $PV(0, others) = 0$;
(ii) Recursion($i = 1, \cdots, n$)**:**
$PV(i, y_i) = \max_{y_{i-1}=\{1,\ldots m\}} \left[PV(i-1, y_{i-1}) \cdot p_{y_{i-1}, y_i}\right]$;
$ptr(i, y_i) = \arg\max_{y_{i-1}} \left[PV(i-1, y_{i-1}) \cdot p_{y_{i-1}, y_i}\right]$;
Store $ptr(i, y_i)$ and $PV(i, y_i)$ in matrices;
(iii) Termination:
$ptr(n+1) = \arg\max_{y_n} [PV(n, y_n)]$;
Obtain $S_p(n)$ by tracing ptr's backward from $ptr(n+1)$;
Return $S_p(n)$;

Fig. 3. VV Algorithm

the value of $p_{y_{n-1}, z}$ depends on y_{n-1} only, we can obtain a recursive equation from Eq. (7) as follows:

$$PV(n, z) = \max_{y_{n-1}=\{1,\ldots m\}} \left[PV(n-1, y_{n-1}) \cdot p_{y_{n-1}, z}\right]. \quad (8)$$

Based on Eq. (8), we can compute $PV(n, z)$ for any system call, $z \in Q$, recursively. The pseudo-code of the developed VV algorithm is shown in Fig. 3. In step (i), we initialize that the probability of future sequences starting from the last system call of online monitored sequence is 1 and the probabilities of starting from other system calls are 0. we apply Eq. (8) in step (ii) to recursively obtain $PV(i, y_i)$ where $PV(i-1, y_{i-1})$ is known for all possible $y_{i-1} \in Q$ by the previous recursion. The value of $PV(i, y_i)$ is stored in a $m \times n$ dynamic programming matrix while a pointer (ptr) is associated with each $PV(i, y_i)$ and point to a system call y_{i-1} which maximizes $PV(i, y_i)$. The algorithm stops at step (iii) when we obtain the $\max_{y_n} [PV(n, y_n)]$, assign the system call y_n which maximizes $PV(n, y_n)$ to the pointer $ptr(n+1)$ and trace pointers backward from $ptr(n+1)$ and return $S_p(n)$.

For each recursion, we need to first enumerate y_i and then enumerate y_{i-1} to find a y_{i-1} maximizing $PV(i, y_i)$ for each $y_i \in Q$ so each recursion costs $O(m^2)$ times. Since there are n recursions, the VV algorithm can obtain $S_p(n)$ in $O(m^2 n)$ times. It is more efficient than the baseline algorithm because it uses dynamic programming matrix to store the local optimal values and avoid enumerating all possible sequences and repeatedly calculating the same probability values.

V. Experimental Study

In this section, we conduct an experimental study on the *sendmail* dataset described in Subsection II. We use the first 20% system calls in abnormal traces and 10% system calls in normal traces as the testing data while other system calls are used as the training data. We also set window sizes of the sliding window in both the training module and the weak anomaly signal processing module to be $w = 3$.

In order to evaluate the performance of the multi-module anomaly detection scheme (called *MADS* in this section), we have conducted the following evaluations: (i) we compare the intrusion detection accuracy of MADS with that of directly using the Markov-based intrusion detection method (called *MID* in this section) proposed by [8]; (ii) we further investigate the effect of increasing the number of predicted future system calls on the intrusion detection accuracy; (iii) we also evaluate the runtime efficiency of the proposed VV algorithm comparing with that of the baseline method.

