2013 6th International Congress on Image and Signal Processing (CISP 2013)

Top-k Future System Call Prediction Based Multi-Module Anomaly Detection System

Zhenghua Xu, Xinghuo Yu,
Zahir Tari and Fengling Han
RMIT University, Melbourne,
VIC 3001, Australia
{zhenghua.xu, x.yu, zahir.tari,
fengling.han}@rmit.edu.au

Yong Feng
Harbin Institute of Technology,
Harbin, 150001, China
yong.feng.5601@gmail.com

Jiankun Hu
University of New South Wales,
ADFA, Canberra, ACT 2600, Australia
J.Hu@adfa.edu.au

***Abstract*—Due to the rapid and continuous development of computer networks, more and more intrusion detection techniques are proposed to protect our systems. However, there is a weak anomaly detection problem among the existing system call based intrusion detection systems: the pattern value range of abnormal system call sequences generated by attacks always overlaps to that by normal behaviors so it is difficult to accurately classify the sequences falling into the overlap area by a unique threshold. Instead of using fuzzy inference, we innovatively solve this problem by proposing a top-k prediction based multi-module (abbreviated as TkPMM) anomaly detection system to enlarge patterns of sequences falling into the overlap area and make them more classifiable. We further develop a scalable linear algorithm called top-k variation of the Viterbi algorithm (called TkVV algorithm) to efficiently predict the top-k most probable future system call sequences. Extensive experimental studies show that TkPMM greatly enhances the intrusion detection accuracy of the existing intrusion detection system by up to 25% in terms of hit rates under small false alarm rate bounds and the complexity of our TkVV algorithm is exponential better than that of the baseline method.**

***Index Terms*—Top-k Prediction, Multi-module System, Intrusion Detection, Viterbi Algorithm.**

I. Introduction

System Call is a program signal for requesting a service from the system's kernel [1]. The existing work [2] has demonstrated that short sequences of system call traces generated by normal program executions are stable and consistent during programs' normal activities and The signature is very likely to be perturbed when abnormal activities (i.e., attacks) occurs. Therefore, they are very good discriminator between the normal use behavior and abnormal activities. System calls have already been widely adopted by many existing intrusion detection systems such as, [3]–[10].

However, there exists a major problem for the existing system call based anomaly detection systems: the pattern value range of anomalous system call sequences generated by attacks always overlaps to that of normal behavior system call sequences; while the existing anomaly detection systems activate intrusion alarms whenever the pattern value deviations between given system call sequences and the normal activities exceed a predefined unique threshold, it is difficult for them to accurately classify the sequences falling into the overlapped area with high hit (true alarm) rate and low false alarm rate. An example is shown in Fig. 1 where 40 pre-labeled anomalous short sequences and normal short sequences are processed by a Markov based intrusion detection system and the natural logarithms of sequence probabilities are used as pattern values. As we can see, the pattern value range of normal sequences is $[0, -10]$ and that of anomalous sequences is $[-5, -20]$ so there exists an overlap area $[-5, -10]$. Due to this overlap, using a unique threshold to classify the sequences falling into this area will inevitably encounter some erroneous classification and lead to either a relatively low hit rate or a high false alarm rate. Therefore, we call the sequences whose pattern values fall into the overlapped area *Weak Anomaly Sequence* and this problem *Weak Anomaly Detection Problem.*

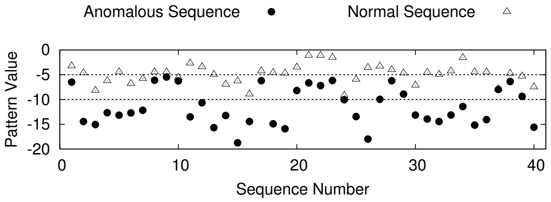

Figure 1. Motivation Example

A most common solution for this problem is to use the fuzzy-based inference mechanism to provide a soft boundary between normal and abnormal sequences [11], [12]. However, the fuzzy inference not only needs to introduce more parameters and make the system become more complicated, but also does not solve this problem from the essential aspect: the distributions of normal and abnormal sequences are not changed and their overlap remains the same. Motivated by this, we propose a **T**op-***k*** **P**rediction based **M**ulti-**M**odule (abbreviated as **TkPMM**) anomaly detection system to solve this problem by utilizing the sequence prediction techniques to enlarge the pattern of weak anomaly sequences and make them more classifiable. It's worth noting that, the fuzzy inference can still be applied after the pattern enlargement to soft the boundary and further improve the intrusion detection accuracy. Moreover, we alaso evelop a **T**op-***k*** **V**ariation of **V**iterbi algorithm (named *TkVV Algorithm*), by which the cost of predicting the top-k future system call sequences is linear to

978-1-4799-2764-7/13/$31.00 ©2013 IEEE

the predicted sequence length.

