Evaluating Host-based Anomaly Detection Systems: Application of the One-class SVM Algorithm to ADFA-LD

Miao Xie, Jiankun Hu and Jill Slay

School of Engineering and Information Technology
University of New South Wales at the Australian Defence Force Academy
Canberra, Australia
Email: {m.xie,j.hu,j.slay}@adfa.edu.au

***Abstract*—ADFA-LD is a recently released data set for evaluating host-based anomaly detection systems, aiming to substitute the existing benchmark data sets which have failed to reflect the characteristics of modern computer systems. In a previous work, we had attempted to evaluate ADFA-LD with a highly efficient frequency model but the performance is inferior. In this paper, we focus on the other typical technical category that detects anomalies with a short sequence model. In collaboration with the one-class SVM algorithm, a novel anomaly detection system is proposed for ADFA-LD. The numerical experiments demonstrate that it can not only achieve a satisfactory performance, but also reduce the computational cost largely.**

I. Introduction

Detection of the cyber attacks against a host through its system call traces has been an active research topic over the past few decades [1]. However, until very recently, the UMN [2] and DARPA [3] intrusion detection data sets, which were compiled a decade ago and have lost most of their relevance to modern computer systems, were still employed as the benchmark to evaluate a host-based anomaly detection system. In this context, ADFA Linux data set (ADFA-LD) is released [4], with thousands of normal traces collected from a host configured to represent a contemporary Linux server and hundreds of abnormal traces resulted from six latest types of cyber attack. It is believed that ADFA-LD will be a reliable substitute for those obsolete benchmark data sets.

The techniques of detecting malicious (abnormal) system call traces can be classified into two major categories: short sequence-based and frequency-based [5]. A short sequence-based technique establishes a model for the sub-sequences of the normal traces, and a test instance deviating significantly from it will be considered as abnormal. For example, the Forrest's pioneered 'lookahead' algorithm builds a database by sliding a window of fixed-length across the normal traces and, then, a test trace containing a percentage of mismatch beyond a threshold is regarded as abnormal [6] [7]. Kosoresow et al. extended the 'lookahead' algorithms by counting the mismatches within small, fixed-length sections of the traces [8], such that the robustness is largely increased for handling long traces. Moreover, statistical learning theory is widely used for dealing with short sequences, which learns a model statistically for summarising the inherent relationships hidden behind the normal traces. The typical examples include hidden Markov model (HMM) [8] [9] [10] [11] [12], artificial neural network (ANN) [13] [14], semantic data mining [15] and support vector machine (SVM) [16]. Although short sequence-based techniques are able to generate an accurate normal profile, their learning procedures are often time-consuming. Frequency-based techniques, on the contrary, are computationally efficient by ignoring the positional information of the system calls within a trace. In particular, each trace is transformed into a fixed-length vector based on the concept of 'frequency', and an abnormal frequency vector can be detected similarly using a variety of algorithms such as k-nearest neighbour (kNN) [17] [18] [19] [20], clustering analysis [21] and SVM [22]. However, their accuracies are usually inferior in modelling a normal profile due to loss of positional information.

Most of the above techniques are developed specifically for the UMN and DARPA intrusion detection data and unable to work properly on ADFA-LD [15]. First, the normal traces in ADFA-LD are collected promiscuously from multiple programs, rather than being assorted according to each specific program. This feature introduces additional complexity into modelling normal behaviours. Second, the cyber attacks launched against that host are more deliberate than their antecedents and, hence, each abnormal trace involves a smaller footprint to be detected. Undoubtedly, the separability between normal and abnormal is weaker in ADFA-LD. Creech et al. constructed a semantic model (dictionary) for the short sequences of ADFA-LD, in the forms of 'word' and 'phrase' [15]. Based on the dictionary, the HMM, extreme learning machine (ELM) and one-class SVM algorithms were evaluated, which shown that, at a false positive rate (FPR) of 15%, a detection actuary (ACC) of 90% can be obtained by the ELM algorithm and 80% by the one-class SVM algorithm. However, learning the dictionary is extremely time-consuming, which takes approximately an entire week. Combining the kNN and k-means clustering (kMC) algorithms with a frequency-based

978-1-4799-5148-2/14/$31.00 ©2014 IEEE

model can reduce the computational cost significantly, as presented in [18] [21]. Nevertheless, it reached only an ACC of up to 60% at a FPR of 20%, which is not a satisfactory performance. Therefore, in this paper, we intend to further exploit the potential of short sequence-based techniques while maintaining the computational cost at an acceptable level, where the one-class SVM algorithm is utilised for detection.