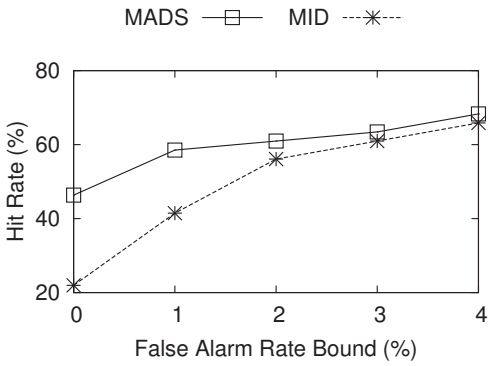

Fig. 4. MADS v.s. MID

The most important intrusion detection accuracy measurement is *Hit Rate*. A hit is a true positive result: it occurs when an abnormal system call sequence is correctly detected as an intrusion. Therefore, the hit rate is defined as the number of hits divided by the total number of abnormal system call sequences and preferred to be as large as possible. Generally, each different detection threshold will result a different "hit rate" but also a different *False Alarm Rate* and increasing detection threshold will enhance both of them. A false alarm is a false positive result: it occurs when a normal system call sequence is detected as an intrusion by error. Therefore, the false alarm rate is defined as the number of false alarms divided by the total number of normal system call sequences. In practice, we normally require the false alarm rates to be very small or not larger than a predefined tolerance bound. Therefore, here, we evaluate the intrusion detection accuracy of both MADS and MID by comparing their hit rates under five different predefined small false alarm rate bounds: 0%, 1%, 2%, 3% and 4%.

Fig. 4 shows the hit rates of both MADS and MID with the number of predicted future system calls in MADS being 2. Overall, we can observe that the hit rates of the proposed MADS outperform those of MID under all three different bounds of false alarm rates. Especially, when the false alarm rate bound is 0%, the hit rates of the proposed MADS is more than twice of that of MID: the hit rate boosts from less than 20% to more than 40%. Therefore, we can say that the proposed MADS can greatly improve the intrusion detection accuracy of the adopted Markov-based intrusion detection system in terms of hit rates under small false alarm rate bounds.

Furthermore, we vary the number of predicted future system calls (i.e., n) in MADS from 1 to 3 to investigate its effect on the intrusion detection accuracy. The results are shown in Fig. 5. we find that, when n increases from 1 to 2, the

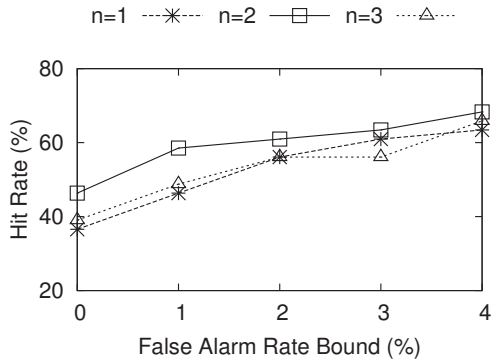

Fig. 5. Varying the number of predicted future system calls in MADS

performance of MADS becomes better. However, when we vary n from 2 to 3, the hit rates of MADS decrease under all false alarm rate bounds. This is because the prediction accuracy of future system calls in the weak anomaly signal processing module falls down when n varies from 2 to 3. Therefore, the performance of the proposed MADS depends on the prediction accuracy of the prediction technique adopted in the weak anomaly signal processing module.

TABLE III
RUNNING TIME OF FUTURE SEQUENCE PREDICTION

Sequence Length	n=1	n=2	n=3
Baseline Method (ms)	3.7	66.2	5917.8
VV algorithm (ms)	5.2	8.6	11.7

To evaluate the runtime efficiency of the proposed VV algorithm comparing with that of the baseline method, we record their running time of predicting future sequences with respect to various predicted sequence length n in Table III. We find that, when $n = 1$, the running time of both the baseline method and the VV algorithm are very short (only $3.7ms$ and $5.2ms$, respectively). When n increases, the running time of the baseline method grows exponentially and reaches up to $66.2ms$ ($n = 2$) and $5917.8ms$ ($n = 3$). On the contrary, the corresponding running time of the VV algorithm is still very small: only $8.6ms$ ($n = 2$) and $11.7ms$ ($n = 3$), respectively. These results confirm the previous complexity analyses: the computational complexity of the VV algorithm is linear to n while that of the baseline method is exponential to n. Therefore, we can assert that the proposed VV algorithm is exponentially more efficient than the baseline method.

