Combining Hidden Markov Models for Improved Anomaly Detection

Wael Khreich, Eric Granger and Robert Sabourin
Laboratoire d’imagerie, de vision et d’intelligence artificielle (LIVIA), école de technologie supérieure, Montreal, QC, Canada, wael.khreich@livia.etsmtl.ca, {eric.granger, robert.sabourin}@etsmtl.ca

Ali Miri
School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, ON, Canada, samiri@site.uottawa.ca

Abstract—In host-based intrusion detection systems (HIDS), anomaly detection involves monitoring for significant deviations from normal system behavior. Hidden Markov Models (HMMs) have been shown to provide a high level performance for detecting anomalies in sequences of system calls to the operating system kernel. Although the number of hidden states is a critical parameter for HMM performance, it is often chosen heuristically or empirically, by selecting the single value that provides the best performance on training data. However, this single best HMM does not typically provide a high level of performance over the entire detection space. This paper presents a multiple-HMMs approach, where each HMM is trained using a different number of hidden states, and where HMM responses are combined in the Receiver Operating Characteristics (ROC) space according to the Maximum Realizable ROC (MRROC) technique. The performance of this approach is compared favorably to that of a single best HMM and to a traditional sequence matching technique called STIDE, using different synthetic HIDS data sets. Results indicate that this approach provides a higher level of performance over a wide range of training set sizes with various alphabet sizes and irregularity indices, and different anomaly sizes, without a significant computational and storage overhead.

Index Terms—Anomaly Detection, Host Based Intrusion Detection Systems, Hidden Markov Models, Multi-Classifier Systems

I. INTRODUCTION

Intrusion Detection Systems (IDS) is used to identify, assess, and report unauthorized computer or network activities. Host-based IDSs (HIDS) are designed to monitor the host system activities and state, while network-based IDSs monitor network traffic for multiple hosts. In either case, IDSs have been designed to perform misuse detection (looking for events that match patterns corresponding to known attacks) and anomaly detection (detecting significant deviations from normal system behavior). In a HIDS for anomaly detection, operating system events are usually monitored. Since system calls are the gateway between user and kernel mode, most traditional host-based anomaly detection systems monitor deviation in system call sequences.

In general, anomaly detection systems seek to detect novel attacks, yet will typically generate false alarms mainly due to incomplete data for training, poor modeling, and difficulty in obtaining representative labeled data for validation. In practice, it is very difficult to acquire (collect and label) comprehensive data sets to design a HIDS for anomaly detection. Forrest et al. [1] confirmed that short sequences of system calls are consistent with normal operation, and unusual burst will occur during an attack. Their anomaly detection system, called Sequence Time-Delay Embedding (STIDE), is based on segmenting and enumerating the training data into fixed-length contiguous sequences, using a fixed-size sliding window, shifted by one symbol. During operations, the sliding window scheme is used to scan the data for anomalies — sequences that are not found in the normal data.

Various anomaly detection techniques have been applied to learn the normal process behavior through system call sequences [2]. Among these, techniques based on discrete Hidden Markov Models (HMMs) [3] have been shown to produce slightly superior results, at the expense of training resource requirements. Estimating the parameters of an HMM requires the choice of the number of hidden states. In the literature on HMMs applied to anomaly detection [4]–[7], the impact of the number of states on performance is often overlooked. It is typically chosen heuristically or empirically by selecting the number of states that provides the best performance on training data. Given incomplete training data, types of anomalies, and detector window (DW) sizes, a single “best” HMM does not provide a high level of performance over the entire detection space.

For a given data set, a multiple-HMMs ($\mu$-HMMs) approach, where each model is trained using a different number of hidden states, allows to outperform any single best HMM. The responses of these HMMs are combined in the receiver operating characteristics (ROC) space, using the Maximum Realizable ROC (MRROC). This approach allows selecting operational HMMs independently from prior and class distributions, and cost contexts. Although the MRROC fusion has been applied to combine responses of different classifiers in fields such as medical diagnostics and image segmentation [8], it has never appeared in HIDS literature. In particular, it has never been proposed in HMM application to anomaly detection due to the overlooked impact on performance of the number of HMM states, and to its expensive training resource requirements, which are emphasized in this work. However, other fusion techniques such as weighted voting has been applied to combine HMMs trained on different features [9].

