


# Towards Long-Term Continuous Tracing of Internet-Wide Scanning Campaigns Based on Darknet Analysis

Chansu Han<sup>1</sup><sup>a</sup>, Akira Tanaka<sup>1</sup>, Jun'ichi Takeuchi<sup>2</sup>, Takeshi Takahashi<sup>1</sup>, Tomohiro Morikawa<sup>3</sup>  
and Tsung-Nan Lin<sup>4</sup>

<sup>1</sup>*National Institute of Information and Communications Technology, Tokyo, Japan*

<sup>2</sup>*Kyushu University, Fukuoka, Japan*

<sup>3</sup>*University of Hyogo, Hyogo, Japan*

<sup>4</sup>*National Taiwan University, Taipei, Taiwan, Republic of China*

**Keywords:** Darknet Analysis, Scanning Campaign, Tracing, Non-Negative Matrix Factorization.


**Abstract:** The darknet is an unused IP address space that can be an effective resource for observing and analyzing global indiscriminate scanning attacks. Scanning traffic on the darknet has expanded dramatically in recent years and numerous constant scans for investigative purposes have been observed. This is problematic because the investigative scans identified by naive rules account for about 60% of the total observed traffic. In earlier work, we detected malware-caused indiscriminate scanning for attack purposes from darknet data by means of anomaly detection methods, but the large amount of activity from investigation-purpose indiscriminate scans led to false positives. We have therefore developed a new method for tracing scanning campaigns. By distinguishing whether the campaign being traced is for attack or investigation purposes, we aim to reduce the number of false positives and improve anomaly detection accuracy. We also intend to clarify the actual state of constant scanner groups by tracing them. In this work, we describe the proposed method, implement a prototype, and conduct experiments on real darknet data to investigate the feasibility of tracing scanning campaigns.

## 1 INTRODUCTION

The number of indiscriminate cyberattacks and Internet-wide scans reaching the darknet<sup>1</sup>, an unused IP address space, has increased dramatically in recent years. Our team monitors approximately 300K IP addresses through NICTER<sup>2</sup>, a large-scale darknet observation system. Although analyzing darknet data is effective in terms of understanding global cyberattack trends, the analysis is costly due to the diversity and volume of observed packets. Furthermore, constant scanning for investigative purposes has recently emerged to form the majority of observed scans, which creates noise that interferes with analyzing essential threats (Endo et al., 2022).

We previously developed a synchronization anomaly detection framework called Dark-TRACER with the goal of detecting malware-induced indiscriminate scanning attack activities before an attack goes viral (Han et al., 2022; Han et al., 2020). We found that while Dark-TRACER detected threats an average of 126.4 days earlier than when they were first observed and recognized by NICTER, it also detected many false positive alerts that were unrelated to (or had little to do with) attack activities.

In our experience, most false positive alerts are anomalies detected due to synchronization by investigative scanners. Identifying whether an anomalous alert is a campaign by investigative scanners unrelated to an attack activity is challenging. Moreover, the investigative scanner packets we identified in (Endo et al., 2022) amounted to 50 to 60% of all packets, even if we only naively examined them, and there is a high possibility that there are more investigative scanners in reality. Investigative scans have previously been ignored because of their small scale and low resemblance to attack activities, but they have

<sup>a</sup> <https://orcid.org/0000-0002-1728-5300>

<sup>1</sup>The term “darknet” is also known as a network tele-scope and should not be confused with anonymous communication systems such as Tor.

<sup>2</sup><https://www.nicter.jp/>

now reached a point of systematic advancement and are large-scale enough to hinder darknet analysis. It is therefore vital to clarify the nature of investigative scanners and reduce false positives by identifying the causes of alerts.

To this end, conducting long-term, sequential analysis of temporal changes in analysis targets may be more beneficial than one-off analysis in terms of identifying and understanding the objectives of the target scanner group's activities. In this study, we present a unique approach for tracing the activities of investigative and offensive scanner groups and capturing the actual status of constant scanner groups. Specifically, we successively perform non-negative matrix factorization (NMF) (Lee and Seung, 2000) for a short-term period over time-series data while shifting the data little by little. We restrict the decomposition results to coinciding with overlapping intervals of preceding and following time-series data; it should be possible to trace analysis targets sequentially over long-term time-series data.

There are three key advantages to our approach:

- Since the NMF is performed sequentially over a long-term period while inheriting the decomposition results, bases are uniquely fixed with respect to the overlap period and do not change. A 'basis' here refers to a group of scanners that exhibit similar temporal changes, and the number of bases is a hyperparameter in the NMF. Therefore, tracing can be flexible even if there are changes to the scanner specifications (scanner IP addresses, scan frequency, etc.). This tracing flexibility is nearly impossible to achieve with other methods and thus forms the most important element of our approach.
- Even if analysis targets are not specified in advance, the NMF decomposes scanner groups with similar (synchronous) temporal patterns, which enables us to trace scanner groups that behave similarly. Of course, tracing a given set of targets in advance is also possible.
- Since the NMF is relatively computationally inexpensive, real-time tracing is possible for large-scale darknet data.

