Potential Malicious Users Discrimination with Time Series Behavior Analysis

Murat Semerci MURAT.SEMERCI@BOUN.EDU.TR
Ali Taylan Cemgil TAYLAN.CEMGIL@BOUN.EDU.TR
Department of Computer Engineering, Bogazici University, 34342, Bebek, Istanbul, Turkey

Bulent Sankur BULENT.SANKUR@BOUN.EDU.TR
Department of Electrical & Electronics Engineering, Bogazici University, 34342, Bebek, Istanbul, Turkey

Abstract

Discriminating the malicious users in a network is crucial in protecting the network entities and preventing any ongoing attacks. In an organized attack, a group users are supposed to behave synchronously in the same manner. In this study, we particularly focus on organized attacks where the attackers create a high volume of requests to overwhelm the server under heavy resource consumption. We propose a novel behavior analysis based on the time series alignment kernel and spectral clustering to determine the group of users that concurrently perform similar behaviors (or dissimilar behavior to that of innocent users). We experiment the proposed model on the simulated data.

1. Introduction

Analysis of time series for classification, prediction, change and outlier detection has been active research topics for decades with particular focus on financial markets (Gupta et al., 2014). Among the plethora of methods proposed one can mention: i) methods that map the time series into a new feature space, such as spectral entropy, autocorrelation etc. (Hyndman et al., 2015); ii) kernel methods for time-series classification with emphasis on sequence alignment (Cuturi, 2011; Sivaramakrishnan et al., 2007; Chen et al., 2013); iii) clustering time series with a combined distance function of triangle similarity and dynamic time warping distance (Zhang et al., 2011); iv) approaches fitting the data to a number of possible models, such as hidden Markov models and autoregressive moving average, and clustering the data based on model instance with the best fit (Oates et al., 1999; Xiong & Yeung, 2002); v) singular spectral analysis where data is embedded, the embedding matrix decomposed and reconstructed into trend, noise and oscillatory components.

In this study, our goal is to build an anomaly detector that analyzes the behavior of multiple time series in order to discriminate the set of malicious users in a cyber attack scenario. The underlying assumption is that in a cyber attack, the attacker group would show correlated behavior patterns and act in a concurrent manner (e.g. botnet attacks). To this purpose, we first develop a measure of user behavior similarity based on the types and timings of their actions. Then, we use spectral clustering techniques to discriminate the malicious users from the ordinary ones.

The paper is organized as follows: Section 2 discusses the method to align two sequences of different length, presents our sequence alignment kernel and defines the pairwise heat kernel. Section 3 elaborates on the discriminate the users. In Section 4, a heuristics for automatic selection of the attacker group is proposed, and the complete algorithm is given. Experiments results and discussion of future research directions are given in Section 5 and Section 6, respectively.

2. Sequence Alignment and the Proposed Kernel

One way to express the similarity between two sequences of possibly different length is by the sum of similarities of all their possible alignments, with no pair repetition. In this method, while proceeding in the alignments, we allow either only one member to vary or both of the members to vary simultaneously. Figure 1 shows an example of all possible alignments for two sequences, which, respectively, have lengths two and three. In the specific example there are 5 possible alignments. Thus, the similarity of two sequences will be higher if they have higher member-pairwise similarity values. In our work the sequences correspond to the message sent by the terminals, characterized

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

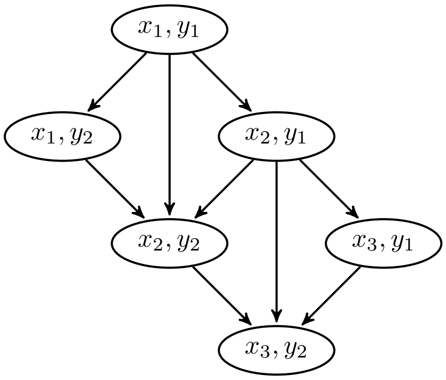

Figure 1. All possible alignments for $x = \{x_1, x_2, x_3\}$ and $y = \{y_1, y_2\}$, where x_i and y_j are denoted as members.

by their types and time stamps.