The objective of this paper consists in comparing the perfor-
mance of $\mu$-HMMs combined through the MRROC fusion, to that of a single best HMM and STIDE for detecting anomalies in system calls sequences. The impact on performance of using different training set sizes, anomaly sizes, and complexities of the monitored processes is assessed using ROC analysis [10]. To this end, a synthetic simulator for normal data generation and anomaly injection has been constructed to overcome the issues encountered when experimenting with real data.

The rest of this paper is organized as follows. The next section describes the application of HMMs in anomaly-based HIDS. In Section III, the proposed $\mu$-HMMs approach is presented. Then, the experimental methodology in Section IV describes data generation, evaluation methods and performance metrics. Finally, simulation results are presented and discussed in Section V.

II. ANOMALY DETECTION WITH HMM

A discrete-time finite-state HMM is a stochastic process determined by the two interrelated mechanisms – a latent Markov chain having a finite number of states, and a set of observation probability distributions, each one associated with a state. At each discrete time instant, the process is assumed to be in a state, and an observation is generated by the probability distribution corresponding to the current state. HMMs are usually trained using the Baum-Welch algorithm [11] – a specialized expectation maximization technique to estimate the parameters of the model from the training data. Theoretical and empirical results have shown that, given an adequate number of states and a sufficiently rich set of observations, HMMs are capable of representing probability distributions corresponding to complex real-world phenomena in terms of simple and compact models. Given the trained model, the Forward algorithm [11] can be used to evaluate the likelihood of the test sequence. For further details regarding HMM the reader is referred to the extensive literature e.g., [3].

Estimating the parameters of a HMM requires the specification of the number of hidden states $(N)$. Warrender et al. [2] present the first application to anomaly-based HIDS with the University of New Mexico (UNM) data sets\(^1\). The authors trained an ergodic HMM on the normal data sequence for each process of the UNM data set. The number of states was selected heuristically. It is set roughly equal to the alphabet size – the number of unique system call symbols used by the process. Each HMM is then tested on the anomalous sequences, looking for unusual state transitions and/or symbol outputs according to a predefined threshold. Compared to other techniques, HMM produced slightly superior results in terms of average detection and false alarm rate, at the expenses of expensive training resource requirements. Indeed, the time complexity of the Baum-Welch algorithm per iteration scales linearly with the sequence length and quadratically with the number of states. In addition, its memory complexity scales linearly with both sequence length and number of states.

Subsequent applications of HMMs to anomaly-based HIDS (e.g., [4]–[7]) follow Warrender in choosing the number of HMM states without further optimization. The authors in [12] are among few who tried to investigate the effect of the number of states and the detector window size on the performance using UNM data sets. However only unique contiguous sequences are used for HMM training, which might affect the model parameters. Furthermore, as explained in Section IV, the data sets are labeled using the STIDE matching technique which makes it as the ground truth for other classifiers.

III. SELECTION AND FUSION OF HMMs IN THE ROC SPACE

The ROC curve displays a trade-off between true detection (or true positive) rate and false alarm (or false positive) rate for all thresholds. In our context, the true detection rate is the number of anomalous sequences correctly detected over the total number of anomalous sequences, and the false alarm rate is the number of normal sequences detected as anomalous over the total number of normal sequences in the test set.

ROC analysis provides a useful tool for combining classifiers given a set of one- or two-class classifiers using the MRROC [8]. The MRROC is a fusion technique that uses randomized decision rules and provides an overall classification system whose curve is the convex hull over all existing detection and false alarm rates. In contrast with traditional ROCs where only operational thresholds change when moving along the curve, the MRROC points may also change other model parameters or even switch models. All operating points on the MRROC are achievable in practice, using an exhaustive search, and are theoretically proven by application of the realizable classifier theory [8]. The procedure is to create from two existing classifiers $C_a$ and $C_b$ a composite one $C_c$ whose performance in terms of its ROC, lies on the line segment connecting $C_a$ and $C_b$. This is done by interpolating the responses of the existing classifiers as follows. Let $(tpr_a, fpr_a)$ and $(tpr_b, fpr_b)$ be the true positive and false positive rates of the existing classifiers $C_a$ and $C_b$, and $(tpr_{ca}, fpr_{ca})$ those of the desired composite classifier $C_c$. The output of $C_c$ for any input $O = \{o_1, \ldots, o_T\}$ from the test set, is a random variable that selects the output of $C_a$ or $C_b$ with probability:

$$\Pr(C_c = C_a) = \frac{fpr_{ca} - fpr_a}{fpr_b - fpr_a} \quad (1)$$

$$\Pr(C_c = C_b) = 1 - \Pr(C_c = C_a)$$

Instead of training the HMM with an arbitrary chosen number of states, our approach consists in training several HMMs over a range of $n = [N_{min}, N_{max}]$ values and combine their decisions in the ROC space using the MRROC. HMMs trained with various number of states capture different temporal structures of the data. As illustrated in Figure 1, for a desired operational system $C_c$ the fusion is done by switching at random between the responses of neighboring models $C_a$ and $C_b$ on the convex hull curve according to eq. (1). This combination leads to the realization of the virtual desired model $C_c$, which is able to achieve in the least a higher performance than any existing models. Since only the HMMs touching the MRROC are potentially optimal, no others need be retained. In addition, this approach allows visualizing systems’ performance and selecting operational