We obtain a short sequence matrix by continuously transforming the training traces into the vectors of fixed-length and, in order to avoid unnecessary computing, a simple method is introduced to eliminate the duplicated vectors occurring in the matrix. In regard of the principle of classification that the one-class SVM algorithm depends upon, stronger separability between normal and abnormal is enforced artificially by weighting each short sequence with its corresponding frequencies of system call computed from the training traces. Based on the resulting short sequence model, the one-class SVM algorithms is applied to training and detecting, with the results given in the form of RoC curves against different parameters.

The rest of this paper is organised as follows. Section II presents a brief introduction for ADFA-LD and details the proposed short sequence model. Section III introduces how to training and detecting with the one-class SVM algorithm. Next, the details of the numerical experiments are given in section IV. Finally, Section V summarises this work and proposes some problems for the future study.

II. ADFA-LD AND THE SHORT SEQUENCE MODEL

A comprehensive introduction for ADFA-LD can be found in [4], including the configuration of the host, normal user behaviours, routes of the cyber attacks and setting of the Audit daemon. In this paper, we present only a brief overview for ADFA-LD, as follows. During a given sampling period, the host that is configured to represent a modern Linux sever captures the system call traces where legitimate programs are operated as usual. Subsequently, the cyber attacks, i.e., Hydra-FTP, Hydra-SSH, Adduser, Java-Meterpreter, Meter-preter and Webshell, are launched in turn against the host, each of which results in 8-20 abnormal traces. The composition of ADFD-LD is shown in Table I.

TABLE I
THE COMPOSITION OF ADFA-LD

Trace Type	Number	Label
Training	833	normal
Validation	4373	normal
Hydra-FTP	162	attack
Hydra-SSH	148	attack
Adduser	91	attack
Java-Meterpreter	125	attack
Meterpreter	75	attack
Webshell	118	attack

The short sequences are obtained simply by transforming the training traces continuously into a $n \times k$ matrix, say $\mathbf{T}$, where each row in $\mathbf{T}$ denotes a short sequence and n is equal to the total number of the system calls involved in the taring traces divided by the given fixed-length k ($k > 1$). The criteria for selecting k may be deliberately designed; for example, Eskin [9] et al. adopted the conditional entropy of the training data set to determine the optimal k and demonstrated that the k corresponding to a lower conditional entropy results often in a better performance. Furthermore, richer information can be presented in the short sequences if it slides a window across the traces. When adjusting the step width of the sliding window, a balance may be reached between the latent information and the computational cost. But, the two enhancements both will incur a computational cost grown exponentially. According to the numerical experiments, in terms of ADFA-LD, similar performance can be obtained by a relatively wide range of k, and the sliding window cannot make an impressive improvement in performance. Thus, in this paper, k is experimentally determined and the sliding window is not adopted.

Since, as the subsequent detection algorithm is based on the principle of classification, duplicated data points will make no sense, the identical rows should be eliminated from $\mathbf{T}$. Direct elimination by comparing the rows with each other will cost $O(n^2)$ computational complexity (n is about 100,000 for $k = 10$ in terms of ADFA-LD) and, hence, is computationally expensive. In this paper, a simple method is proposed for deleting duplicated rows. Supposing that x ($1 \times k$ vector) denote a short sequence, for all $x \in \mathbf{T}$, $x^* = x \times c$, where $c = [1, 2, \cdots, k]^T$ is a coefficient vector. Next, we delete the duplicated rows in $\mathbf{T}$ which have identical values for x^* by using a regular search algorithm (e.g., binary search algorithm). It can be noted that this method costs only a computational complexity of up to $O(nlog(n))$. Moreover, the coefficient vector can be specified arbitrarily, with the criteria that the higher the gradient, the fewer collisions occur for the different short sequences as to x^*.

Secondly, a better performance can be expected if stronger separability is enforced between normal and abnormal. We combine the system calls with their frequencies appeared in the training data set, i.e., a short sequence composed of frequently used system calls will deviate more significantly from those composed of rarely used system calls. Supposing that t denotes the index of a system call and f_t its corresponding frequency appeared in $\mathbf{T}$, a new training data set $\bar{\mathbf{T}}$ can be obtained by computing

$$t = \mathbf{T}(i, j) \quad \bar{\mathbf{T}}(i, j) = t \times f_t$$

for $i = 1, 2, \cdots, k$ and $j = 1, 2, \cdots, n$.