In this work, we present the details of the proposed method, discuss our prototype implementation, and report the results of experiments on real darknet traffic data to evaluate the feasibility of tracing. Our findings demonstrate that the proposed method requires less processing time than the original NMF and has fewer deviations of decomposition results, and that judgment of tracing success or failure is feasible.

## 2 TRACING TARGET

The ultimate goal of tracing in this study is to automatically identify scanner groups that behave similarly over the long-term and investigate their activities. For this purpose, we need a flexible tracing method that can trace scanners even when their specifications (e.g., IP addresses or scanning frequency) change. Since there are various potential tracing targets, it is difficult to define them, but here we describe a few specific ones.

Hosts infected with the same worm-type malware execute scans with similar temporal patterns over a long-term period. We want to identify and trace such similar infected hosts as a group campaign. In addition to worms, we analyze systemized scanners from Internet-wide scanning service providers such as Shodan, Censys (Durumeric et al., 2015), Rapid7, Onyphe, Shadowserver, and BinaryEdge<sup>3</sup>, as well as those from research institutions such as the University of Michigan<sup>4</sup>. Even if multiple scanner groups use well-known scanning tools such as ZMap (Durumeric et al., 2013), Masscan, or Nmap, we want to identify and trace each as a distinct group rather than tracing the scanning tools.

There are considered to be advanced scanning tools/programs among the scanners. For example, there are a random scan and a stealth scan. Are they tracing targets in this study? First, the random scan performs reconnaissance on random destination IP addresses. We have a worldwide network of darknet observation nodes and monitor the large-scale darknet. Fast random scans show similar spatiotemporal properties within a certain period in our large-scale darknet. Thus, random scans fall within the scope of our analysis target.

Next, the stealth scan performs slow and sporadic reconnaissance to conceal its scanning activity. In this case, stealth scan hosts may not scan with similar temporal patterns. However, their slowness makes them unsuitable for malicious scanning activities that require rapid scan execution (e.g., spreading malware infections or probing for vulnerable devices). Therefore, stealth scanners with malicious intent are considered to be scarce. On the other hand, stealth scanners with benign intentions are considered small in scale and do not cause problems in cyberspace. Although stealth scanners are outside the scope of our analysis target this time, groups of stealth scanners that do not

<sup>3</sup>From Shodan to BinaryEdge in order: <https://www.shodan.io/>, <https://censys.io/>, <https://www.rapid7.com/>, <https://www.onyphe.io/>, <https://www.shadowserver.org/>, and <https://www.binaryedge.io/>.

<sup>4</sup><https://cse.engin.umich.edu/about/resources/connection-attempts/>.

Table 1: Notation of mathematical symbols.

Notation	Description
$V \in \mathbb{N}_{\geq 0}^{n \times m}$ ( $\mathbb{N}_{\geq 0} = \{0, 1, 2, \dots\}$ )	Long-term time-series data matrix whose component $(i, j)$ is the number of packets at time $j$ for host $i$
$V[j : j'] \in \mathbb{N}_{\geq 0}^{n \times (j' - j + 1)}$	A submatrix of columns $j$ through $j'$ of matrix $V$
$t := [st + 1, st + m']$ $\subseteq \mathbb{N}_{\geq 0}$ ( $t \in \{0, 1, \dots\}$ )	The time width $t$ is defined as the interval from time $st + 1$ to time $st + m'$ . The width of sliding window is $s$ , where $m'$ is the width of interval, satisfying $s < m'$ .
$V_t := V[st + 1 : st + m']$ $\in \mathbb{N}_{\geq 0}^{n \times m'}$ ( $t \in \{0, 1, \dots\}$ )	A submatrix of $V$ at a time width $t$

emerge on the surface are interesting targets and are the subject of future research. The proposed method described in this paper may be able to trace stealth scanners, but we do not evaluate this possibility.

We briefly describe related works now. Conventional studies (Mazel et al., 2017; Griffioen and Doerr, 2020; Tanaka et al., 2021) that identify scanners in a rule-based manner are unable to analyze unknown groups, perform fine-grained groupings, and perform long-term tracing. The clustering method of scanners (Cohen et al., 2020) also cannot perform long-term tracing of clusters when scanning hosts' IP addresses change. It was reported that when an infected device sends many packets in the short term and becomes overloaded, it repeatedly disconnects/reconnects PP-PoE, causing a single host to change to multiple IP addresses (Endo et al., 2022).

### 3 PROPOSED METHOD

In this section, we define the notation and then introduce the proposed method and its pseudo-code.

#### 3.1 Notation

Table 1 shows the notation used in this paper. Although we consider darknet traffic data here, it is applicable to any general multivariate time-series data. As a brief explanation of this notation, imagine a very long horizontal time-series data matrix  $V$  of length  $m$  and many submatrices  $V_0, V_1, V_2, \dots$  of width  $m'$  at each time width  $t$ , sliding by  $s$ -widths. Note that the submatrices  $V_t$  and  $V_{t+1}$  at the time before and after overlap the submatrix  $V[s(t+1)+1 : st+m']$  of column length  $m' - s$  (colored in red) and are exactly equal, as shown in the following equation.

$$\begin{aligned}
 V_t &= V[st+1 : st+m'] \\
 &= \left( V[st+1 : s(t+1)] \mid V[s(t+1)+1 : st+m'] \right) \\
 V_{t+1} &= V[s(t+1)+1 : s(t+1)+m'] \\
 &= \left( V[s(t+1)+1 : st+m'] \mid V[st+m'+1 : s(t+1)+m'] \right)
 \end{aligned}$$