---

\(^1\)http://www.cs.unm.edu/~immsec/systemcalls.htm
HMMs independently from both prior and class distributions as well as cost contexts [10]. As conditions change, the MRROC does not change; only the portion of interest will. This change is accounted for by shifting the operational system to another point on the facets of the MRROC. In contrast with the commonly used anomaly index measure in related work, this approach also permits to use the Area Under the ROC Curve (AUC), which has been proven an effective evaluation measure in contexts with variable misclassification costs or skewed data [13].

$$\text{Disregarded HMM}$$

![Illustration of the $\mu$-HMMs using the MRROC fusion.](image)

The time and memory complexity required for Baum-Welch training of $n$ different HMMs for the $\mu$-HMMs is comparable to that of training an HMM over a range of $n$ different $N$ values, and selecting the single best HMM (i.e., $O(nN.DW^2)$). During operations, the worst-case time complexity of the Forward-Backward algorithm to evaluate a sequence of data (for a given detection and false alarm rate) with the $\mu$-HMM solution is two times of the single best HMM solution (i.e., $O(N.DW^2)$). For a specific detection and false alarm rate, the MRROC requires in the worst-case the responses of two HMMs to interpolate between two convex hull facets in the ROC space. The worst-case memory complexity required to store HMM parameters may be significantly higher than that of a single best HMM. However, HMMs with a ROC curve that does not touch the MRROC are suboptimal and may be discarded.

IV. EXPERIMENTAL METHODOLOGY

The UNM data sets are commonly used for benchmarking anomaly detections based on system calls sequences. Normal data are collected from the monitored process in a secured environment, while testing data are the collection of the system calls when this process is under attack [2]. Since it is very difficult to isolate the manifestation of an attack at the system call level, the UNM test sets are not labeled. Therefore, in related work, intrusive sequences are usually labeled by comparing normal sequences, using STIDE matching technique. This labeling process considers STIDE response the ground truth, and leads to a biased evaluation and comparison of techniques, which depends on both training data size and detector window size.

The need to overcome issues encountered when using real-world data for anomaly-based HIDS (incomplete data for training, and labeled data) has lead to the implementation of a synthetic data generation platform for proof-of-concept simulations. It is intended to provide normal data for training and labeled data (normal and anomalous) for testing. This is done by simulating different processes with various complexities then injecting anomalies in known locations.

Inspired by the work of Tan and Maxion [14]–[16], the data generator is based on the Conditional Relative Entropy (CRE) of a source. It is defined as the conditional entropy divided by the maximum entropy ($\text{MaxEnt}$) of that source, which gives an irreversibility index to the generated data. For two random variables $x$ and $y$ the CRE can be computed by:

$$\text{CRE} = \frac{-\sum_x p(x) \sum_{y|x} p(y | x) \log p(y | x)}{\text{MaxEnt}}$$

where for an alphabet of size $\Sigma$ symbols, $\text{MaxEnt} = -\Sigma \log(1/\Sigma)$ is the entropy of a theoretical source in which all symbols are equiprobable. It normalizes the conditional entropy values between $\text{CRE} = 0$ (perfect regularity) and $\text{CRE} = 1$ (complete irregularity or random). In a sequence of system calls, the conditional probability, $p(y | x)$, represents the probability of the next system call given the current one. It can be represented as the columns and rows (respectively) of a Markov Model with the transition matrix $M = \{a_{ij}\}$, where $a_{ij} = p(S_{t+i} = j | S_t = i)$ is the transition probability from state $i$ at time $t$ to state $j$ at time $t+1$. Accordingly, for a specific alphabet size $\Sigma$ and $\text{CRE}$ value, a Markov chain is first constructed, then used as a generative model for normal data. This Markov chain is also used for labeling injected anomalies as described below. Let an anomalous event be defined as a surprising event which does not belong to the process normal pattern. This type of event may be a foreign-symbol anomaly sequence that contains symbols not included in the process normal alphabet, a foreign $n$-gram anomaly sequence that contains n-grams not present in the process normal data, or a rare $n$-gram anomaly sequence that contains n-grams that are infrequent in the process normal data and occurs in burst during the test.