Moreover, in this proposed system, we can adopt various sequence based intrusion detection methods to calculate the pattern value of system call sequences and detect anomalies. To keep it simple, in this paper, we use a Markov-based intrusion detection method [13] where the *Sequence Probability* is used as pattern value. Extensive experimental studies are conducted on the benchmark synthetic *sendmail* data to evaluate the performance of the TkPMM anomaly detection system and the TkVV algorithm.

The rest of this paper is organized as follows. The TkPMM anomaly detection system and the TkVV algorithm are presented in detail in Section II and Section III, respectively. We include the extensive experimental studies and their results in Section IV. We conclude the paper and discuss the future work in Section V.

II. TkPMM Anomaly Detection System

The proposed TkPMM anomaly detection system consists of three modules: *Training Module*, *Detection Module* and *Weak Anomaly Sequence Processing Module*.

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 T_{max} or lower than a predefined minimum threshold 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, they are first identified as a weak anomaly sequence and then sent to the weak anomaly sequence 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 sequences and will be sent to the weak anomaly sequence processing module.

In the weak anomaly sequence processing module, we enlarge the pattern of the weak anomaly sequences to make them more classifiable by predicting their top-k most probable future system call sequences based on either the normal or abnormal activity model generated in the training module. We obtain the top-k most probable extended system call sequences through combining the predicted top-k most probable future sequences with the given weak anomaly sequences. Then, based on the top-k most probable extended sequences, a post-processing step is invoked to calculate a new pattern value, which is compared with a new extended threshold to classify the weak anomaly sequences.

The diagram of TkPMM anomaly detection system is shown in Fig. 2 and some details of each module will be presented in the following subsections.

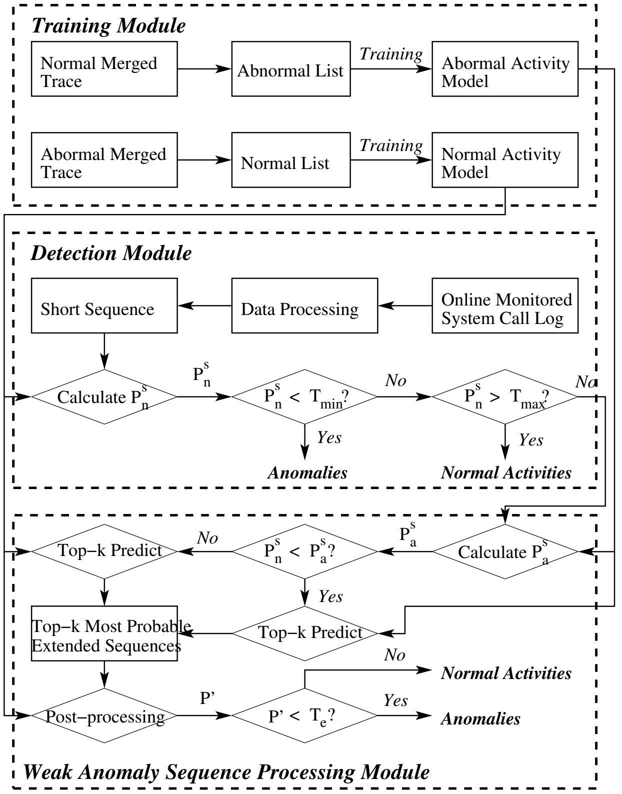

Figure 2. TkPMM Anomaly Detection System

A. Training Module

In the 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 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,

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

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.

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

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

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.

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.