#### 3.2 Proposed Method: LST-NMF

We call the proposed method long short-term non-negative matrix factorization (LST-NMF) because it performs the NMF (Lee and Seung, 2000) sequentially by gradually sliding short-term data out of long-term time-series data. The conventional NMF is a one-shot analysis method in which the matrix  $V$  is approximately decomposed by the multiplicative update (MU) algorithm to obtain matrices  $W$  and  $H$  so that  $V \approx WH$ . For matrices  $W$  and  $H$ , latent groups of frequent patterns are obtained for the number of bases. For instance, the NMF decomposition of matrix  $V$  yields latent host groups of synchronized spatiotemporal patterns for a given number of bases.

However, it cannot successively analyze the temporal dependencies of analysis targets, which is the problem we are trying to solve. Therefore, we first consider how to address the problem without significantly modifying the original NMF algorithm.

We decompose the data matrix  $V_t$  at time width  $t$  in NMF to be  $V_t \approx W_t H_t$ , and then decompose the data matrix  $V_{t+1}$  at the next time width using the decomposition results  $W_t$  and  $H_t$  ( $W_t \in \mathbb{R}^{n \times r}$ ,  $H_t \in \mathbb{R}^{r \times m'}$ ). Here, matrix  $W$  denotes the spatial feature (scan host) information,  $H$  denotes the temporal feature information, and  $r$  denotes the number of bases (ranks). The specific method of utilizing the decomposition results  $W_t, H_t$  is that when decomposing matrix  $V_{t+1}$ , the temporal feature matrix  $H_{t+1}$  is fixed for the overlap period with  $H_t$ . In other words, multiplicative update (MU) is performed with the condition  $H_{t+1}[1 : m' - s] = H_t[s + 1 : m']$ . The intuitive picture of the matrix decomposition at preceding and following times can be interpreted as follows. The submatrices highlighted in red are the blocks to be fixed.

$$\begin{aligned}
 V_t &= \left( V_t[1 : s] \mid V_t[s+1 : m'] \right) \\
 &\approx \left( W_t \right) \left( H_t[1 : s] \mid H_t[s+1 : m'] \right) \\
 V_{t+1} &= \left( V_{t+1}[1 : m' - s] \mid V_{t+1}[m' - s + 1 : m'] \right) \\
 &\approx \left( W_{t+1} \right) \left( H_{t+1}[1 : m' - s] \mid H_{t+1}[m' - s + 1 : m'] \right)
 \end{aligned} \tag{1}$$

We confirmed that the original NMF is valid even when the decomposition matrices  $W$  and  $H$  are updated with some of their values fixed. The convergence of the NMF can be proved by using the auxiliary function method to show that the target function is monotonically decreasing by a multiplicative update (MU). The solutions of the auxiliary function and its minimization problem are invariant regardless of the fixation. The update is performed only for the unfixed elements of matrices  $W$  and  $H$ . As a result, the decomposition of matrix  $V_{t+1}$  in Eq. (1) is updated only for the unfixed elements of matrices  $W_{t+1}$  and  $H_{t+1}[m' - s + 1 : m']$  (colored in blue).

### 3.3 Problem Formulation

Here we formulate the problem in LST-NMF. Optimization problems  $P_0, P_1, P_2, \dots$  are solved in order:

$$\begin{aligned}
 P_0 : \min_{W_0, H_0} & \|V_0 - W_0 H_0\|_F^2, \\
 P_1 : \min_{W_1, H_1} & \|V_1 - W_1 H_1\|_F^2 \\
 & \text{s.t. } H_1[1 : m' - s] = H_0[s + 1 : m'], \\
 & \vdots \\
 P_t : \min_{W_t, H_t} & \|V_t - W_t H_t\|_F^2 \\
 & \text{s.t. } H_t[1 : m' - s] = H_{t-1}[s + 1 : m'],
 \end{aligned}$$

where  $\|\cdot\|_F^2$  is the sum-of-squares error of the Frobenius norm. Also, note that problem  $P_0$  is not subject to a fixation condition statement.

### 3.4 Pseudo-Code

The pseudo-code of LST-NMF is provided in Algorithm 1. The first data matrix  $V_0$  is decomposed using the original NMF (line 1). After that, the data matrix at the following time is sequentially decomposed using the decomposition results of the preceding time.

The temporal feature matrix  $H_t$  is initially fixed at  $H_t[1 : m' - s] \leftarrow H_{t-1}[s + 1 : m']$  (line 5), and only unfixed blocks are updated in the update formula (lines 10–11). The spatial feature matrix  $W_t$  also uses the decomposition result  $W_{t-1}$  of the preceding time as the initial value (line 6). The singular value decomposition (SVD) is used for other blocks with no initial values.

As a result, the number of iterations is reduced because only a small number of elements in matrix  $H$  are updated and the range to be updated is small. Also, the decomposition result  $W_t$  is expected to be a matrix with values close to  $W_{t-1}$ . The smaller the sliding width  $s$  compared to the data interval width  $m'$ , the larger the expected effect.