Generating training data consists of constructing Markov transition matrices for an alphabet of size $\Sigma$ symbols with the desired irregularity index ($\text{CRE}$) for the normal sequences. The normal data sequence with the desired length is then produced with the Markov chain, and segmented using a sliding window (shift one) of a fixed size, $DW$. To produce the anomalous data, a random sequence ($\text{CRE} = 1$) is generated, using the same alphabet size $\Sigma$, and segmented into subsequences of a desired length using a sliding window with a fixed size of $AS$. Then, the original generative Markov chain is used to compute the likelihood of each sub-sequence. If the likelihood is lower than a threshold it is labeled as anomaly. The threshold is set to $(\min(a_{ij}))^{AS-1}$, $\forall i, j$, the minimal value in the Markov transition matrix to the power ($AS - 1$), which is the number of symbol transitions in the sequence of size $AS$. This ensures that the anomalous sequences of size $AS$ are not associated with the process normal behavior.

2This is in contrast with other work which consider rare event as anomalies. Rare events are normal, however they may be suspicious if they occurs in high frequency over a short period of time.
and hence foreign n-gram anomalies are collected. The trivial case of foreign-symbol anomaly is disregarded since it is easy to be detected. Rare n-gram anomalies are not considered since we seek to investigate the performance at the detection level, and such kind of anomalies are accounted for at a higher level by computing the frequency of rare events over a local region. Finally, to create the testing data another normal sequence is generated, segmented and labeled as normal. The collected anomalies of the same length are then injected into this sequence at random according to a mixing ratio.

The experiments conducted in this paper cover a wide range of the parameters space: $\Sigma = 8 - 50$ symbols, training set size $= 100 - 1000,000$ symbols and $CRE = 0.3 - 0.8$, to approach as much as possible to real-world processes [17]. The sizes of injected anomalies are assumed equal to the detector window size $AS = DW = \{2, 4, 6, 8\}$, and different normal/anomalous ratios are considered for the test phase.

For each training set of size $DW$, different discrete-time ergodic HMMs are trained with various number of hidden states $N$. The number of symbols is taken equal to the process alphabet size. The iterative Baum-Welch algorithm is used to estimate HMM parameters [11]. To reduce overfitting effects, the evaluation of the log-likelihood on an independent validation set is used as a stopping criterion. The training process is repeated ten times using a different random initialization to avoid local minima. Finally, the model that gives the highest log-likelihood value on validation data is selected. This procedure is replicated ten times with different training, validation and testing sets, and the results are averaged and presented along with the standard deviations.

After training an HMM, the log-likelihood of a test sub-sequences is evaluated using the forward algorithm [3]. By sweeping all the decision thresholds, HMM evaluation results in an ROC curve for each value of $N$. In contrast, STIDE testing consists of comparing the test sub-sequences with its normal database, which gives a point in the ROC space. It should be noted that since in this paper, the detector window size is always considered equal to the anomaly size ($DW = AS$), STIDE detection rate is always 100% and only the false alarm rate varies. This is due to its blind regions – STIDE only misses anomalous sequences that are larger that the detector window size ($DW < AS$) and have all of its sub-sequences in STIDE normal database. The window will slide on its sub-sequences that are all normal, without being able to discover that the whole sequence is anomalous [14]-[16].

V. SIMULATION RESULTS

Due to space limitations only a limited number of results are presented. First, the results of a simpler scenario are illustrated in Figure 2 and 3, and then those of a more complex scenario are shown in Figure 4. The simpler scenario involves a relatively simple case with a small alphabet size $\Sigma = 8$ and a low irregularity index $CRE = 0.3$, while for the more complex scenario $\Sigma = 50$ symbols and the $CRE = 0.4$. For both scenarios, the presented results are for test sets that comprise 75% of normal and 25% of anomalous data. However, these results are consistent throughout the range of experiments not shown in this paper.