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

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

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 [14] 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 [4], [15], [16] can be used to calculate various pattern values according to different requirements. To keep it simple, a Markov-based intrusion detection method proposed by [13], which uses the *Sequence Probability* as the pattern value, is adopted in this work. The sequence probability indicates how probably this sequence occurs in the given Markov model and is defined as follows:

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

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

In [13], 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 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 sequences.

In order to solve this weak anomaly detection problem, in our system, 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} (T_{min}) is defined so large (small) that, if the resulted P_n^s of a short sequence is larger (smaller) than T_{max} (T_{min}), this sequence is a sequence of normal activities (anomalies) 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 sequence and will be sent to weak anomaly sequence processing module for further handling.

C. Weak Anomaly Sequence Processing Module

The weak anomaly sequence processing module can enlarge the pattern of a weak anomaly sequence to make it more classifiable through predicting its top-k most probable future system call sequences based on a chosen *Prediction Model*.

The prediction model is either the normal activity model or the abnormal activity model generated in the training module. For a given weak anomaly 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 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 normal activity model so we should use abnormal activity model as its prediction model. Otherwise, the normal activity model is chosen.

Given the selected prediction model, we can predict the top-k most probable future system call sequences. The top-k most probable future sequences are defined as the sequences that matches the prediction model top-k best (i.e., the ones with the top-k maximum sequence probabilities among all possible sequences). Therefore, given a weak anomaly 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 sequence. The future sequences with the top-k highest value are the most probable future system call sequences. 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 top-k variation of Viterbi algorithm (named TkVV algorithm) to obtain the most probable future system call sequence more efficiently in $O(km^2n)$ times. This algorithm will be described detailedly in Section III.

Then, the top-k most probable extended system call sequences are obtained through combining the predicted top-k most probable future sequences with the given weak anomaly sequence. Given these top-k most probable extended sequences, A post-processing step is invoked to calculate a new pattern value, where we first calculate the sequence probability values of these top-k extended sequences based on normal activity model and use a heuristic post-processing equation to unite these values and get a new pattern value (P'). The heuristic post-processing equation used in this work is as follows:

$$P'=\sum_{i=1}^{k}\left(P_i^s\cdot\frac{w_i}{Z}\right), \tag{7}$$

where P_i^s is the sequence probability of the top i^{th} extended sequence, w_i is the weight of P_i^s and Z is the summation of w_i. The value of w_i can be assigned heuristically or estimated by training historic data.

Finally, This new pattern value is compared with a new extended threshold (T_e) to identify this weak anomaly sequence as a normal activity sequence (if $P' \geq T_e$) or an anomaly (if $P' < T_e$).

III. TkVV Algorithm

Besides intrusion detection accuracy, runtime efficiency is also very important for the proposed TkPMM anomaly detection system. Because the detection module online monitors the system call log to generate real-time short sequences rapidly and continuously, it requires the weak anomaly sequence 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 system becomes meaningless.

As mentioned above, the computational complexity of the baseline solution is exponential to the length of predicted future sequence so it is intractable. Furthermore, as the first work to use sequence prediction solutions in this area, there is not any existing solution. Therefore, we present the Top-k Variation of Viterbi (TkVV) algorithm to improve the runtime efficiency. This algorithm can obtain the top-k most probable future system call sequences with length n more efficiently with $O(km^2n)$ time complexity.

A. Algorithm

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 the following two symbols: i) $TV(k,n,z)$ is a k times 1 matrix where its i^{th} item is the sequence probability value of the i^{th} most probable future system call sequence ending at system call v with length n; ii) $TK_{j=\{1,\cdots,m\}}[f(x_j)]$ is a symbol which returns a k times 1 matrix whose i^{th} item is the i^{th} largest values of $f(x_j)$ when $j = \{1,\cdots,m\}$. Therefore, the aim of the TkVV algorithm is to efficiently obtain $TV(k,n+1,s_{end}) = TK_{z=\{s_1,\ldots,s_m\}}[TV(k,n,z)]$, where n is the length of predicted future system call sequence, m is the cardinality of Q and s_{end} is a virtual state indicating the end of sequence.