---

#### Algorithm 1: LST-NMF: Long Short-Term Non-negative Matrix Factorization.

---

**Require:** A sequence of long-term time-series data matrices  $\mathbb{V} = (V_0, V_1, V_2, \dots)$  ( $V_t \in \mathbb{N}_{\geq 0}^{n \times m'}$ ), rank parameter  $r < \min(n, m')$ , thresholds  $\epsilon, L$ .

**Ensure:** Sequences of decomposed matrices  $\mathbb{H} = (H_0, H_1, H_2, \dots)$  and  $\mathbb{W} = (W_0, W_1, W_2, \dots)$ . ( $H_t \in \mathbb{R}^{r \times m'}$ ,  $W_t \in \mathbb{R}^{n \times r}$ ,  $V_t \approx W_t H_t$ ,  $H_t[1 : m' - s] = H_{t-1}[s + 1 : m']$ )

```

/* Compute first matrix  $V_0$  with original NMF */
1:  $H_0, W_0 \leftarrow \text{NMF}(V_0, r, \epsilon, L)$ 

/* Compute LST-NMF with fixing duplicate block */
2: while  $t \leftarrow 1, 2, 3, \dots$  do
3:    $\ell, f(0) \leftarrow 0$ 
4:    $H_t \leftarrow \text{SVD}(V_t)$  // Initialization with SVD
5:    $H_t[1 : m' - s] \leftarrow H_{t-1}[s + 1 : m']$  // Initialization
6:    $W_t \leftarrow W_{t-1}$  // Initialization
7:    $V_t' \leftarrow V_t[m' - s + 1 : m']$  // Obtain unfixed block
8:   while  $\delta < \epsilon$  or  $\ell \geq L$  do
9:      $H_t' \leftarrow H_t[m' - s + 1 : m']$  // Obtain unfixed block
10:     $H_t' \leftarrow H_t' \frac{W_t^T V_t'}{W_t^T W_t H_t'}$  // Update only unfixed block
11:     $W_t \leftarrow W_t \frac{V_t H_t'^T}{W_t H_t H_t'^T}$  // Update only unfixed block
12:     $\ell \leftarrow \ell + 1$ 
13:     $f(\ell) \leftarrow \|V_t - W_t H_t\|_F^2 / (n_t \cdot m)$  // Norm
14:     $\delta \leftarrow |f(\ell - 1) - f(\ell)|$  // Reduction in norm
15:  end while
/* Normalization */
16:   $\Lambda \leftarrow \text{diag}(\sum_j H_t(r, j))$ 
17:   $W_t \leftarrow W_t \Lambda, H_t \leftarrow \Lambda^{-1} H_t$ 
18: end while

```

---

### 3.5 Normalization

Next, the normalization is performed in lines 16–17 of Algorithm 1. Since the values of matrices  $W, H$  obtained by the NMF are not uniquely determined, we normalize these matrices to align the value scales. We update matrix  $W$  to match its scale by summing each row (basis) of matrix  $H$  equal to 1. Expressed in equations, the normalized matrices are  $\tilde{H}(r, j) = H(r, j) / \sum_j H(r, j)$  and  $\tilde{W}(i, r) = W(i, r) \cdot \sum_j H(r, j)$ . If we set the diagonal matrix  $\Lambda = \text{diag}(\sum_j H(r, j))$ , we can easily calculate the normalization as  $\tilde{W} = W\Lambda$ ,  $\tilde{H} = \Lambda^{-1}H$ . The normalized value can then be interpreted as  $\tilde{W}(i, r)$  packets observed in the temporal pattern represented by the  $r$ -th row of  $\tilde{H}$  from the  $i$ -th host.

The NMF computed with the rank parameter  $r$  can be expressed as  $V(i, j) \approx \sum_r \tilde{W}(i, r) \tilde{H}(r, j)$ . Summing this expression in the column direction yields

$$\begin{aligned}
 \sum_j V(i, j) & \approx \sum_j \left( \sum_r \tilde{W}(i, r) \tilde{H}(r, j) \right) \\
 & = \sum_r \tilde{W}(i, r) \left( \sum_j \tilde{H}(r, j) \right) = \sum_r \tilde{W}(i, r).
 \end{aligned}$$



The total number of packets observed from the  $i$ -th host equals the sum of  $\tilde{W}$  in row  $i$ .

## 4 EVALUATION

In this section, we first describe the experimental setup and data preprocessing. We then present a comparison of the proposed LST-NMF and the original NMF. Finally, we introduce a tracing method in LST-NMF and report the results of the tracing.

### 4.1 Experimental Setup

In the actual implementation of the LST-NMF prototype, the data matrix  $V_t$  handles only the hosts that appear in each time width to avoid unnecessary memory consumption. The initial value of matrix  $W_t$  in line 6 of Algorithm 1 is initialized with  $W_{t-1}$  only for hosts that overlap with the preceding time, and SVD is used for hosts that do not overlap with the preceding time. Otherwise, Algorithm 1 remains unchanged.

We conducted experiments using real darknet traffic data, specifically, darknet sensor data of a subnet /20 scale (approx. 4K IP addresses) for one day on September 1, 2022. Only TCP-SYN packets, which are indiscriminate scan attacks, were analyzed. This dataset will be available on our website. This dataset is available on our website<sup>5</sup>.