III. DETECTION USING THE ONE-CLASS SVM ALGORITHM

The one-class SVM algorithm was proposed by Schölkopf et al. in [23] which, unlike its original linear version that separates two classes of data maximally by a hyper plane in feature space, attempts to separate the entire data set from the origin. In other words, it aims to find a hyperplane that separates the data from the origin with maximal margin, of which the

optimisation problem can be described as

$$\min_{\zeta_i,\rho\in\Re} \quad \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\zeta_i - \rho$$
$$\text{subject to} \quad (\omega\cdot\phi(x_i)) \geq \rho - \zeta_i, \zeta_i > 0$$

where ω is the normal vector of the hyperplane in the feature space, ρ offset from the origin, $0 < \nu < 1$ a parameter that controls the tradeoff between the maximised distance from the origin and the maximal number of the data points covered by the region and ζ_i the slack variables that allow some of the data points to lie within a soft margin. Using the method of Lagrange multipliers, the above problem can be rewritten as

$$\min \quad \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j K_\phi(x_i, x_j)$$
$$\text{subject to} \quad 0 \leq \alpha_i \leq \frac{1}{\nu n}, \sum_{i=1}^{n}\alpha_i = 1$$

where α_i are the Lagrange multipliers and K_ϕ is a kernel function that maps the input space implicitly into the feature space through the computations of dot product. Table II summarises the commonly used kernel functions, where x_1 and x_2 denote two row vectors. According to the Lagrange multipliers, the decision function is

$$f(y) = \text{sgn}\left((w\cdot\phi(x_i)) - \rho\right) = \text{sgn}\left(\sum_{i=1}^{n}\alpha_i K_\phi(y, x_i) - \rho\right)$$

where y is the test instance.

TABLE II
KERNEL FUNCTIONS

Type	Definition
linear	$K_\phi(x_1, x_2) = x_1 x_2^T$
polynomial	$K_\phi(x_1, x_2) = \left(\gamma x_1 x_2^T + c_0\right)^d$
radial basis function	$K_\phi(x_1, x_2) = \exp\left(-\gamma\|x_1 - x_2\|^2\right)$
sigmoid	$K_\phi(x_1, x_2) = \tanh\left(\gamma x_1 x_2^T + c_0\right)$

The produced decision function will be used to detect abnormal traces, which is operated in an online manner. That is, once a process is being monitored, the operating system captures its system calls constantly; every k system calls are organised immediately into a short sequence y (a row vector) and weighted with the frequencies obtained form $\bar{\mathbf{T}}$, i.e.,

$$t = y_i \quad \bar{y}_i = t \times f_t$$

for $i = 1, 2, \cdots, k$. It detects $\bar{y}$ using the decision function, with '+1' and '-1' indicating normal and abnormal respectively. For different processes, the number of the short sequences being detected may be varying, the final decision is made by examining the rate that divides the number of the short sequences labelled as '-1' by the total number against a given threshold η. Assuming that the rate is denoted by r, if $r \geq \eta$, a trace will be eventually labelled as abnormal; otherwise, normal.

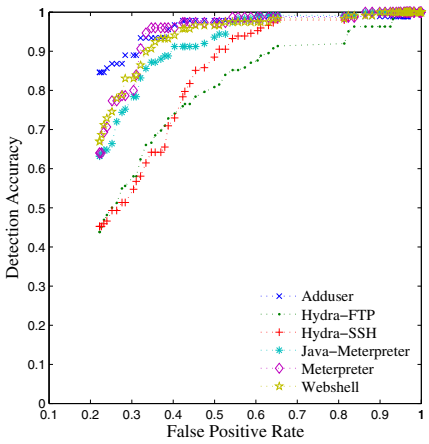

Fig. 1. RoC curves, $k = 3$, $0 < \eta < 1$

IV. NUMERICAL EXPERIMENTS

The numerical experiments are implemented using LIBSVM V3.18 [24] on Matlab R2011a. They seek to answer the following questions: (1) what is the difference between the kernel functions in terms of performance; (2) how is the performance relating to the parameters; (3) how the fixed-length k affects the performance; (4) what is the optimal range of η; and (5) how much time the entire detection algorithm costs in practice.