Moreover, since the n^{th} system call is fixed to be z, $TV(k,n,z)$ is a top-k maximization over the first $n-1$ future system calls. We are able to represent $TV(k,n,z)$ as Eq. (8).

$$TV(k,n,z) = TK_{y_1,\ldots,y_{n-1}=\{s_1,\ldots,s_m\}} \left[\prod_{i=1}^{n-1} p_{y_{i-1},y_i} \cdot p_{y_{n-1},z} \right], \tag{8}$$

where y_i is the variable of i^{th} future system call in weak anomaly sequence and p_{y_{i-1},y_i} is the transition probability between variables y_{i-1} and y_i. Due to the Markov assumption, the value of $p_{y_{n-1},z}$ depends on y_{n-1} only. Thereby, all probable future sequences with the same $y-1$ future system call will share the same transition probability value $p_{y_{n-1},z}$ such that we can transform the Eq. (8) to be a recursive equation as shown in Eq. (9):

$$TV(k,n,z) = TK_{y_{n-1}=\{s_1,\ldots,s_m\}} \left[TV(k,n-1,y_{n-1}) \cdot p_{y_{n-1},z} \right]. \tag{9}$$

```
TkVV Algorithm
Input: Markov model defined by $Q$, $M$ and $V$;
   $s_{last}$ : the last monitored system call;
   $m$ : cardinality of $Q$; $n$ : sequence length;
   $P^s$ : sequence probability of monitored sequence; $k$.
Output: $TV(k, n+1, s_{end})$

(i) Initial:
   $TV(k, 0, s_{last}) = P^s$;
   $TV(k, 0, others) = 0$;
(ii) Recursion($i = 1, \cdots, n$):
   For $y_i$ from $s_1$ to $s_m$
      For $y_{i-1}$ from $s_1$ to $s_m$
         calculate $TV(k, i-1, y_{i-1}) \cdot p_{y_{i-1},y_i}$;
      store top-$k$ results in $TV(k, i, y_i)$;
      store $TV(k, i, y_i)$ in a three-dimensional matrix;
(iii) Termination:
   For $y_n$ from $s_1$ to $s_m$
      save top-$k$ values in $TV(k, n, y_n)$ to $TV(k, n+1, s_{end})$;
Return $TV(k, n+1, s_{end})$;
```

Figure 3. TkVV Algorithm

Based on Eq. (9), we can compute $TV(k,n,z)$ for any system call, $z \in Q$, recursively. The pseudo-code of the TkVV algorithm is shown in Fig. 3.

In step (i) of the TkVV algorithm, we set up two types of initial values: the probability of future sequences starting from the last system call of online monitored sequence is set to be the sequence probability value of the monitored sequence, i.e., P^s (to maintain consistency, it is represented as $TV(k,0,s_{last})$ but it is a value instead of matrix); and the probabilities of starting from other system calls are set to be 0 (similarly, they are represented as $TV(k,0,others)$.

we apply Eq. (9) in step (ii) to recursively obtain $TV(k,i,y_i)$ for i from 1 to n. Specifically, for each length i, we iteratively assign all possible system calls $(s_1,\ldots,s_m)$ to be the values of y_i and y_{i-1} which are variables of the i^{th} and $i-1^{th}$ system calls respectively in predicted sequence. Since the matrix $TV(k,i-1,y_{i-1})$ has already been obtained for all possible $y_{i-1} \in Q$ by the previous recursion, for each y_i, we can iteratively calculate $TV(k,i-1,y_{i-1}) \cdot p_{y_{i-1},y_i}$ and store the top-k results in matrix $TV(k,i,y_i)$. Then we store the resulted $TV(k,i,y_i)$ in a $m \times n \times k$ three-dimensional dynamic programming matrix for reuse. This algorithm stops at step (iii) when matrix $TV(k,n+1,s_{end})$ is obtained by selecting and storing the top-k values in among all possible $TV(k,n,y_n)$. We return $TV(k,n+1,s_{end})$ as the result of running this algorithm and the k items in matrix $TV(k,n+1,s_{end})$ are the sequence probability values of the top-k most probable extended system call sequences, i.e., P_i^s.