The hyperparameters utilized in the experiments are described below. The data matrix consisted of 30 minutes of data, counted in packets at 1-minute intervals and sliding by 1 minute (i.e.,  $m' = 30, s = 1$ ). The rank parameter in LST-NMF was set to  $r = 5$ , and update stop thresholds were set to  $\epsilon = 10^{-5}, L = 3,000$ . This hyperparameter setting is based on our previous study with Dark-NMF (Han et al., 2021).

Experiments were performed on Ubuntu 18.04 with a Ryzen Threadripper 2990WX CPU and 128GB memory.

### 4.2 Data Preprocessing

Before starting the experiment, we removed redundant noise hosts (scanners) in the data as a preprocessing step. The following hosts were excluded:

- A host  $i$  with a low average number of packets per minute in the 30-minute data matrix  $V_t$  ( $\frac{\sum_j V_t(i,j)}{m'} \leq a$ )
- A host  $i$  with numerous packets per minute in the entire data matrix  $V$  ( $V(i,j) \geq b$ )

- A host  $i$  with few destination IP addresses that sent packets in the 30-minute data matrix  $V_t$  ( $\text{ip.dst}(i) \leq c$ )

Most of the hosts in this dataset had a small number of packets sent. In terms of computational reduction, they were excluded as noise because their presence is negligible and the reduction in the number of hosts is significant. Conversely, hosts with a large number of packets were excluded because they are greatly affected by large values when performing the approximated decomposition in NMF. Finally, hosts with few destination IP addresses were excluded as noise because they are not considered scanners.

Histograms were calculated for the three preprocessing rules described above, and the threshold values  $a, b$ , and  $c$  were determined appropriately on the basis of the overall data distribution. The number of hosts in data matrix  $V_t$  per minute at each time is shown in Fig. 1 after removing hosts by setting  $a = 0.1, b = 60, c = 1$ . Consequently, the average host reduction rate was 66.5% at each time point, and the average number of duplicate hosts was about 97.8% at each time point due to the preprocessing. The average number of hosts after preprocessing at each time was about 6,500, and the size of data matrix  $V_t$  was about  $\mathbb{N}_{\geq 0}^{6500 \times 30}$  (the average number of duplicate hosts was about 6,300). The fluctuation rate of duplicate hosts at each time was not large (red line in Fig. 1b).

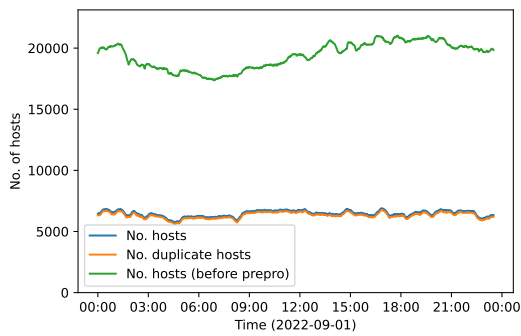
### 4.3 Comparative Evaluation of LST-NMF and Original NMF

We computed LST-NMF and the original NMF on the real darknet traffic data specified in Section 4.1 and compared the results from the following four perspectives:

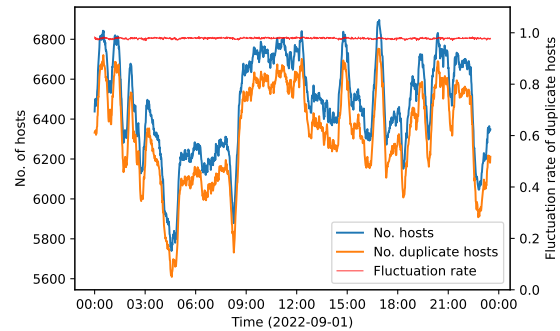
- **Iterations:** the number of iterations at the end of the algorithm
- **Runtime:** the time required to compute the algorithm
- **Approximation Error** ( $\|V_t - W_t H_t\|_F^2$ ): error at the end of the algorithm. A large approximation error indicates poor decomposition accuracy.
- **Deviation** ( $\|W_t H_t - \tilde{W}_t \tilde{H}_t\|_F^2$ ): deviation of the approximate decomposition results between LST-NMF ( $W_t, H_t$ ) and the original NMF ( $\tilde{W}_t, \tilde{H}_t$ ). A large deviation indicates a large gap between both algorithms' decomposition results.

First, our comparison of the number of iterations showed that the average iteration ratio at each time point was 4.21, with LST-NMF completing the calculation in approximately four times fewer iterations.

<sup>5</sup><https://csdataset.nict.go.jp/darknet-2022/>

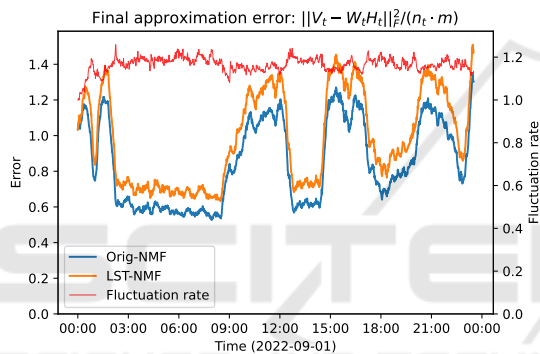


(a) Green: no. of hosts before preprocessing, blue: no. of hosts after preprocessing, orange: no. of duplicate hosts in the preceding and following data matrices. The blue and orange lines are enlarged in Fig. 1b.