The training data set $\bar{\mathbf{T}}$ to be input into the one-class SVM algorithm is obtained from the 833 training traces. The produced decision function is then used to test every short sequence embedded in the validation and attack traces. The final decision made for each trace, either normal or abnormal, is determined by the r and η. If a validation trace is reported as abnormal, a false positive is incurred. The FPR is equal to the total number of false positives divided by the length of the validation data set. Conversely, the total number of the successful detections against the attack traces divided by the length of the attack data set yields the ACC.

TABLE III
PARAMETERS

$k = 3$	$\nu = 0.5$ $\gamma = 0.33$ $c_0 = 0$ $d = 3$
$k = 5$	$\nu = 0.5$ $\gamma = 0.2$ $c_0 = 0$ $d = 3$
$k = 8$	$\nu = 0.5$ $\gamma = 0.125$ $c_0 = 0$ $d = 3$
$k = 10$	$\nu = 0.5$ $\gamma = 0.1$ $c_0 = 0$ $d = 3$

In total, the aforementioned kernel functions together with $k = 3, 5, 8, 10$ are tested. It is found that the different kernel functions produce an almost identical result for a given k and, when the corresponding parameters are manually adjusted within a reasonable range, there is no significant change occurred in the performance. Therefore, Figures 1-4 show only the RoC curves resulted from the polynomial kernel function for different k against $0 < \eta < 1$ and the default parameters are summarised in Table III.

Obviously, the performance degrades along with k's increase, mainly reflected by the rising FPRs. However, a over

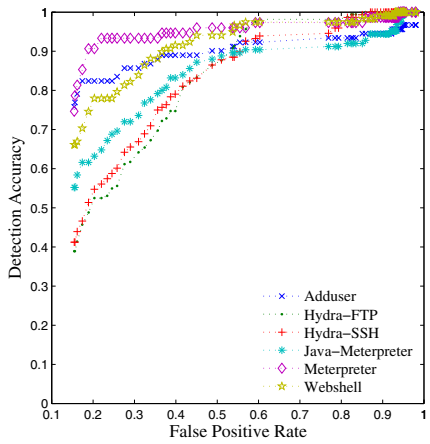

Fig. 2. RoC curves, $k = 5$, $0 < \eta < 1$

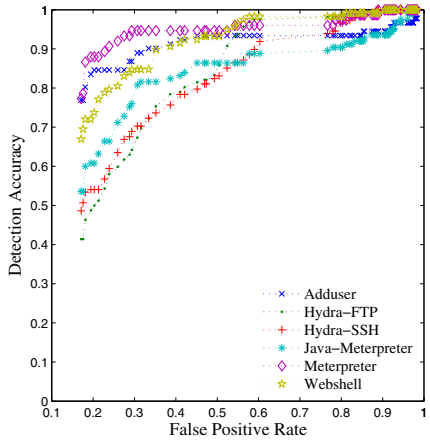

Fig. 3. RoC curves, $k = 8$, $0 < \eta < 1$

small k (e.g., $k = 3$) will cause a worse performance too. Overall, the best performance occurs for $k = 5$, where an average ACC of 70% is achieved at a FPR of around 20%. This performance is already quite close to that reached by applying the one-class SVM algorithm to a semantic model [15]. η is tested with a step width of 0.01 between 0 and 1. In terms of all the figures, it is consistent that an optimal balance between ACC and FPR happens for η's 3^{rd}-5^{th} value. It indicates that $\eta \in [3\%, 5\%]$ can be employed as a criteria to specify threshold in practice. With respect to the type of cyber attack, Adduser and Meterpreter are relatively easy to be detected, for which the best ACCs can reach 90% and 82% respectively at a FPR of 20%. On the contrary, against Hydra-FTP and Hydra-SSH, the best ACCs are only 52% and 48% with a high FPR (about 25%). This result suggests that the proposed technique is not generally effective against any type of cyber attack, and a further study is still ongoing. Finally, the largest contribution made by the proposed technique is the reduction in computing, which means each experiment,

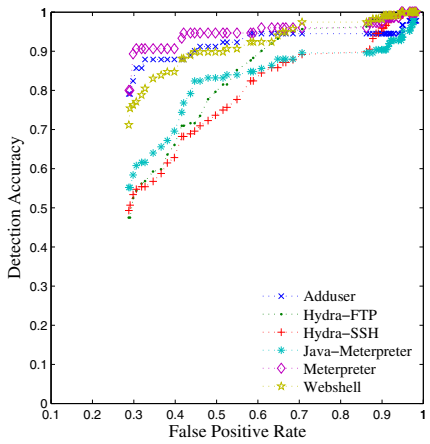

Fig. 4. RoC curves, $k = 10$, $0 < \eta < 1$

including both the procedures of training and testing, takes only around 15s rather than a couple of days.