B. Complexity Analysis

For each recursion in i, we need to iteratively enumerate the values of both $y_i - 1$ and y_i from s_1 to s_m so there are m^2 possible pairs of $TV(k, i-1, y_{i-1})$ and p_{y_{i-1},y_i}, i.e., $O(m^2)$. Furthermore, because $TV(k, i-1, y_{i-1})$ is a k times 1 matrix, the operation of multiplying $TV(k, i-1, y_{i-1})$ by p_{y_{i-1},y_i} costs $O(k)$ times. Therefore, the computational cost of each recursion is $O(km^2)$. Due to there are n recursions, the time complexity of the TkVV algorithm is $O(km^2n)$, which is linear to the length of predicted most probable future system call sequence n. Therefore, we can state that the TkVV algorithm make the prediction of future sequence tractable and scalable. It is far more efficient than the baseline algorithm because it uses three-dimensional dynamic programming matrix to store the local optimal values and avoid enumerating all possible sequences and repeatedly calculating the same probability values.

IV. Experimental Study

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 [2] and [10]. 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 a unique sliding window size in both the training module and the weak anomaly sequence processing module, i.e., $w = 3$.

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 TkPMM anomaly detection system and MID by comparing their hit rates under six different predefined small false alarm rate bounds: 0%, 1%, 2%, 3%, 4% and 5%.

A. Effect of TkPMM Anomaly Detection System

Fig. 4 shows the results of both TkPMM anomaly detection system and MID. Here, the values of variables in TkPMM system are as follows: $n = 2$, $k = 2$, $w_1 = 0.9$ and $w_2 = 0.1$. Generally, we can witness the hit rates of the proposed TkPMM anomaly detection system outperform those of MID under all six different false alarm rate bounds. When the bound of false alarm rate is 0%, the hit rate of TkPMM system is more than twice of that of MID. i.e., the hit rate jumps from 20% up to more than 45% by using TkPMM system. A 20% enhance resulted by the TkPMM anomaly detection system is also found when the false alarm rate bound is set to be 1%. As for other bounds, the TkPMM anomaly detection system stably boosts the hit rate of MID by around 5%.

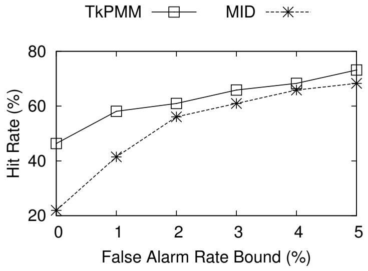

Figure 4. TkPMM v.s. MID

Therefore, we can say that the TkPMM anomaly detection system can greatly improve the intrusion detection accuracy of the adopted Markov-based intrusion detection system by up to 25% in terms of hit rates under small false alarm rate bounds. Furthermore, the smaller the false alarm rate bound, the higher the degree of improvement.

B. Effect of varying n

Furthermore, we vary the number of predicted future system calls (i.e., n) in TkPMM anomaly detection system 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 hit rates of the TkPMM anomaly detection system increase $5\% - 10\%$ under all false alarm rate bounds. However, when we vary n from 2 to 3, the hit rates of TkPMM system decrease. This is because the prediction accuracy of future system calls in the weak anomaly sequence processing module falls down when n varies from 2 to 3. Therefore, the performance of the proposed TkPMM anomaly detection system depends on the prediction accuracy of the prediction technique adopted in the weak anomaly sequence processing module.

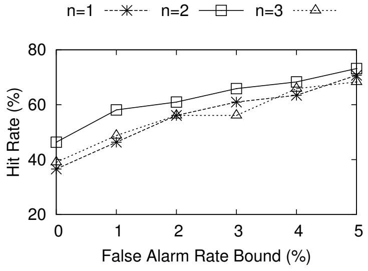

Figure 5. Varying the number of predicted future system calls in TkPMM system

C. Effect of varying k

In this subsection, we investigate the impact of the value of k on the performance of the TkPMM anomaly detection system where $n = 2$ and the value of k is varied from 1 to 3. We heuristically set $w_1 = 0.9$ and $w_2 = 0.1$ when $k = 2$, while $w_1 = 0.8$, $w_2 = 0.15$ and $w_3 = 0.05$ when $k = 3$. The experimental results are shown in Fig. 6; overall, TkPMM system achieves the best performance when $k = 2$. Specifically, when the false alarm rate bound is extremely small (i.e., from 0% to 2%), the hit rates of $k = 2$ and $k = 3$ are higher than that of $k = 1$. Therefore, taking more highly probable future sequeneces into account will increase the tolerance of prediction errors which lead to higher anomaly detection accuracy in turn. We also find that the performance of $k = 1$ and $k = 2$ are better than that of $k = 3$ when the bound of false alarm rate is larger than 2%. This implies that the performance of TkPMM system may be weakened if the tolerance of prediction errors is too high.