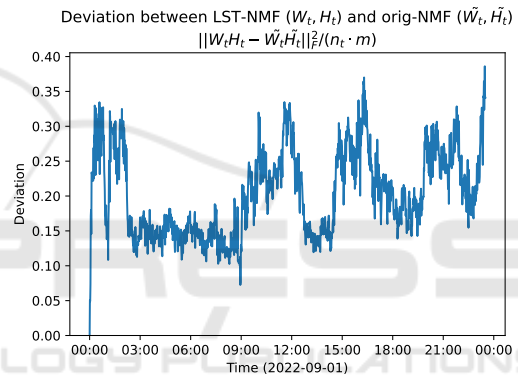


(b) Enlarged view of the number of hosts after preprocessing in Fig. 1a (red: fluctuation rate of duplicate hosts).

Figure 1: The number of hosts in data matrix  $V_t$  at each time (every minute) before and after preprocessing with  $\sum_j V_t(i, j)/m' \leq 0.1$ ,  $V(i, j) \geq 60$ , and  $ip.dst(i) \leq 1$ .



(a) Approximation errors (red: fluctuation rate of both errors).



(b) Deviation between LST-NMF and original NMF.

Figure 2: Comparative evaluation of LST-NMF and original NMF.

The average iterations and standard deviations were 43.62 and 31.84 for LST-NMF and 130.47 and 51.27 for the original NMF, respectively. Similar to the comparison of iterations, the average runtime ratio at each time point was 6.23, with LST-NMF completing processing about six times faster. The average runtimes and standard deviations were 0.19 and 0.14 s for LST-NMF and 0.83 and 0.33 s for the original NMF, respectively.

The LST-NMF had fewer iterations and shorter runtimes than the original NMF because it utilizes the preceding time's decomposition result as the initial value and keeps most of the values fixed during the iterative update.

Next, we compared the approximation error  $\|V_t - W_t H_t\|_F^2$  and the deviation  $\|W_t H_t - \tilde{W}_t \tilde{H}_t\|_F^2$  between LST-NMF ( $W_t, H_t$ ) and the original NMF ( $\tilde{W}_t, \tilde{H}_t$ ). As shown in Fig. 2, the average approximation error ratio at each time point was 1.17, with the LST-NMF showing an error approximately 17% larger than the

original NMF. The average approximation error and standard deviation were 0.99 and 0.26 for LST-NMF and 0.85 and 0.24 for the original NMF, respectively. The average and std of the deviation between  $WH$ , which approximates LST-NMF and the original NMF to the original matrix  $V$ , were 0.2 and 0.06, respectively.

Due to the iterative update with fixed duplicate blocks, the approximation error of LST-NMF was about 17% worse than the original NMF. However, the error did not increase with time, and there were no intervals where the fluctuation rate changed significantly (red line in Fig. 2a). The slightly higher loss compared to the original NMF does not deteriorate the performance of long-term tracing. Also, the deviation was 4 to 5 times smaller than the approximation error, indicating that the LST-NMF results were not far from the original NMF results.

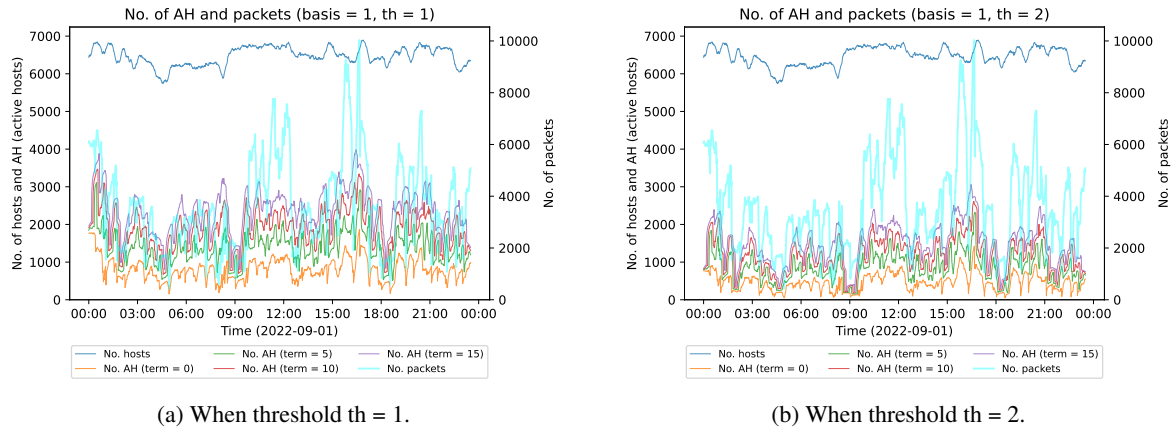


Figure 3: AS determination results for basis number 1. Blue: no. of all hosts, cyan: no. of packets in basis number 1 ( $\sum_i W_t(i, 2)$ ), other: no. of ASes per loose judgment  $X$  (AH = AS, term =  $X$ ).