- $(x_1, y_1), (x_1, y_2), (x_2, y_2), (x_3, y_2)$
- $(x_1, y_1), (x_2, y_2), (x_3, y_2)$
- $(x_1, y_1), (x_2, y_1), (x_2, y_2), (x_3, y_2)$
- $(x_1, y_1), (x_2, y_1), (x_3, y_2)$
- $(x_1, y_1), (x_2, y_1), (x_3, y_1), (x_3, y_2)$

The similarity of two sequences is a function of pairwise similarity of sequence members. Thus, the similarity of two sequences is higher if they have high overall pairwise similarity values.

A global alignment kernel has been proposed in (Cuturi et al., 2007), which uses dynamic programming to compute the similarity of all possible alignments of two sequences. We use a variation of this algorithm as detailed in Algorithm 1, where we employ a pairwise heat kernel that is based on the Mahalanobis distance and differences of time stamps. The sequence alignment kernel for two sequences of length, respectively, m and n, is given as $k(\mathbf{X}, \mathbf{Y}) = T_{n+1,m+1}$. This kernel considers all possible alignments of time stamps of the two sequences, sums these terms weighted by the pairwise kernel function, $\kappa(\mathbf{x}_i, \mathbf{y}_j)$. Here $\kappa(\mathbf{x}_i, \mathbf{y}_j)$, measures the similarity of two time stamped message events $\mathbf{x}_i$ and $\mathbf{y}_j$, and each such vector incorporates the qualifier for the type of message one of the d possible message types) and the time of arrival of the message. After the kernel matrix for all user pairs is obtained, we unit-diagonal normalize the kernel matrix in order to eliminate scaling issues:

$$k'(\mathbf{X}, \mathbf{Y}) = \frac{k(\mathbf{X}, \mathbf{Y})}{\sqrt{k(\mathbf{X}, \mathbf{X})}\sqrt{k(\mathbf{Y}, \mathbf{Y})}} \rightarrow k'(\mathbf{X}, \mathbf{Y}) \in [0, 1] \tag{1}$$

Algorithm 1 Global Alignment Kernel

```
For any two given user behavior sequences $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, $\mathbf{X} \in \Re^{(d+1)\times n}$ and $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_m)$, $\mathbf{Y} \in \Re^{(d+1)\times m}$ in a state space $\Psi$, create $\mathbf{T} \in \Re^{(n+1)\times(m+1)}$.
Set the members in the first row and in the first column of $\mathbf{T}$ to zero ($T_{1,:} = T_{:,1} = 0$). Set $T_{1,1} = 1$.
for $i = 2$ to $n + 1$ do
    for $j = 2$ to $m + 1$ do
        $T_{i,j} = (T_{i,j-1} + T_{i-1,j-1} + T_{i-1,j})\kappa(\mathbf{x}_{i-1}, \mathbf{y}_{j-1})$
    end for
end for
Return $T_{n+1,m+1}$
```

After this normalization, similar behaving user pairs get values close to 1 and the dissimilar pairs get values close to 0.

Each users network actions are characterized by the multi-time series, each series corresponding to the ordered time-stamped sequence of one type of message. At any one time at most one type of message can be sent. Thus, for a message vector $\mathbf{x}' = [x'_1, x'_2, \ldots, x'_d]^\top$, $x'_s \in \{0, 1\}$ and $\sum_s x'_s = 1$, where d is the number of message types. Within an observation interval, users can send arbitrary number of different messages so that messaging event sequences can have different lengths.

A kernel function (the pairwise heat function) for any two time-stamped message vector, $\mathbf{x}_i = (\mathbf{x}'_i, t_{\mathbf{x}'_i})$ and $\mathbf{y}_j = (\mathbf{y}'_j, t_{\mathbf{y}'_j})$ is evaluated as:

$$\kappa((\mathbf{x}'_i, t_{\mathbf{x}'_i}), (\mathbf{y}'_j, t_{\mathbf{y}'_j})) = exp(-\gamma D(\mathbf{x}'_i, \mathbf{y}'_j) - \rho|t_{\mathbf{x}'_i} - t_{\mathbf{y}'_j}|) \tag{2}$$

where $D(\mathbf{x}'_i, \mathbf{y}'_j)$ is a distance function such as Euclidean or Mahalanobis distance. The squared Mahalanobis distance for a given Mahalanobis matrix, $\mathbf{M}$, can be evaluated as follows:

$$D(\mathbf{x}'_i, \mathbf{y}'_j) = (\mathbf{x}'_i - \mathbf{y}'_j)^\top \mathbf{M} (\mathbf{x}'_i - \mathbf{y}'_j) \tag{3}$$

Note that $\kappa((\mathbf{x}'_i, t_{\mathbf{x}'_i}), (\mathbf{y}'_j, t_{\mathbf{y}'_j})) = 1$ iff $\mathbf{x}'_i == \mathbf{y}'_j$ and $t'_{\mathbf{x}_i} == t'_{\mathbf{y}_j}$.

