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 roughly 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 attacks. It is believed that ADFA-LD will be a reliable substitute for those obsolete benchmark data sets.

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], artificial neural network (ANN) [12] [13], semantic data mining [14] and support vector machine (SVM) [15]. 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) [16] [17], clustering analysis [18] and SVM [19]. 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 [14]. 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’ [14]. 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
model can reduce the computational cost significantly, as presented in [17] [18]. 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 Audit daemon. In this paper, we present only a brief overview for ADFA-LD, as follows. During a 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-SSH, Hydra-FTP, Meterpreter, Java-Meterpreter, Adduser, 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>
<thead>
<tr>
<th>Trace Type</th>
<th>Number</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>833</td>
<td>normal</td>
</tr>
<tr>
<td>Validation</td>
<td>4373</td>
<td>normal</td>
</tr>
<tr>
<td>Hydra-FTP</td>
<td>162</td>
<td>attack</td>
</tr>
<tr>
<td>Hydra-SSH</td>
<td>148</td>
<td>attack</td>
</tr>
<tr>
<td>Adduser</td>
<td>91</td>
<td>attack</td>
</tr>
<tr>
<td>Java-Meterpreter</td>
<td>125</td>
<td>attack</td>
</tr>
<tr>
<td>Meterpreter</td>
<td>75</td>
<td>attack</td>
</tr>
<tr>
<td>Webshell</td>
<td>118</td>
<td>attack</td>
</tr>
</tbody>
</table>

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 training 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 will a negligible rate of collision, as follows. 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 $O(n \log(n))$. 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 $x^*$. Due to the page limit, we leave its rigourous proof to be given in future work.

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 $\mathbf{T}$ can be obtained by computing

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

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 [20] 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
The optimisation problem can be described as
\[
\min_{\omega, \rho \in \mathbb{R}} \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \zeta_i - \rho
\]
subject to \( (\omega \cdot \phi(x_i)) \geq \rho - \zeta_i, \zeta_i > 0 \)

where \( \omega \) is the normal vector of the hyperplane in the feature space, \( \rho \) 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 \( \zeta_i \) 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
\[
\min \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K_\phi(x_i, x_j)
\]
subject to \( 0 \leq \alpha_i \leq \frac{1}{\nu n}, \sum_{i=1}^{n} \alpha_i = 1 \)

where \( \alpha_i \) are the Lagrange multipliers and \( K_\phi \) 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
\[
f(y) = \text{sgn} ((w \cdot \phi(x_i)) - \rho) = \text{sgn} \left( \sum_{i=1}^{n} \alpha_i K_\phi(y, x_i) - \rho \right)
\]
where \( y \) is the test instance.

### TABLE II
**KERNEL FUNCTIONS**

<table>
<thead>
<tr>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear</td>
<td>( K_\phi(x_1, x_2) = x_1 x_2^T )</td>
</tr>
<tr>
<td>polynomial</td>
<td>( K_\phi(x_1, x_2) = (\gamma x_1 x_2^T + c_0)^d )</td>
</tr>
<tr>
<td>radial basis</td>
<td>( K_\phi(x_1, x_2) = \exp (-\gamma |x_1 - x_2|^2) )</td>
</tr>
<tr>
<td>sigmoid</td>
<td>( K_\phi(x_1, x_2) = \text{tanh}(\gamma x_1 x_2^T + c_0) )</td>
</tr>
</tbody>
</table>

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 \( \bar{T} \), i.e.,

\[
t = y_\bar{T} \quad \bar{y}_i = t \times f_t
\]

for \( i = 1, 2, \cdots, k \). It detects \( \bar{y} \) using the decision function, with ‘+1’ and ‘-1’ indicating normal and abnormal respectively. With regard to different processes, the number of the short sequences to be 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 \( \eta \). Assuming that the rate is denoted by \( r \), if \( r \geq \eta \), a trace will be eventually labelled as abnormal; otherwise, normal.

### IV. NUMERICAL EXPERIMENTS

The numerical experiments are implemented using LIBSVM V3.18 [21] 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 \( \eta \); and (5) how much time the entire detection algorithm costs in practice.

The training data set \( \bar{T} \) 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 \( \eta \). 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**

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \nu )</th>
<th>( \gamma )</th>
<th>( c_0 )</th>
<th>( d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.35</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.05</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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
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 [14]. \( \eta \) 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

\[ \eta \approx 0.1 \]

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, as previously mentioned, a rigorous proof of the elimination method is required, to guarantee that information will not be lost during this procedure. Second, more advanced strategies may be utilized to weight the short sequences for maximising the separability between normal and abnormal. Third, some details may be further sharpened, such as the criteria of selecting \( k \), introduction of user-defined kernel functions into the one-class SVM algorithm etc.

**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, as previously mentioned, a rigorous proof of the elimination method is required, to guarantee that information will not be lost during this procedure. Second, more advanced strategies may be utilised to weight the short sequences for maximising the separability between normal and abnormal. Third, some details may be further sharpened, such as the criteria of selecting \( k \), introduction of user-defined kernel functions into the one-class SVM algorithm etc.

**References**