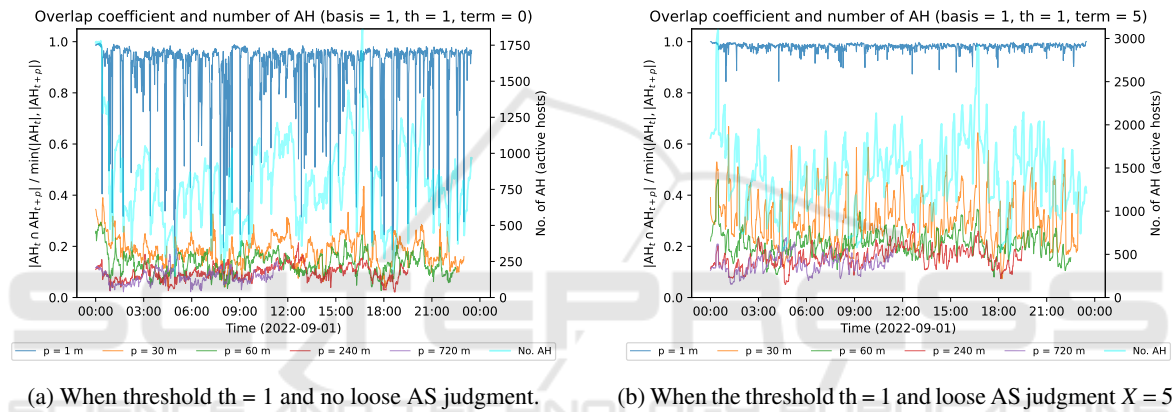


Figure 4: AS fluctuation indicator results for basis number 1.

#### 4.4 Tracing Method and Result

LST-NMF enables us to analyze the temporal dependence of analysis targets in long-term time-series data sequentially over a long period. Here, we discuss a method for tracing whether a target is observed over a long-term period from the LST-NMF analysis results.

The LST-NMF analysis results for preceding and following times are fixed, so the bases are uniquely fixed and do not change. In other words, since the bases are maintained, the long-term LST-NMF analysis results for each basis enable us to identify scanner groups whose packets are observed in a similar temporal pattern over a long period. The tracing procedure is as follows:

1. Determine the active scanners (hereafter, ASes) on each basis from the LST-NMF analysis results at each time.
2. Judge the success or failure of tracing based on the fluctuation indicator of ASes on each basis.
3. Determine the primary ASes in the successfully

traced bases, and analyze the behavior of these scanner groups.

Active scanners indicate scanners with (significant) activities on a basis. The thresholds for determining the ASes (step 1) and setting the fluctuation indicator (step 2) are described in Section 4.4.1. By analyzing the behavior of primary ASes, we can investigate the cause and the purpose of the scanner group’s activities and get a better understanding of the actual situation. However, in this paper, we do not analyze the primary AS, which is planned for future study.

##### 4.4.1 Judgment Result of Active Scanner

Here we introduce the AS determination method and AS fluctuation indicator and investigate the feasibility of determining whether or not the analysis target can be successfully traced. First, ASes were determined by setting a threshold value from the spatial feature matrix  $W_t$  at each time width. As explained in Section 3.5, the normalized spatial feature matrix  $W$  value represents the number of packets observed from each host on

each basis. Therefore, hosts with a value of less than 1 in the normalized  $W$  can be interpreted as inactive on the basis. Considering that the approximation error of LST-NMF is close to 1, we decided to determine AS by adopting a threshold value instead of just “less than 1”. In addition, as a consideration for hosts that did not appear accidentally because the sliding width  $s$  and data interval width  $m'$  were too fine for 1 minute and 30 minutes, we introduced the following loose judgment method for AS: If a host is AS at least once in  $X$  minutes before or after a certain time, it is considered AS at that time.

Figure 3 shows the results of AS determination for thresholds  $th = 1$  and  $2$  and the loose judgments  $X = 0, 5, 10,$  and  $15$ .  $X = 0$  means no loose judgment. Since not all five bases can be listed due to space limitations, only the results for basis number 1 are included. We found that the number of ASes decreased when the threshold  $th$  increased, and the number of ASes increased when the loose judgment  $X$  increased.

Next, we utilized the AS fluctuation indicator to judge the success or failure of tracing. The Simpson coefficient (aka overlap coefficient) was utilized as the AS fluctuation indicator, and the results for basis number 1 are shown in Fig. 4. Let  $AS_t$  denote the set of ASes at time width  $t$ . The AS fluctuation indicator (Simpson coefficient) that compares the set of ASes  $p$  time width ahead was calculated as  $|AS_t \cap AS_{t+p}| / \min(|AS_t|, |AS_{t+p}|)$ . The  $p$  was set to 1 m, 30 m, 1 h, 4 h, and 6 h, and we checked the fluctuation of the AS set at each  $p$ . *If the AS fluctuation indicator (Simpson coefficient) is large, we can assume that the ASes are successfully traced. Conversely, a small value indicates that the ASes have changed significantly, which can be interpreted as the basis disappearing or changing its behavior.*

As shown in Fig. 4a, the AS fluctuation indicator for  $p = 1$  was greatly blurred when there was no loose AS judgment. In contrast, Fig. 4b shows that, since the loose AS judgment at  $X = 5$  allowed for consideration of hosts that did not appear accidentally, there was less blurring of the AS fluctuation indicator, which continued to exhibit a high value. These results demonstrate that tracing the success or failure can be determined from this indicator.