3. Spectral Clustering

A similarity matrix is created of the users from the pairwise user-to-user similarities as in Equation 1. The similarity matrix then corresponds to a weighted adjacency graph. In order to partition this graph such that users with similar messaging behavior are collected in the same sub-graph. To this effect, we have used a spectral clustering algorithm. Such algorithms are conceived to realize graph partitioning solutions in clustering problems, and in the literature there are various spectral clustering algorithms (Luxburg, 2007).

We have applied the normalized spectral clustering algorithm over the kernel matrix obtained from the user behaviors, $\mathbf{K}'$ (a $|U| \times |U|$ matrix where U is the set of users). This matrix is also interpreted as the weighted adjacency matrix (Shi & Malik, 2000) of a fully connected graph. Each user is a vertex in this graph while the edge between two users is their messaging behavior similarity. We aim to partition the graph into two sub-graphs such that the malicious users, typically synchronized and organized, fall into one cluster, and the rest is in the other cluster. The degree of l^{th} user is evaluated as:

$$d_l = \sum_{u=1}^{|U|} \mathbf{K}'(l, u) \tag{4}$$

The degree matrix $\mathbf{D}$ is a diagonal matrix whose diagonal elements consists of degree values, $d_1, d_2, \ldots, d_{|U|}$. The Laplacian matrix is evaluated as in Equation 5 and spectral clustering algorithm is given in Algorithm 2.

$$\mathbf{L} = \mathbf{D} - \mathbf{K}' \tag{5}$$

Algorithm 2 Normalized Laplacian Spectral Clustering

```
Given $\mathbf{K}'$, evaluate $\mathbf{D}$ and $\mathbf{L}$, which are all in $\Re^{|U|\times|U|}$.
Compute the first two eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ of the two smallest eigenvalues $0 = \lambda_1 < \lambda_2$ for the generalized eigenproblem $\mathbf{Lv} = \lambda\mathbf{Dv}$.
Augment $\mathbf{v}_1$ and $\mathbf{v}_2$ to obtain $\mathbf{V} \in \Re^{|U|\times 2}$. Use the rows of $\mathbf{V}$ as the new feature vectors in the mapped space, $\mathbf{y} \in \Re^2$. Apply $k$-means clustering with $k = 2$.
Return the cluster label vector $\mathbf{C}$ from $k$-means clustering.
```

4. Assignment of Malicious Users to a Cluster

We intend to cluster the users into two sets: Potentially malicious users, characterized by repetitive and correlated behaviors, and the rest of users, characterized by uncoordinated and diverse behaviors. Once the two clusters are formed, then the final task is that of determining that of attackers, for which we use a heuristic algorithm.

For each of the two clusters, we compute the sample co-variance matrix of the user message sequence vectors in that cluster. Recall that the elements of vectors consists of the message types and their time stamps. Since the malicious user cluster is assumed to consist of similar messaging behaviors, such message vectors are expected to be more strongly aligned along a few particular axes. In fact, in the extreme case when all messages in the cluster are of the same type and are perfectly synchronized, the sample co-variance matrix would be the 0 matrix. Therefore, we assign the cluster with significantly higher eigenvalue concentration to malicious users. This algorithm based on the heuristics that malicious users must be somewhat coordinated to mount an attack and therefore the data vectors must concentrate along a few eigenvectors is given in Algorithm 3.

Algorithm 3 Cluster Selection Heuristics

```
For the given cluster label vector $\mathbf{C}$, determine the two clusters, $C_1$ and $C_2$.
For the two clusters, evaluate the sample co-variance matrix of message vectors.
if a cluster has $\mathbf{0}$ co-variance matrix then
    Return the cluster with $\mathbf{0}$ co-variance matrix.
else
    Evaluate the eigenvalues of the cluster co-variance matrices.
    For each matrix, evaluate the proportional eigenvalues with respect to the total sum of eigenvalues.
    Return the cluster with the highest maximum proportional eigenvalue.
end if
```

Putting all of these steps together, the algorithm to discriminate the malicious users is summarized in Algorithm 4..

Algorithm 4 Potential Malicious Users Discrimination

```
Set the weight parameters $\gamma$ and $\rho$ of the pairwise heat kernel.
Evaluate the kernel matrix $\mathbf{K}$ such that $\forall\ (\mathbf{U}_i, \mathbf{U}_j) \in U \times U$, we have $\mathbf{K}_{i,j} = k(\mathbf{U}_i, \mathbf{U}_j)$ as defined in Algorithm 1, where $\mathbf{U}_i$, $\mathbf{U}_j$ are the time-stamped message sequences of $i^{th}$ and $j^{th}$ users, respectively and $U$ is the set of all user sequences.
Unit-diagonal normalize $\mathbf{K}$ to obtain $\mathbf{K}'$.
Apply symmetric normalized Laplacian spectral clustering over $\mathbf{K}'$ such that # clusters = 2, as defined in Algorithm 2.
Use cluster label vector $\mathbf{C}$ returned by spectral clustering in clustering selection heuristic as defined in Algorithm 3.
Return the cluster selected by the heuristics as the set of malicious users.
```

5. Experiments

To evaluate the performance of the proposed algorithm we implement a simulation setup, where we create variable length sequences for the normal and malicious users (attackers). In our setup, we assume that in any instance of attack, the attackers choose and send one type of message, and furthermore their timings are very close. We use the Mahalanobis distance in Equation 3, which is calculated as the inverse of the sample co-variance matrix of the number of messages observed within a time interval ($\mathbf{M} = \mathbf{\Sigma}^{-1}$).

In the simulations, the system is sampled at 1 second intervals. Users can choose from 5 different types of messages and they can send from 1 up to 5 messages of any type in a 1 second interval. We have set the attack strength to three levels in terms of the number of messages an attacker can send: Low-level (3-10), mid-level (5-10) and high-level (10-15). The simulation environment is dimensioned for 100 users in the system.

We have demonstrated the performance of the malicious user detection system as a function of variations in the attack duration, the proportion of attackers and the magnitude of attack. The attack duration is represented as at which interval quarter the attack starts. The attack can start at time $0.0; 0.25; 0.5; 0.75$ second within a time interval of 1 second.

Since we know the labels in the simulated data, we can show the performance of the proposed system in terms F-Measure. The ideal case would be when F-measure is 1, which can be obtained only when there is no falsely accused attackers (i.e., P = 1) and all the attackers are identified.

$$\begin{aligned}
\text{Precision (P)} &= \frac{\text{\# assigned true attackers}}{\text{\# assigned attackers}} && (6)\\
\text{Recall (R)} &= \frac{\text{\# assigned true attackers}}{\text{\# all attackers}} && (7)\\
\text{F-Measure (F)} &= 2\frac{P \times R}{P + R} && (8)
\end{aligned}$$

Figure 2 shows an example where our algorithm has successfully detected the malicious user groups according to the computed kernel. The uppermost sub-figure is the ground-truth label matrix ($K_{i,j} = 1$ if and only if both u_i and u_j are attackers). The second (mid) sub-figure is the computed kernel matrix. The bottom sub-figure shows the cluster label matrix obtained according to the Algorithm 4.

Figure 3 shows the effect of attack duration on the detection performance. Recall that, independent of the duration of the attack, the number of messages that an attacker sends is within a fixed interval. Thus, if the attack duration is long (e.g., attack starts at 0.25 sec) then the messages are more dispersed in time; on the contrary, if the attack duration is short (e.g., attack starts at 0.75 sec), then all messages are concentrated within a shorter interval and our algorithm can detect them more accurately, since the pairwise heat kernel returns higher values when the messages have been sent with close timings.

The effect of increased attacker number is shown in Figure 4. The higher the number of attackers in the system, the more accurately the algorithm detects them. It detects all the attackers almost without any false alarms. Not surprisingly, in both cases, the attackers are detected more accurately when the intensity of the attack becomes higher.

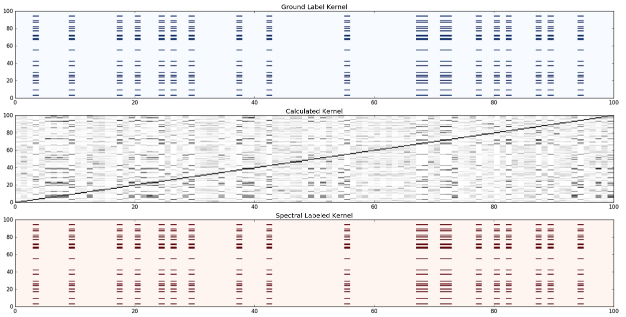

Figure 2. Detected attackers on simulated data using spectral clustering

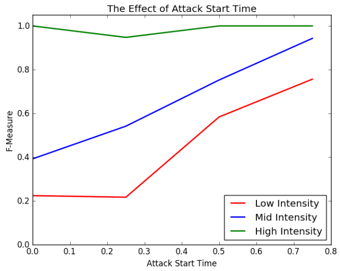

Figure 3. The Effect of Attack Duration for Fixed Number Attackers

6. Conclusion and Future Work

We have proposed a novel method to find the group of attackers within a group of users and tested it using simulation data. The attackers are characterized by coordinated message sending behaviors, similar to the botnet DDoS attacks. The proposed method discriminates the attackers from the victims using similarities between them.

Each user is regarded as a time series where each message is represented as a unit vector. A sequence alignment kernel is used to measure similarity between the message sequences and their timings. Then the users are clustered into two groups using spectral clustering. Finally, a heuristics is applied for autonomous attacker cluster selection.

The performance of the proposed method improves if the attackers send high number of messages or they send the messages in bursts in small time intervals. Similarly, the method performs more accurately if the number of attackers increase.

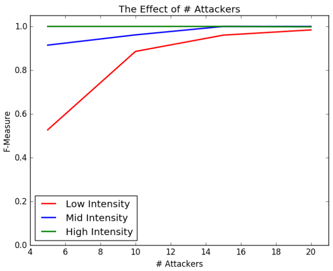

Figure 4. The Effect of the Number of Attackers for a Fixed Attack Duration

Acknowledgements

This study is a Bogazici University - NETAS collaboration and it is funded with TEYDEB project number 3140701, ”Realization of Anomaly Detection and Prevention with Learning System Architectures, Quality Improvement, High Rate Service Availability and Rich Services in a VoIP Firewall Product”, by the Scientific and Technological Research Council Of Turkey (TUBITAK). NOVA V-Gate is a trademark cyber-security product of NETAS.

References

Chen, H., Tang, F., Tino, P., and Yao, X. Model-based kernel for efficient time series analysis. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’13, pp. 392–400, New York, NY, USA, 2013. ACM.

Cuturi, M. Fast global alignment kernels. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 929–936, 2011.

Cuturi, M., Vert, J. P., Birkenes, O., and Matsui, T. A kernel for time series based on global alignment. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2007 (ICASSP’07)*, volume 2, pp. 413–416, 2007.

Gupta, M., Gao, J., Aggarwal, C. C., and Han, J. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, 2014.

Hyndman, R. J., Wang, E., and Laptev, N. Large-scale unusual time series detection. In *IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA, November 14-17, 2015*, pp. 1616–1619, 2015.

Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Oates, T., Firoiu, L., and Cohen, P.R. Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 Workshop on Neural, Symbolic, and Reinforcement Learning Methods for Sequence Learning*, 1999.

Shi, J. and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Sivaramakrishnan, K. R., Karthik, K., and Bhattacharyya, C. Kernels for large margin time-series classification. In *2007 International Joint Conference on Neural Networks*, pp. 2746–2751, Aug 2007.

Xiong, Y. and Yeung, D.-Y. Mixtures of arma models for model-based time series clustering. In *Proceedings of the IEEE International Conference on Data Mining*, 2002.

Zhang, X., Liu, J., Du, Y., and Lv, T. A novel clustering method on time series data. *Expert Systems with Applications*, 38(9):11891 – 11900, 2011.