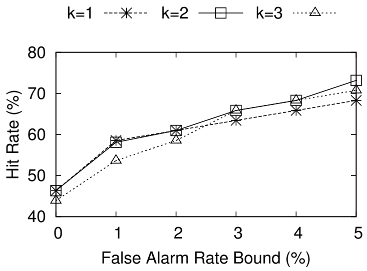

Figure 6. Varying the value of k in TkPMM system

D. Runtime Efficiency

To evaluate the runtime efficiency of the proposed TkVV 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 I. We find that, when $n = 1$, the running time of both the baseline method and the TkVV algorithm are very short (only $3.7ms$ and $5.6ms$, 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 TkVV algorithm is still very small: only $10.3ms$ ($n = 2$) and $15.1ms$ ($n = 3$), respectively. These results confirm the previous complexity analyses: the computational complexity of the TkVV algorithm is linear to n while that of the baseline method is exponential to n. Therefore, we can assert that the proposed TkVV algorithm is tractable, scalable and also exponentially more efficient than the baseline method.

TABLE I
RUNNING TIME OF FUTURE SEQUENCE PREDICTION

Sequence Length	n=1	n=2	n=3
Baseline Method (ms)	3.7	66.2	5917.8
TkVV algorithm (ms)	5.6	10.3	14.1

V. CONCLUSION AND FUTURE WORK

This paper has proposed and implemented a Top-k Prediction based Multi-Module anomaly detection system, which has demonstrated to be able to greatly improve the intrusion detection accuracy of the adopted Markov-based intrusion detection method by up to 25% in terms of hit rates under different small false alarm rate bounds; and the lower the false alarm rate bound, the higher the degree of improvement. A top-k variation of Viterbi (TkVV) algorithm is proposed to predict the future system call sequence in linear time. In the future, more types of intrusion detection methods will be implemented and tested by using the TkPMM anomaly detection system to further evaluate its advantages on improving the anomaly detection accuracy.

ACKNOWLEDGMENT

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

REFERENCES

[1] WikiPedia. System call. [Online]. Available: http://en.wikipedia.org/wiki/System_call

[2] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 1996, pp. 120–128.

[3] X. D. Hoang, J. Hu, and P. Bertok, "A multi-layer model for anomaly intrusion detection using program sequences of system calls," in *Proceedings of 11th IEEE International Conference on Network (ICON)*, 2003, pp. 531–536.

[4] 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, no. 1, pp. 42–47, 2009.

[5] 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, Oct. 2007.

[6] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 1999, pp. 120–132.

[7] 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, Oct. 2010.

[8] 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.

[9] B. Salamat, T. Jackson, A. Gal, and M. Franz, "Orchestra: intrusion detection using parallel execution and monitoring of program variants in user-space," in *Proceedings of the 4th ACM European conference on Computer systems (EuroSys)*, 2009, pp. 33–46.

[10] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 1999, pp. 133–145.

[11] J. Gmez, F. Gonzlez, and D. Dasgupta, "An immuno-fuzzy approach to anomaly detection," in *in Proceedings of the IEEE International Conference on Fuzzy Systems FUZZIEEE*, 2003.

[12] X. D. Hoang, J. Hu, and P. Bertok, "A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference," *J. Netw. Comput. Appl.*, vol. 32, no. 6, pp. 1219–1228, Nov. 2009.

[13] 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.

[14] Z. Ghahramani. Bayesian methods for machine learning. [Online]. Available: http://www.gatsby.ucl.ac.uk/ zoubin/tmp/tutorial.pdf

[15] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni, "A fast automaton-based method for detecting anomalous program behaviors," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2001, pp. 144–155.

[16] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, "Intrusion detection using fuzzy association rules," *Applied Soft Computing*, vol. 9, no. 2, pp. 462–469, Mar. 2009.