VI. CONCLUSION AND FUTURE WORK

This paper is to utilize the sequence prediction techniques to solve the weak anomaly signal detection problem and improve the performances of the existing intrusion detection systems. A variation of Viterbi (VV) algorithm is proposed to predict the future system call sequence exponentially more efficiently. Experimental results demonstrate that: (i) the proposed scheme can greatly improve the intrusion detection accuracy of the existing intrusion detection system (e.g.,this Markov-based intrusion detection method) in terms of hit rates under small false alarm rate bounds; (ii) the performance of the proposed scheme depends on the prediction accuracy of the adopted prediction technique; (iii) the developed VV algorithm is much more efficient than the baseline method.

In the future, we will investigate more types of prediction and intrusion detection methods to further improve the intrusion detection accuracy in terms of hit and false alarm rates.

ACKNOWLEDGMENT

This work is supported by the Australian Research Council's Linkage funding scheme (project number LP100200538).

REFERENCES

[1] J. Evers. (2002) Computer crime costs $67 billion, fbi says,. [Online]. Available: http://www.ctan.org/
[2] E. Mills. (2009) Cybercrime cost firms us$1 trillion globally. [Online]. Available: http://www.zdnetasia.com/news/
[3] P. Garcia-Teodoro, J. E. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, 2009.
[4] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," in *USENIX Security Symposium*, 1998, pp. 1–6.
[5] U. Lindqvist and P. A. Porras, "Detecting computer and network misuse through the production-based expert system toolset (p-best)," in *IEEE Symposium on Security and Privacy (S&P)*, 1999, pp. 146–161.
[6] N. Ye, X. Li, and S. M. Emran, "Decision tree for signature recognition and state classification," in *IEEE Systems, Man and Cybernetics Information Assurance & Security Workshop*, 2000, pp. 1–6.
[7] X. D. Hoang, J. Hu, and P. Bertok, "A multi-layer model for anomaly intrusion detection using program sequences of system calls," in *IEEE International Conference on Network (ICON)*, 2003, pp. 531–536.
[8] N. Ye, Y. Zhang, and C. M. Borror, "Robustness of the markov-chain model for cyber-attack detection," *IEEE Transactions on Reliability*, pp. 116–123, 2004.
[9] Y. Feng, F. Han, X. Yu, Z. Tari, L. Li, and J. Hu, "Terminal sliding mode observer for anomaly detection in tcp/ip networks," in *International Conference on Computer Science and Network Technology (ICCSNT)*, vol. 1, 2011, pp. 617–620.
[10] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *IEEE Symposium on Security and Privacy (S&P)*, 1996, pp. 120–128.
[11] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, pp. 151–180, 1998.
[12] J. Hu, X. Yu, D. Qiu, and H.-H. Chen, "A simple and efficient hidden markov model scheme for host-based anomaly intrusion detection," *Network Magazine of Global Internetworking*, vol. 23, pp. 42–47, 2009.
[13] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *The VLDB Journal*, vol. 16, no. 4, pp. 507–521, 2007.
[14] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *IEEE Symposium on Security and Privacy (S&P)*, 1999, pp. 133–145.
[15] D. Mutz, F. Valeur, G. Vigna, and C. Kruegel, "Anomalous system call detection," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 61–93, 2006.
[16] B. Salamat, T. Jackson, A. Gal, and M. Franz, "Orchestra: intrusion detection using parallel execution and monitoring of program variants in user-space," in *the 4th ACM European conference on Computer systems (EuroSys)*, 2009, pp. 33–46.
[17] F. Maggi, M. Matteucci, and S. Zanero, "Detecting intrusions through system call sequence and argument analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 4, pp. 381–395, 2010.
[18] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni, "A fast automaton-based method for detecting anomalous program behaviors," in *IEEE Symposium on Security and Privacy (S&P)*, 2001, pp. 144–155.
[19] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *IEEE Symposium on Security and Privacy (S&P)*, 1999, pp. 120–132.
[20] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, "Intrusion detection using fuzzy association rules," *Applied Soft Computing*, vol. 9, no. 2, pp. 462–469, 2009.