V. Conclusion

In this paper, we apply the one-class SVM algorithm to ADFA-LD on the basis of a short sequence model. Since duplicated entries are eliminated from the short sequences, and stronger separability between normal and abnormal is enforced, the proposed technique is able to achieve an acceptable performance while maintaining the computational cost at a low level. There are still some gaps to be filled in future work. First, more advanced strategies may be utilised to weight the short sequences for maximising the separability between normal and abnormal. Second, some details may be further sharpened, such as the criteria of selecting k and introduction of user-defined kernel functions into the one-class SVM algorithm etc.

References

[1] J. Hu, "Host-based anomaly intrusion detection," in *Handbook of Information and Communication Security*, P. Stavroulakis and M. Stamp, Eds. Springer Berlin Heidelberg, 2010, pp. 235–255.

[2] [Online]. Available: http://www.cs.unm.edu/~immsec/systemcalls.htm

[3] [Online]. Available: http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/

[4] G. Creech and J. Hu, "Generation of a new ids test dataset: Time to retire the kdd collection," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 4487–4492.

[5] S. Forrest, S. Hofmeyr, and A. Somayaji, "The evolution of system-call monitoring," in *Computer Security Applications Conference, 2008. ACSAC 2008. Annual*. IEEE, 2008, pp. 418–430.

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

[7] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of computer security*, vol. 6, no. 3, pp. 151–180, 1998.

[8] A. P. Kosoresow and S. Hofmeyer, "Intrusion detection via system call traces," *Software, IEEE*, vol. 14, no. 5, pp. 35–42, 1997.

[9] E. Eskin, W. Lee, and S. J. Stolfo, "Modeling system calls for intrusion detection with dynamic window sizes," in *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, vol. 1. IEEE, 2001, pp. 165–175.

[10] X. D. Hoang and J. Hu, "An efficient hidden markov model training scheme for anomaly intrusion detection of server applications based on system calls," in *Networks, 2004.(ICON 2004). Proceedings. 12th IEEE International Conference on*, vol. 2. IEEE, 2004, pp. 470–474.

[11] X. D. Hoang, J. Hu, and P. Bertok, "A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference," *Journal of Network and Computer Applications*, vol. 32, no. 6, pp. 1219–1228, 2009.

[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, IEEE*, vol. 23, no. 1, pp. 42–47, 2009.

[13] A. K. Ghosh, J. Wanken, and F. Charron, "Detecting anomalous and unknown intrusions against programs," in *Computer Security Applications Conference, 1998. Proceedings. 14th Annual*. IEEE, 1998, pp. 259–267.

[14] A. K. Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior profiles for intrusion detection." in *Workshop on Intrusion Detection and Network Monitoring*, vol. 51462, 1999.

[15] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *Computers, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

[16] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of data mining in computer security*. Springer, 2002, pp. 77–101.

[17] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[18] M. Xie and J. Hu, "Evaluating host-based anomaly detection systems: A preliminary analysis of adfa-ld," in *6th International Congress on Image and Signal Processing (CISP), 2013*. IEEE, 2013.

[19] M. Xie, J. Hu, and B. Tian, "Histogram-based online anomaly detection in hierarchical wireless sensor networks," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. IEEE, 2012, pp. 751–759.

[20] M. Xie, J. Hu, S. Han, and H. Chen, "Scalable hyper-grid k-nn-based online anomaly detection in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 8, pp. 1661–1670, 2013.

[21] M. Xie, J. Hu, X. Yu, and E. Chang, "Evaluating host-based anomaly detection systems: Application of the frequency-based algorithms to adfa-ld," in *8th International Conference on Network and System Security (NSS), 2014*. Springer, 2014.

[22] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of svm and ann for intrusion detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.

[23] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.