The ROC curves in Figure 2 show the impact of using different training set and anomaly sizes on the performance of STIDE and HMM, where HMMs are trained with different numbers of states $N$. Results in this figure indicate that the AUC of HMM grow with the training set size. As illustrated in Figure 2c, when the anomaly size increases, the search space grows exponentially (according to $\Sigma^{AS}$) which also expands both anomalous and normal partitions of the space. For instance, for $\Sigma = 8$ and $AS = 2$ the space of combinations comprises $8^2 = 64$ sequences, part of which is normal and the rest anomalous (depending on the $CRE$ the data). By changing the anomaly size to $AS = 8$ the combinations grows to $8^8 = 16,777,216$ sequences, which imposes a much larger amount of training data for STIDE to memorize the normal space. The storage capacity and detection time associated with STIDE increases considerably. Accordingly, for a fixed training set size, the performance of STIDE is significantly degraded with the increase of the anomaly size. Nevertheless, any normal sequence that was not collected during training will be classified as anomaly, triggering false alarms in operational mode. This problem is usually mitigated by introducing an anomaly counter or index in local regions according to an arbitrary threshold, which may however be exploited by an adversary.

In contrast, even when an HMM is trained on a relatively small data set, its discrimination capabilities increases with the anomaly size. This is due to the HMM abilities to capture dependencies in temporal patterns and to generalize in the case of unknown test sequences. Indeed, when evaluating the likelihood $Pr(O \mid HMM)$, an HMM computes the product of probabilities (over states transitions and emissions) for symbols in the sequence presented for testing. Anomalous sequences should lead to significantly lower likelihood values than normal ones with a well trained HMM. Therefore, with the increase of the anomaly size, the likelihood of anomalous sequences becomes smaller at a faster rate than normal ones, and hence increases HMM detection rate.

The impact of the number of HMM states $N$ on performance is rarely addressed in the HIDS applications. Figures 2 and 3 illustrates this effect on AUC measures versus different amounts of training data and different detector window sizes $DW$. It can be seen that the value of $N$ that provides the best performance depends on the training data set size and $DW$. More importantly, the common practice (in most application to HMMs for anomaly-based HIDS) of selecting the number of states equal to the alphabet size (e.g., $N = \Sigma = 8$ in Figure 3), does not provide a high level of performance over the wide range of training set sizes and $DW$ values. This effect is further illustrated in Figure 4 with the average AUC values (over ten independent replications) for the more complex scenario.

The optimal number of states does not exist over the whole range of detection space to design a “single best” HMM. Combining the responses of $\mu$-HMMs with the MRROC fusion is shown to achieve the highest over all performance over anyone of the HMMs alone. Furthermore, the resulting convex hull (from the MRROC combination) provides a smoother curve than any existing single model. The operational system
may therefore be changed according to the desired detection and false alarm rate without compromising the performance. In fact, moving along the convex hull curve allows switching to another model or interpolating the responses of two models. In contrast, as illustrated in Figure 2a and b, using a single HMM typically results a staircase-shaped ROC curve and this effect is even worse in real-world cases since the anomalies are relatively rare with reference to normal data.

Finally, in contrast with STIDE, HMM and the \(\mu\)-HMMs are capable of detecting different anomaly sizes with the same detector window size. This impact along with the combination of HMMs trained with different window sizes is being characterized and will appear in future work.

This paper presents a multiple-HMMs approach, where each HMM is trained using a different number of hidden states capturing different temporal structures of the data. The HMM responses are then combined in the ROC space according to the MRROC technique. Results have shown that the overall performance of the proposed \(\mu\)-HMMs approach increased considerably over a single best HMM and STIDE. In addition, this combination provides a smooth convex hull for composite operational systems without significantly increasing the computational overhead. Future work also involves characterizing the performance of the \(\mu\)-HMMs approach using for instance the UNM data and observing the impact of combining HMMs trained with different window sizes to detect various anomaly sizes.

VI. CONCLUSIONS

This paper presents a multiple-HMMs approach, where each HMM is trained using a different number of hidden states capturing different temporal structures of the data. The HMM responses are then combined in the ROC space according to the MRROC technique. Results have shown that the overall performance of the proposed \(\mu\)-HMMs approach increased considerably over a single best HMM and STIDE. In addition, this combination provides a smooth convex hull for composite operational systems without significantly increasing the computational overhead. Future work also involves characterizing the performance of the \(\mu\)-HMMs approach using for instance the UNM data and observing the impact of combining HMMs trained with different window sizes to detect various anomaly sizes.

REFERENCES

Fig. 3. μ-HMMs vs HMM performance evaluation using the AUC for various training set sizes, DW and N (Σ = 8 and CRE = 0.3).

Fig. 4. Average AUC of single HMMs and the μ-HMMs with MRROC combination for various training set sizes, DW and N values (Σ = 50 and CRE = 0.4). Error bars are standard deviations.