Similar AS determinations and AS fluctuation indicator results were obtained for the other three bases, except for one. The one-basis exception was almost ten times larger than the others, and most hosts were determined to be ASes. This may have been due to the small number of bases  $r$ , so tuning will be performed.

## 5 DISCUSSION AND FUTURE WORK

As stated in Section 1, the biggest advantage of LST-NMF is that the bases do not change, which is a factor that cannot be achieved with the original NMF. While it is also possible to analyze a long-term time-series data matrix in one operation using the original NMF, the matrix size would be huge and the processing time would be extremely long. Moreover, local events would likely not be detected due to the inability to decompose the matrix into fine units.

One concern with the current LST-NMF is that it fixes overlapping data at the time before and after the decomposition, which may have a long-term effect on past decomposition results. In the future, we would like to incorporate flexibility by simultaneously decomposing data matrices before and after times and by approximating overlapping data at both times instead of fixing them.

We also plan to conduct a tracing case study by analyzing the behavior of primary ASes. Since LST-NMF can identify scanner groups that have been sending packets with similar temporal patterns for a long-term period, it is expected to be relatively convenient to perform behavior analysis. Such analysis should be able to capture distinctive behaviors based on the statistics of destination port information, other header information, and reverse lookup information of source IP addresses.

We have been working on a platform for rapid analysis and notification of critical incidents in the cybersecurity field. We perform the cross-checking analysis with the darknet and different sources such as honeypots, malware analysis, and cyber threat intelligence (Takahashi et al., 2021). The results of scanner groups with similar long-term behavior identified and traced by LST-NMF will also be contributed to this platform and deployed to the public.

## 6 CONCLUSION

In this paper, we proposed LST-NMF, a method for tracing scanner group activities. Our experiments showed that, compared to the original NMF, the processing time of the proposed method was shorter and there was only a small deviation of the decomposition results. We also confirmed the feasibility of tracing success or failure. As this suggests that LST-NMF can determine and distinguish investigative scanner activities, we should be able to significantly reduce the false positives from Dark-TRACER alerts that are not related to true threats and thereby better understand



the actual status of scanners. In particular, if we can filter out investigative scanners, it will be possible to focus our analysis on only essential threats and scanning activities. Moreover, LST-NMF can be applied to multivariate time-series data in various fields and extended to high-dimensional time-series tensors.

## ACKNOWLEDGEMENTS

This research was conducted as part of the “MITI-GATE” project in “Research and Development for Expansion of Radio Wave Resources (JPJ000254)”, which was supported by the Ministry of Internal Affairs and Communications, Japan.

## REFERENCES

- Cohen, D., Mirsky, Y., Kamp, M., Martin, T., Elovici, Y., Puzis, R., and Shabtai, A. (2020). DANTE: A framework for mining and monitoring darknet traffic. In *Computer Security – ESORICS 2020*, pages 88–109. Springer International Publishing.
- Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. (2015). A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM.
- Durumeric, Z., Wustrow, E., and Halderman, J. A. (2013). Zmap: Fast internet-wide scanning and its security applications. In *Proceedings of the 22th USENIX Security Symposium*, pages 605–620.
- Endo, Y., Mori, Y., and Kubo, M. (2022). NICTER statistics – Q2, 2022, Cybersecurity Laboratory, National Institute of Information and Communications Technology. [https://blog.nicter.jp/2022/08/nicter\\_statistics\\_2022\\_2q/](https://blog.nicter.jp/2022/08/nicter_statistics_2022_2q/) (In Japanese).
- Griffioen, H. and Doerr, C. (2020). Discovering collaboration: Unveiling slow, distributed scanners based on common header field patterns. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE.
- Han, C., Shimamura, J., Takahashi, T., Inoue, D., Takeuchi, J., and Nakao, K. (2020). Real-time detection of global cyberthreat based on darknet by estimating anomalous synchronization using graphical lasso. *IEICE Transactions on Information and Systems*, E103.D(10):2113–2124.
- Han, C., Takeuchi, J., Takahashi, T., and Inoue, D. (2021). Automated detection of malware activities using non-negative matrix factorization. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE.
- Han, C., Takeuchi, J., Takahashi, T., and Inoue, D. (2022). *Dark-TRACER*: Early detection framework for malware activity based on anomalous spatiotemporal patterns. *IEEE Access*, 10:13038–13058.
- Lee, D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.
- Mazel, J., Fontugne, R., and Fukuda, K. (2017). Profiling internet scanners: Spatiotemporal structures and measurement ethics. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE.
- Takahashi, T., Umemura, Y., Han, C., Ban, T., Furumoto, K., Nakamura, O., Yoshioka, K., Takeuchi, J., Murata, N., and Shiraishi, Y. (2021). Designing comprehensive cyber threat analysis platform: Can we orchestrate analysis engines? In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE.
- Tanaka, A., Han, C., Takahashi, T., and Fujisawa, K. (2021). Internet-wide scanner fingerprint identifier based on TCP/IP header. In *2021 Sixth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE.