IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 23, 2021, accepted January 18, 2022, date of publication January 25, 2022, date of current version February 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3145966

Dark-TRACER: Early Detection Framework for Malware Activity Based on Anomalous Spatiotemporal Patterns

CHANSU HAN[1], (Member, IEEE), JUN'ICHI TAKEUCHI[2], (Member, IEEE), TAKESHI TAKAHASHI[1], (Member, IEEE), AND DAISUKE INOUE[1], (Member, IEEE)

[1]National Institute of Information and Communications Technology, Koganei 184-8795, Japan
[2]Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan

Corresponding author: Chansu Han (han@nict.go.jp)

This work was supported by the Ministry of Internal Affairs and Communications, Japan, conducted under a Contract of MITIGATE among Research and Development for Expansion of Radio Wave Resources (JPJ000254).

ABSTRACT As cyberattacks become increasingly prevalent globally, there is a need to identify trends in these cyberattacks and take suitable countermeasures quickly. The darknet, an unused IP address space, is relatively conducive to observing and analyzing indiscriminate cyberattacks because of the absence of legitimate communication. Indiscriminate scanning activities by malware to spread their infections often show similar spatiotemporal patterns, and such trends are also observed on the darknet. To address the problem of early detection of malware activities, we focus on anomalous synchronization of spatiotemporal patterns observed in darknet traffic data. Our previous studies proposed algorithms that automatically estimate and detect anomalous spatiotemporal patterns of darknet traffic in real time by employing three independent machine learning methods. In this study, we integrated the previously proposed methods into a single framework, which we refer to as *Dark-TRACER*, and conducted quantitative experiments to evaluate its ability to detect these malware activities. We used darknet traffic data from October 2018 to October 2020 observed in our large-scale darknet sensors (up to /17 subnet scales). The results demonstrate that the weaknesses of the methods complement each other, and the proposed framework achieves an overall 100% recall rate. In addition, *Dark-TRACER* detects the average of malware activities 153.6 days earlier than when those malware activities are revealed to the public by reputable third-party security research organizations. Finally, we evaluated the cost of human analysis to implement the proposed system and demonstrated that two analysts can perform the daily operations necessary to operate the framework in approximately 7.3 h.

INDEX TERMS Anomalous synchronization estimation, darknet, malware activity, spatiotemporal pattern.

I. INTRODUCTION

In recent years, an increasingly large number of indiscriminate cyberattacks have been observed on the Internet, and it is therefore becoming increasingly costly to analyze these attacks. To maintain security of the Internet, it is necessary to quickly recognize global cyberattack trends, specify their causes, devise countermeasures, and alert the world of the details of the threat. *For this purpose, it is important to detect the indiscriminate scanning attack activities caused by malware at an early stage before a particular attack becomes a pandemic.*

However, it is challenging to identify malware scanning attacks among the massive amount of benign traffic in regular networks. Therefore, we adopted unused IP address spaces (darknets). The term "darknet" refers to observation networks, also known as "network telescopes," and should not be confused with anonymous communication networks such as `Tor`. In the darknet, legitimate communication (noise) does not occur; therefore, indiscriminate scanning communication (signal) is observed more noticeably. Thus, the signal-to-noise ratio is high. This makes it an effective way to identify trends and tendencies in global cyberattacks.

The associate editor coordinating the review of this manuscript and approving it for publication was Kashif Saleem.

However, the volume of traffic observed in the darknet is increasing each year exponentially. Moreover, there are many communications whose intentions are unknown, as only the initial communications are observed. For example, in a darknet, we observe numerous independent cyberattacks occurring simultaneously, as well as many communications that are unrelated to attacks, such as scanning activities that are conducted for benign investigation purposes, communications with unknown causes, and misconfigured communications. *As a research target, we should distinguish such noisy communications from malicious attack communications in detail.*

Devices infected with similar malware, that is, ones which share scanning modules, tend to scan in a similar spatiotemporal pattern to compromise new infection targets [1]. Such a tendency is also observed on the darknet [2]. Here, the distributions of source hosts and destination ports for packets observed in a certain period are referred to as spatial features. The features observed in the temporal variation of these spatial features are thus referred to as spatiotemporal patterns. The hosts and destination ports that send packets with similar spatiotemporal patterns are then referred to as being synchronized. *Even in case of small-scale infection activity of malware, a high degree of synchronicity is expected to occur in the associated spatiotemporal patterns, and early detection of malware activity can be realized by estimating the synchronicity and detecting anomalies.*

In our previous studies, we focused on such synchronization and attempted to detect potential malware activities by estimating the group of source hosts with high synchronization in their spatiotemporal patterns on a large-scale darknet. We adopted the following three different machine learning methods in this study: *Graphical Lasso* [3], nonnegative matrix factorization (*NMF*) [4], and nonnegative Tucker decomposition (*NTD*) [5] to estimate the synchronization of spatiotemporal patterns from packet counts by spatial feature per unit time in darknet traffic data. The *Graphical Lasso* algorithm can sparsely estimate conditionally independent variable pairs that are not synchronous from a covariance matrix. The *NMF* and *NTD* algorithms can decompose synchronous latent frequent patterns from data matrices or tensors into superpositions of multiple groups. We previously proposed the following different methods to estimate the synchronization in real time to automatically use the aforementioned algorithms and detect the source host space groups that show abnormal synchronization: *Dark-GLASSO* [6], [7], *Dark-NMF* [8], and *Dark-NTD* [9].

In our previous studies, we confirmed that each method is capable of detecting malware activities well. However, we did not comparatively evaluate the methods and examine their early malware activity detection performance. In this study, we first modularized the three previously proposed methods and integrated common components such as feature extraction and alert issuing into a single framework. We refer to this integrated framework *Dark-TRACER*. As the main challenge, we conducted two experiments on *Dark-TRACER*—one is to evaluate the quantitative detection performance, and the other is to evaluate the feasibility of early detection. In the first experiment, to quantitatively evaluate the detection performance of malware activity, we used the ground truth of reliable malware activity in October 2018, which was manually created, and performed parameter tuning to minimize false negatives and false positives in each module. Although we have previously presented the evaluation results of a conventional method *ChangeFinder* [10] and the proposed modules *Dark-GLASSO* and *Dark-NMF*, we evaluate *Dark-NTD* for the first time using the same criteria. In the second experiment, we manually generated a new ground truth of events (from June 2019 to October 2020) that clearly shows the time of infection spread of malware activities and used it to evaluate the feasibility of the proposed framework for early detection.

As a result, *Dark-GLASSO*, *Dark-NMF*, and *Dark-NTD* achieved 97.1%, 100%, and 97.1% `recall`, respectively. We also identified the pros and cons of each module and found that the integration of all the proposed modules into a single framework, *Dark-TRACER*, complemented each individual module's weaknesses. In addition, the results of the early detection feasibility evaluation show that *Dark-TRACER* can detect threats 153.6 days earlier than when the threats were revealed to the public by reputable third-party security research organizations. We also assessed the human analysis cost and found that daily operation with two analysts could be performed in an average of 7.3 h, assuming that one analyst requires 15 min of analysis time per port.

In summary, this study afforded the following contributions:

- We integrated our three prior methods (modules) into a single framework, *Dark-TRACER*. To the best of our knowledge, our approach is the first method that focuses on the synchronization of spatiotemporal patterns of the darknet traffic. *Dark-TRACER* can detect malware activities that show anomalous synchronization.
- This work is also the most advanced practical study that quantitatively evaluated the detection performance of malware activities and the feasibility of early detection.
- We found that *Dark-TRACER* complements the weaknesses of each module, and achieves a 100% recall rate. In addition, the results demonstrate that *Dark-TRACER* detects threats on average 153.6 days earlier than when the threats are revealed to the public. We also demonstrated that two analysts can conduct the necessary daily operations of the framework in approximately 7.3 h.

Currently, *Dark-TRACER* is being implemented in real-world contexts for actual operation. It is expected to provide information on detected global malware activities to organizations such as the Computer Security Incident Response Team (`CSIRT`) and the Security Operation Center (`SOC`), and to assist in their ability to implement prompt countermeasures such as investigating the causes and conducting detailed analysis.

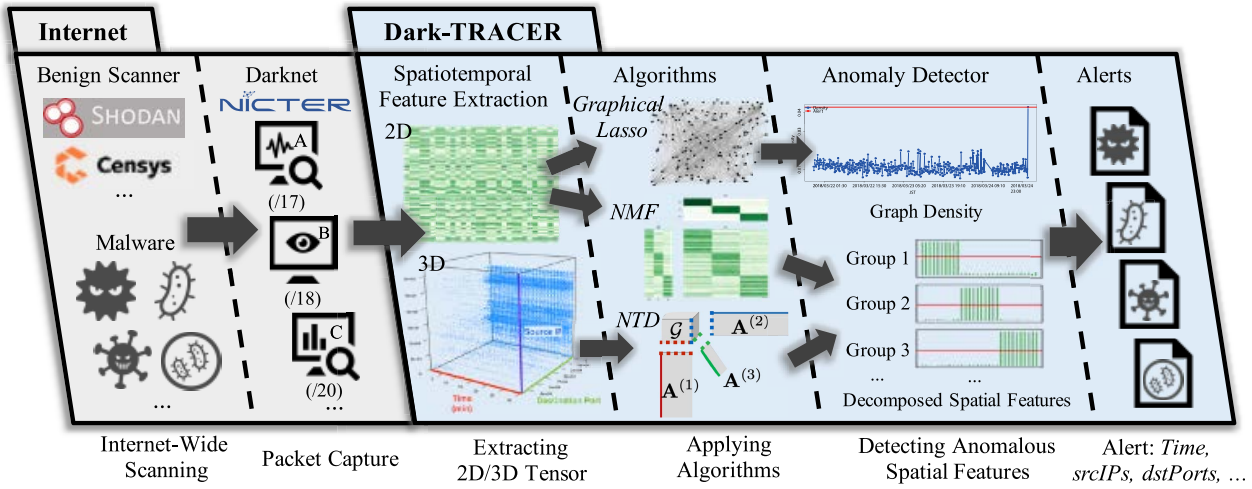

FIGURE 1. Illustration of the framework of *Dark-TRACER*.

The remainder of the paper is organized as follows. The proposed framework, *Dark-TRACER*, and its three modules are presented in Section II. In Section III, we present the methodology and results from the parameter tuning and quantitative evaluation experiments on the detection performance of malware activities for each module. Section IV describes the feasibility evaluation of the proposed method for the early detection of malware activities. In Section V, we discuss the advantages of *Dark-TRACER* through a comprehensive comparison of each proposed module, consideration of the likely adversarial attacks, ways to reduce false-positive alerts, and practical operation methods. Section VI provides a summary of related work on darknet measurement analysis, malware activity detection, and investigative scanners, and we conclude the paper in Section VII.

II. PROPOSED FRAMEWORK

The overall framework of *Dark-TRACER* is shown in Fig. 1. Three algorithms, *Graphical Lasso* [3], *NMF* [4], and *NTD* [5], are used to estimate the synchronicity of spatiotemporal features, and the modules which incorporate these algorithms are referred to as *Dark-GLASSO* [6], [7], *Dark-NMF* [8], and *Dark-NTD* [9], respectively, to distinguish them.

The following advantages over existing malware activity detection methods can be achieved by focusing on synchronicity: 1) We can reduce the effect of benign noise communication in the darknet traffic and highlight the malicious communication. 2) In addition, malware activities that are difficult to trace by conventional manual operations, such as threats that are small-scale, orchestrated, or have no visible explicit spikes, can be captured before the malware infection becomes widespread by detecting anomalously synchronized spatial features. 3) Finally, if a malware activity is found to be synchronized with other malware activities at a time when the scale of infection is small (i.e., before it spreads in earnest), it can be detected at that early stage.

The pseudocode of *Dark-TRACER* framework is presented in Algorithm 1. The parameters are described in this section and Section III-C. For more specific details of the three algorithms employed in this study, the original paper reporting on each algorithm [3]–[5] or our previous works [6]–[9] may be referred to. Based on Fig. 1 and Algorithm 1, the modules are described in greater detail in Algorithm 1.

A. DATA OBSERVATION

Dark-TRACER targets darknet traffic data for analysis. As mentioned previously, the darknet has the advantage of a high signal-to-noise ratio, because regular communication (noise) is not typically observed there, and indiscriminate scanning communication (signal) is monitored in abundance. However, not all communications that are observed in the darknet are malicious communications caused by malware. Among the totality of communications observed in the darknet, some communications are not related to attacks, such as scanning activities for investigation purposes, such as `Shodan` and `Censys` [11],[1] unexplained communications, and misconfigured communications. *Dark-TRACER* is a framework that detects intrinsic attacks and malware activities by ignoring and eliminating such noisy communications.

We have implemented a large-scale darknet observation system, the `NICTER` project,[2] which aims to understand global trends in indiscriminate cyberattacks. Darknet observation systems (sensors) have been installed in several countries and organizations, and approximately 300,000 IP addresses are currently being monitored. The observed data of these darknet sensors differ slightly depending

[1]https://www.shodan.io/ and https://censys.io/
[2]https://www.nicter.jp/en

Algorithm 1 The Framework of *Dark-TRACER*

Require: Common: T, M, t, `sensor` // *Dark-GLASSO*: $\gamma, \lambda, K, \theta$ // *Dark-NMF*: r, α, β, f // *Dark-NTD*: $\tilde{R}_n, R_n$, `epochs`, `th`
Ensure: *Alerts*
while Every t seconds in darknet `sensor` **do**
  /* Data Observation (Section II-A) */
  Darknet traffic data for T s is newly updated, then preprocess it.
  /* Spatiotemporal Feature Extraction (Section II-B) */
  Generate $V_h \in \mathbb{N}_0^{M \times N_h}$, $V_p \in \mathbb{N}_0^{M \times N_p}$, $V_{hp} \in \mathbb{N}_0^{M \times N_h \times N_p}$
  /* Algorithms and Anomaly Detection (Section II-C, II-D) */
  /** *Dark-GLASSO* **/
  if $N_h > \gamma$ **then** $V_h \leftarrow$ `random_sampling`(V_h), $N_h \leftarrow \gamma$ **end if**
  Precision matrix $\Sigma_\lambda^{-1} \in \mathbb{R}^{N_h \times N_h} \leftarrow$ `graphical_lasso`(V_h, λ)
  $d_\lambda \in \mathbb{R} \leftarrow$ `graph_density`(Σ_λ^{-1})
  $outliers_1 \leftarrow$ `anomaly_detection`(d_λ, K, θ) in *Dark-GLASSO*
  /** *Dark-NMF* **/
  $W \in \mathbb{R}^{M \times r}$, $H \in \mathbb{R}^{r \times N} \leftarrow$ `NMF`$(V_h$ or $V_p, r)$
  $outliers_2 \leftarrow$ `anomaly_detection`(W, α, β, f) in *Dark-NMF*
  /** *Dark-NTD* **/
  $outliers_3 \leftarrow$ `NULL`
  for `epochs` **do**
    $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)} \leftarrow$ `LRA-NTD`$(V_{hp}, \tilde{R}_n, R_n)$
    $outliers_3 \leftarrow outliers_3 |$ `anomaly_detection`$(\mathcal{G}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}$,`th`$)$ in *Dark-NTD*
  end for
  /* Issuing Alerts (Section II-E) */
  $outliers \leftarrow outliers_1 | outliers_2 | outliers_3$
  $Alerts \leftarrow$ `issuing_alerts`$(outliers)$
end while

on their geographical location and the scale of observation. For this reason, *Dark-TRACER* analyzes each sensor separately.

Next, as a data preprocessing step, *Dark-TRACER* analyzes only TCP-SYN packets because TCP packets other than TCP-SYN that reach the darknet are not considered to be attack scans. In addition, the upper 16 bits of the IP address are adopted as the unit of the source host. This means that hosts are aggregated on a regional or organizational level. Finally, to highlight the observation of unknown malware activities, we excluded well-known and frequently observed threat ports.

B. SPATIOTEMPORAL FEATURE EXTRACTION

First, we prepared darknet traffic data for a certain period (T seconds). We assumed that N_h unique numbers of source hosts and N_p unique numbers of destination ports were observed in the darknet traffic data. Then, at a sampling interval of T/M seconds, the number of packets was counted for each source host or destination port, and these are referred to as the spatial feature variables. Here, M is a hyperparameter. From the above, three types of tensors representing spatiotemporal features were generated from the observed data: $V_h \in \mathbb{N}_0^{M \times N_h}$, $V_p \in \mathbb{N}_0^{M \times N_p}$, and $V_{hp} \in \mathbb{N}_0^{M \times N_h \times N_p}$ ($\mathbb{N}_0 = \{0, 1, 2, \cdots\}$). This feature extraction was processed in real time and sequentially every t seconds.

C. APPLYING ALGORITHMS

This section briefly introduces the main characteristics of the sparse structure learning algorithm *Graphical Lasso* [3] and the two tensor decomposition algorithms, Nonnegative Matrix Factorization (*NMF*) [4] and Nonnegative Tucker Decomposition (*NTD*) [5]. In addition to the above algorithms, *Dark-TRACER* can be applied with other methods to estimate the synchronization of spatiotemporal features, but an anomaly detection method that is appropriate for the method must be considered.

1) GRAPHICAL LASSO

The *Graphical Lasso* (package name: *glasso*[3]) algorithm is a sparse structure learning method that can calculate the ''intrinsic relationships'', rather than spurious correlations, between variables. Here, ''no intrinsic relationship'' between two variables is equivalent to the conditional independence of the two variables given the other variables. In a Gaussian graphical model, which is a structural learning model that assumes a multivariate Gaussian distribution, the above problem can be considered as the problem of estimating a precision matrix (i.e., an inverse covariance matrix). *Graphical Lasso* uses maximum likelihood estimation with a ℓ_1 regularization term to obtain a sparse precision matrix, thereby introducing sparsity into the relationship between variables.

The obtained precision matrix can be represented as an undirected graph, as shown in the *Graphical Lasso* section of Fig. 1. The node set represents the set of variables, and the edge set represents the ''presence or absence of a relationship'' between the variables. In other words, when there is no relationship between variables, no edges are drawn between the nodes corresponding to those variables. Alternatively, if there is a relationship, an edge is drawn. *Graphical Lasso* has often been applied to the field of anomaly detection. *Graphical Lasso* has been applied to a wide range of real-world problems, such as outlier detection [12], [13] based on the relationship of the candidate outlier with the other variables, such as in *Dark-GLASSO*, and for detecting changes in a graph structure [14], [15].

a: DARK-GLASSO MODULE

Dark-GLASSO uses *Graphical Lasso* to estimate and graph the intrinsic relationship between spatial feature variables from a spatiotemporal feature matrix (V_h or V_p). This can be interpreted as a representation of the synchronization between the variables.

2) TENSOR DECOMPOSITION

Tensor decomposition is a method of decomposing latent frequent patterns from a matrix or tensor into a superposition of multiple groups. Several models have been proposed depending on the rank of the tensor and the decomposition method. Tensor decomposition has been

[3]https://cran.r-project.org/web/packages/glasso/

applied in a variety of fields, such as recommendation systems in the marketing domain [16], feature extraction in electroencephalograms [17], image classification [18], and foreground filtering and activity detection in videos [19].

The tensor data generated in Section II-B does not accept negative values. To make the decomposition results realistic and interpretable, we employed the tensor decomposition methods *NMF* and *NTD* with non-negative constraints. *NMF* is a decomposition method for rank-2 tensors (matrices), whereas *NTD* is a decomposition method for rank-D tensors (in this case, $D = 3$). *NTD* can be regarded as an extension of *NMF* to higher dimensions. Next, the application process of the method is briefly explained for each module.

a: DARK-NMF MODULE

As shown in the *NMF* part of Fig. 1, *NMF* is a method of approximate decomposition of a non-negative matrix $V \in \mathbb{N}_0^{M\times N}$ into a product of two non-negative factor matrices $W \in \mathbb{R}^{M\times r}$, $H \in \mathbb{R}^{r\times N}$ ($V \approx WH$). Here, r is the number of basis vectors, and refers to the number of patterns to be decomposed ($r < N, M$). The *NMF* minimizes the error function $||V - WH||_F^2$ (s.t. $W \geq 0$, $H \geq 0$) (Frobenius norm). Although several approximate decomposition algorithms have been proposed for *NMF*, we employed the most well-known multiplicative update algorithm proposed by Lee *et al.* [4]. In this algorithm, the initial values of W, H are given, and the optimization is performed by alternately updating W, H until the error function is minimized. In *Dark-NMF*, the values of the singular value decomposition were taken as the initial values.

In summary, *Dark-NMF* approximately decomposes a spatiotemporal feature matrix (V_h or V_p) into two factor matrices W, H using *NMF*. The decomposed matrices contain the same number of potentially synchronous groups of spatiotemporal feature variables as the number of bases. Each component of the two decomposed matrices W, H can be interpreted as follows:

- W: ***Temporal features.*** Each basis vector represents a temporal traffic pattern of a different type.
- H: ***Spatial features of the source host or destination port.*** The source hosts or the destination port numbers corresponding to the indices of each basis vector are presumed to have synchronous communication.

b: DARK-NTD MODULE

As shown in the *NTD* part of Fig. 1, *NTD* decomposes a rank-D tensor into one small tensor and several matrices. *Dark-NTD* works with rank-3 tensors and decomposes them each into one small tensor $\mathcal{G}$ and three matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}$. The tensor $\mathcal{V} \in \mathbb{R}^{I_1\times I_2\times I_3}$ can be decomposed using the decomposition equation as $\mathcal{V} \approx \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}$. Here, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n\times R_n}$ ($n \in \{1, 2, 3\}$), $\mathcal{G} \in \mathbb{R}^{R_1\times\cdots\times R_3}$, the scalars I_1, I_2, I_3 denote the length of each axis (mode), and the symbol $\times_i$ denotes the product in the mode-i direction, which multiplies the tensor by the matrix in mode i. The factor matrix $\mathbf{A}^{(n)}$ can be regarded as an extracted feature for mode n. The tensor $\mathcal{G}$ is referred to as a core tensor and represents the weights of the basis vectors of each mode and the strength of the relationship. In addition, R_1, R_2, R_3 are the ranks, which determine how many basis vectors are extracted for each mode and can be interpreted as the number of frequent patterns.

The *NTD* algorithm minimizes the error function $||\mathcal{V} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}||_F^2$ (Frobenius norm). The optimization is performed by alternately updating $\mathcal{G}$ and $\mathbf{A}^{(n)}$. However, when the tensor to be decomposed, $\mathcal{V}$, becomes large, the decomposition becomes practically impossible because a massive amount of memory and computation is required to perform the exact calculation. To address this problem, *Dark-NTD* utilizes the fiber sampling tensor decomposition (FSTD) [20] to perform a low-rank approximation of the tensor $\mathcal{V}$ in advance. Based on the results of this low-rank approximation, *LRA-NTD* [21], which efficiently approximates *NTD*, is applied to save memory and accelerate the decomposition process without reducing its precision. For details of this acceleration, please refer to Kanehara *et al.*'s previous paper on *Dark-NTD* [9].

In summary, *Dark-NTD* utilizes the accelerated *NTD* algorithm to decompose the rank-3 spatiotemporal feature tensor V_{hp} into a core tensor $\mathcal{G}$ and three matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}$. The results of the approximate decomposition contain the same number of potentially synchronous groups of spatiotemporal feature variables as the number of bases in each matrix. Each component of this decomposition result can be interpreted as follows:

- $\mathbf{A}^{(1)}$: ***Temporal features.*** Each basis vector represents a temporal traffic pattern of a different type.
- $\mathbf{A}^{(2)}$: ***Source host spatial features.*** The source hosts corresponding to the indices of each basis vector are assumed to be synchronized and in the same group.
- $\mathbf{A}^{(3)}$: ***Destination port spatial features.*** The port numbers corresponding to the indices of each basis vector are assumed to be received from the same group.

D. ANOMALY DETECTION

In this section, we present a method for detecting anomalies in spatial feature variables based on the application results of each algorithm, module by module.

1) ANOMALY DETECTION IN DARK-GLASSO

From a graph of the calculated precision matrix, the degree of synchronization between the variables is quantified by the graph density $|E|/N(N-1)$. Here, $|E|$ is the number of elements in the edge set, and N is the number of spatial feature variables. The closer the graph density is to 1, the more strongly all variables are related to each other. Graph density is also referred to as its sparsity. The graph density value is calculated from observation data every T seconds in a continuous period and is recorded sequentially. Outlier

detection is performed when time-series data of the graph density value are collected for a period of fixed window size (K). First, two variances are calculated: one when the largest element in the time-series data is excluded and the other when it is not excluded. If the ratio of the two variances exceeds a threshold θ, it is considered an outlier and is deleted from the time-series data. The outliers are identified successively by the next largest element until they no longer exceed the threshold θ. If there are no outliers and the data size exceeds the fixed window size (K), the oldest data are deleted in chronological order. From the above, it is possible to determine the periods which have anomalous graph density values as compared to the other periods.

2) ANOMALY DETECTION IN DARK-NMF

Because the scales of the matrices W, H are not unique, we first normalize them. To ensure that the sum of each column of W is aligned to 1, we normalize W, H as $W = W\Lambda$, $H = \Lambda^{-1}H$ using a diagonal matrix $\Lambda \in \mathbb{R}^{r\times r}$, whose diagonal component is an inverse of the sum of each column. The elements of the normalized H are scaled to actual observed packet counts. Spatial features with values of H less than 1 are considered inactive features in the corresponding basis vector. For such active spatial features, if there are more than β features that exceed α (%) of the maximum value of elements (maximum number of packets), active spatial features are judged as anomalies. In addition, when judging anomalous spatial features, a parameter $f \in \{0, 1\}$ is utilized to determine whether to treat all active spatial features or only those that are more anomalous. From the above, we can determine the anomalous spatial features of a specific period.

3) ANOMALY DETECTION IN DARK-NTD

If there are two or more host spatial features that exceed the threshold value for $\mathbf{A}^{(2)}$, the group of hosts is considered to have synchronized activities and its IP addresses are recorded. In addition, $\mathcal{G}$ and $\mathbf{A}^{(3)}$ are utilized to identify the destination port features through which this group of hosts communicated. From $\mathcal{G}$, we identify a port group of $\mathbf{A}^{(3)}$ that is linked to a group of hosts that have been determined to have synchronous activity from $\mathbf{A}^{(2)}$. In the identified port group, the destination port features that exceed the threshold are determined to be the targeted ports in the synchronized host group of $\mathbf{A}^{(2)}$. We can determine the anomalous host groups and their targeted ports in a specific period.

E. ISSUING ALERTS

The final process collects information that has been determined to be anomalous from each module and outputs an alert in a uniform format. For *Dark-GLASSO*, we used the entire darknet traffic data for a period that has been identified as anomalous. For *Dark-NMF*, we used the data for the spatial features identified as anomalous. If numerous source hosts sent many packets to a specific destination port, we aggregated the information regarding the time, destination port numbers, and source hosts and issued an alert. Finally, *Drak-NTD* issued an alert directly using the anomalous host group, targeted port, and time information found in the anomaly detection step.

III. EVALUATING QUANTITATIVE COMPARISON OF DETECTION PERFORMANCE

We evaluated the performance of each proposed module and describe the results of two different experiments to demonstrate the relationships between modules and their practicality. In the first experiment in this section, we quantitatively evaluated the ability of each module to accurately detect malware activities. In Section IV, the second experiment evaluated the feasibility of the early detection of malware activities. Darknet traffic was preprocessed using `tcpdump` and passed to *Dark-TRACER*, implemented in the `R` language. All experiments were conducted in a unified manner in Japan Standard Time, with CPUs running on AMD RYZEN TR 2990WX and 128GB memory.

In this experiment, we manually gathered TCP ports for which malware activities were clearly observed in October 2018 and generated the ground truth for a total of 35 TCP ports. This ground truth evaluation aimed to determine a hyperparameter set that minimized the number of false negatives, even if there were some false positives in each module, and evaluated the detection accuracy at that time. The conventional method, *ChangeFinder*, and the proposed modules, *Dark-GLASSO* and *Dark-NMF*, have already been tested and the results of those evaluations have been published [7], [8], whereas *Dark-NTD* was now evaluated for the first time using the same criteria. The following subsection describes the details of the dataset, the parameter tuning of *Dark-NTD*, and the comparison results from each module.

A. DETAILS OF DATASET

The dataset and the ground truth for evaluation were the same as those used in the previous reports for *Dark-GLASSO* and *Dark-NMF* and are publicly available.[4] Specifically, we employed data from eight darknet sensors `A` to `H`, which are located around the world and have different observation scales. The observation scale of each sensor ranges from approximately 30,000 IP addresses (/17 subnet) to approximately 2,000 IP addresses (/21 subnet), for a total of approximately 80,000 IP addresses in the darknet observation network. The period of data used in the experiment was in the month of October in 2018. The average number of packets per day for sensor `A`, which has the largest observation scale, was 81.4 M, and the data size was 5,605 MB. To highlight the observation of unknown malware activities, the following 11 known and constantly observed TCP ports were excluded during preprocessing: *22, 23, 80, 81, 445, 1433, 2323, 3389, 5555, 8080, 52869*.

[4]https://csdataset.nict.go.jp/darknet/

Next, regarding details of the ground truth, Table 1 shows the TCP ports where malware activities were monitored and the characteristics of the malware activities by threat type. The threat types were primarily classified into Internet of things (IoT) malware such as `Mirai`, `Hajime`, and `HNS` (Hide and Seek), vulnerabilities related to router manufacturers, and vulnerabilities related to other off-the-shelf service protocols.

As a fingerprint, or key identifier, of `Mirai`, it is commonly known that the sequence number matches the destination IP address in the SYN packet [22], [23]. A fingerprint of `Hajime` is that its window size is fixed at *14600*, and an upper or lower 1 byte of the sequence number is 0. A common feature of router vulnerability types is that there are many cases wherein each router manufacturer's login screen was confirmed when an HTTP connection was made to the source hosts that sent the scan. Cohen *et al.* [24] also identified that ports *5379, 6379, and 7379* were observed during the same period. Please refer to that previous paper for more details, including time-series graphs of the unique host counts of malware activities in this ground truth.

B. PARAMETER TUNING IN DARK-NTD

In this section, we describe how to tune the following five hyperparameters in the *Dark-NTD*.

1) `sensor`: which darknet sensor is used for the observed data
2) $\tilde{R}_n$: Number of bases in *FSTD*, a low-rank approximation method for acceleration.
3) R_n: Number of bases in *NTD*.
4) `epochs`: How many times the calculation for the same data is repeated
5) `th`: Threshold for alert determination

The above five hyperparameters are tuned by grid search.

The search range and interval include our long-term empirical rules. For the `sensor`, we compared the performances of selecting one of the eight darknet sensors against the use of all eight sensors. Next, the larger the number of bases $\tilde{R}_n$ in *FSTD*, the better is the low-rank approximation of the original tensor. In addition, $\tilde{R}_n$ should be set to be larger than R_n, the number of bases in *NTD*. In this grid search, we worked within the range of $\tilde{R}_n \in \{25, 49, 81, 121\}$ and $R_n \in \{3, 5, 8\}$. Furthermore, because the initial values of *FSTD* and *NTD* are randomly chosen, the calculation results are not unique. Therefore, we need to know how many times the same data can be iterated to obtain a stable and sufficient accuracy. In this tuning experiment, we iterated `epochs` up to 15 times. Finally, for alert determination thresholding `th`, we worked within a fixed range of $\{0.05, 0.1, 0.2, \cdots, 0.9\}$ and an adaptive method called "Otsu's thresholding method [32]," which is a commonly used image thresholding algorithm.

Here we describe the results of the above five parameter tuning. It was not practical to tune all five parameters simultaneously, because the number of combinations would be immense. As an evaluation strategy, we divided the parameters into two groups: $\tilde{R}_n, R_n$, which is directly related to *NTD*, and `sensor`, `epochs`, and `th`, which are not. Because `sensor`, `epochs`, and `th` have a more significant impact on detection performance, we first roughly tuned the detection accuracy with these three parameters and then fine-tuned it with $\tilde{R}_n, R_n$. In this experiment, the spatiotemporal feature extraction in Section II-B was performed by generating and using tensors V_{hp} for October 2018 with the observation time unit T set to 1,800 s, the sampling interval M set to 30, and the online processing time unit t set to 600 s.

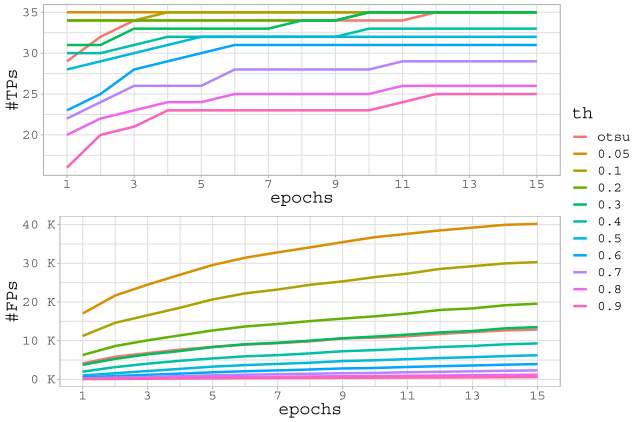

FIGURE 2. Results of true positives (`TPs`) and false positives (`FPs`) for each threshold (`th`) when all eight sensors were utilized. (horizontal axis: `epochs`).

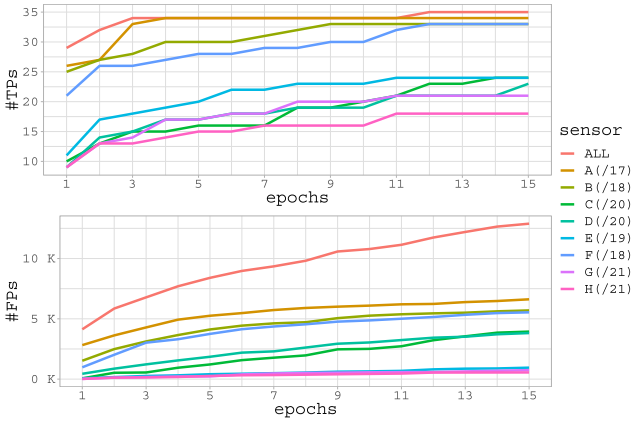

FIGURE 3. Results of true positives (`TPs`) and false positives (`FPs`) for each `sensor` when the threshold (`th`) was Otsu's thresholding method. (horizontal axis: `epochs`).

1) TUNING EVALUATION OF SENSOR, EPOCHS, AND TH

We evaluated the tuning of `sensor`, `epochs`, and `th` after fixing the values to $\tilde{R}_n = 25$, $R_n = 5$, which were empirically used in an earlier study [9]. The results of the evaluation are presented in Figs. 2 and 3. The horizontal axis represents the number of `epochs`, and the vertical axis represents the number of true positives (`TPs`) and false positives (`FPs`) of the port numbers. When the number of `TPs` is close to 35 and the number of `FPs` is low, we can observe that the detection

TABLE 1. Characteristics of malware activities observed on our darknets in October 2018. Malware activities with similar characteristics are grouped according to TCP ports.

Threat Type	Threat Group	TCP Ports	Characteristics (observed from our darknets)
IoT Malware	Mirai I	*82,83,84,85,88,8000, 8001,8081,8088,8888*	Approximately 1K hosts with the Mirai feature constantly probed our darknets. A spike of 6K hosts was observed on October 22.
	Mirai II	*88,443,8081,8443*	A spike of 7K hosts was observed on October 20 with the Mirai feature. Most packets originated from China, and the window size was fixed at *14100*.
	Mirai III	*444,7547,8010*	We observed spikes at different periods on each port with Mirai feature (< 1K hosts). *7547*/TCP: Most packets originated from Egypt [25]. *444*/TCP: Most packets originated from Greece.
	Hajime, HNS (Hide and Seek)	*5358,9000* (Hajime), *2480,5984* (HNS)	Hajime: Over 1K hosts with the Hajime feature constantly probed our darknets [26]. HNS: Over 2K hosts constantly probed our darknets [27].
Router Vulnerability	MikroTik	*21,25,110,443,8291, 23023,65000*	We observed spikes multiple times for each port from hosts that seemed to be router products of MikroTik (about 6K hosts) [28]. The window size was fixed at *1024*.
	Huawei NUUO, ASUS	*37215* (Huawei), *8181* (NUUO), *8001,8081* (ASUS)	We observed spikes for each port from hosts that seemed to be router products of Huawei, NUUO, and ASUS [29], [30]. They had the features of Mirai.
	BroadCom UPnP	*5431*	We observed regular infection activities targeting the vulnerability of BroadCom UPnP (Universal Plug and Play) in about 100K hosts [31].
Application Vulnerability	5 Vulnerabilities	*1701* (L2TP VPN), *49152* (IPMI/BMC), *5900* (VNC), *2004* (WordPress), *5379,6379,7379* (Redis)	*1701*: A spike of 3K hosts was observed on October 9 (from China). *49152*: A spike of 6K hosts was observed on October 14 (from Egypt, Mirai feature). *5900*: A spike of 4K hosts was observed on October 29 (window size=*8192*). *2004*: A spike of 300 hosts was observed on October 15 (window size=*14600, 29200*). *5379,6379,7379*: Spikes of 1K hosts were observed on October 31 for each port [24].

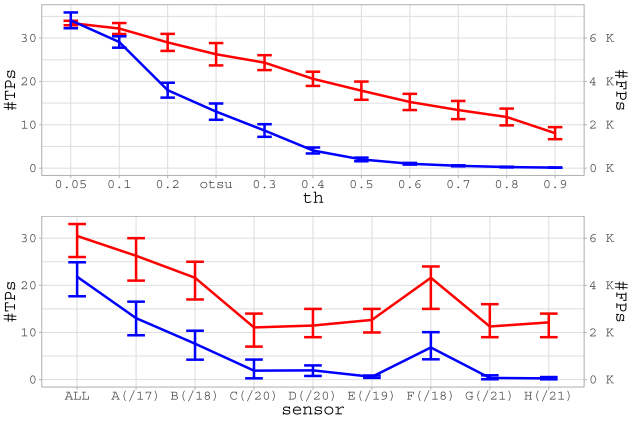

FIGURE 4. Dispersion results of the number of TPs (red) and FPs (blue) for each epoch over 15 iterations. Error bars represent standard deviations. The upper graph shows the results for each th when sensor A is set, and the lower graph shows the results for each sensor when th is Otsu's thresholding method.

accuracy is excellent. Fig. 2 shows the results for each th when all eight sensors are utilized. This indicates that th results in more TPs with fewer epochs while keeping the number of FPs relatively low when using Otsu's thresholding method (otsu) as compared with any other fixed value. Next, Fig. 3 shows the results for each sensor when the th used is Otsu's thresholding method. The results demonstrate that when using only the sensor A, the same number of TPs is achieved with fewer epochs while keeping the number of FPs lower than when using either all sensors in combination, or other sensor alone. Based on the above, we conclude that the best solution is to use only A for sensor, 4 for epochs, and Otsu's thresholding method for th.

Next, we provide a brief discussion regarding each parameter. Fig. 4 shows the dispersion of TPs and #FPs

TABLE 2. Tuning evaluation results of $\tilde{R}_n$, R_n in *Dark-NTD*.

$\tilde{R}_n$ \ R_n	#TPs 3	#TPs **5**	#TPs 8	#FPs 3	#FPs **5**	#FPs 8	Time (s)
25	31	**34**	33	3931	**4932**	4814	61.5
49	33	34	32	4916	5898	6235	148.9
81	32	32	33	6085	7120	6893	328.5
121	32	33	33	6441	7353	7153	638.7

for each epoch over 15 iterations. The red graph is the mean number of TPs, the blue graph is the mean number of FPs, and the error bars represent the standard deviations. From these results, we can conclude that the randomness of the initial value selection of *FSTD* and *NTD* does not dramatically affect the detection performance because a similar number of TPs and #FPs was recorded each time. In terms of the sensor, *Dark-NTD* recorded a good number of TPs for sensors with a large observation scale. Finally, Otsu's thresholding method achieves a similar level of accuracy to a fixed value of 0.3 but has the advantage of adaptively determining a threshold value from the data.

2) TUNING EVALUATION OF $\tilde{R}_n$ AND R_n

In this section, we set sensor, epochs, and th to the values determined above, and then evaluated the tuning of $\tilde{R}_n, R_n$. The results are shown in Table 2. Contrary to expectations, increasing the value of $\tilde{R}_n$ decreased the number of TPs and increased both the number of FPs and the average processing time. This result supports the fact that even at $\tilde{R}_n = 25$, we can sample enough important information (fiber) for low-rank approximations. Therefore, we determined that $\tilde{R}_n = 25$ is appropriate. In the case of R_n, there were no significant differences in the average processing time for any value, and the greatest number of TPs was achieved with $R_n = 5$, so we conclude that $R_n = 5$ is most appropriate.

TABLE 3. Comparative evaluation results of detection performance of malware activities. *Dark-TRACER* is the result of integrating *Dark-GLASSO*, *Dark-NMF*'s SET1, and *Dark-NTD*'s Tuned.

Modules		#TPs	#FNs	#FPs	Recall
ChangeFinder [10]		24	11	0	68.6%
Dark-GLASSO [7]		**34**	**1**	**0**	**97.1%**
Dark-NMF [8]	SET1	**31**	**4**	**9**	**88.6%**
	SET2	35	0	1074	100%
	SET2′	35	0	519	100%
Dark-NTD	CONV [9]	29	6	4131	82.9%
	Tuned	**34**	**1**	**4932**	**97.1%**
	Tuned′	30	5	3026	85.7%
Dark-TRACER		**35**	**0**	**4935**	**100%**

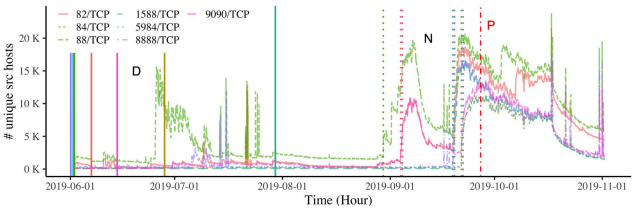

FIGURE 5. The number of unique source hosts per hour on NICTER, where Moobot-related malware activities are observed. A synchronized fluctuation in the number of hosts was confirmed on September 19 and September 21. D: The earliest time detected by *Dark-TRACER*, N: The time observed by NICTER operators, and P: The time when was is revealed to the public.

C. COMPARATIVE EVALUATION RESULTS OF DETECTION PERFORMANCE

In this section, we quantitatively compare and evaluate the detection accuracy of malware activities among the modules. The results are shown in Table 3. *ChangeFinder* [10] is an existing method that was applied in NICTER before proposing each module of *Dark-TRACER*, and it is an algorithm for detecting change points in time-series data with a low computational cost. The *ChangeFinder* algorithm is based on the sequential discounting autoregressive (*SDAR*) forgetting learning algorithm, which calculates only new time series data and reduces the influence of past data by improving the autoregressive model to learn sequentially. We implemented *ChangeFinder* on two types of time-series data: the number of packets and unique source hosts in 10 min. The parameters of each module used in this experiment are described below.

ChangeFinder:
autoregressive order = 2, forgetting parameter = 0.005, smoothing range (two steps) = {10, 5}, threshold for change detection = 3

Dark-GLASSO:
$T = 600$, $M = 12$, $t = 600$, used matrix V_h, $K = 432$, $\theta = 0.98$, $\lambda = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, $\gamma = 1000$

Dark-NMF:
$T = 1800$, M=30, t=600, used matrices (V_h, V_p), $\alpha = 30$, $\beta = 2$, $r = \{1, 2, \cdots, 10\}$, $f = 0$ for SET1, $f = 1$ for SET2

Dark-NTD:
$T = 1800$, $M = 30$, $t = 600$, used tensor V_{hp}, sensor=A, $\tilde{R}_n = 25$, $R_n = 5$, epochs=4, th=Otsu's thresholding

Here, λ in *Dark-GLASSO* is a regularization coefficient for *Graphical Lasso*. Due to the high computational complexity of *Dark-GLASSO*, random sampling was conducted when the number of hosts N_h exceeded γ to maintain real-time performance. All other parameters are explained in Section II.

Next, we explain the notation used in Table 3. SETs in *Dark-NMF* indicates the difference between 0 and 1 settings of f. CONV in *Dark-NTD* is a parameter setting introduced in previous research, and Tuned is the parameter setting determined by tuning in the previous section. The difference between CONV and Tuned is that Tuned has epochs and uses only sensor A. Note that only Tuned in *Dark-NTD* utilizes sensor A, whereas the other modules utilize all eight sensors. #FNs is the number of false negatives, which is #TPs+#FNs = 35. Recall is an evaluation metric calculated as #TPs / (#TPs+#FNs).

Lastly, we describe the symbol [′] attached to SET2′ and Tuned′. The results of SET2 and Tuned show that the number of FPs is very high. The primary cause of false positives is synchronized scans by investigative scanners, such as Shodan and Censys [11]. To address this problem, at least temporarily, we attempted to exclude alerts from investigative scanners by applying a simple rule to the alert results of SET2 and Tuned. The simple rule was: if a large number, or a sequential number, of TCP ports were seen concurrently from the same source hosts in the alerts, those alerts are excluded. The application results of the rule were SET2′ and Tuned′. *Dark-NMF* has an effect of halving the number of FPs while maintaining the number of TPs, whereas *Dark-NTD* does not have such an effect.

From the results of a comparative evaluation in Table 3, Dark-TRACER achieves a recall rate of 100%, although there are some FPs*, by integrating the results of each module.* Next, we examine the characteristics of the ports that are false negative in each module. *ChangeFinder* tends to perform poorly at detecting small host sizes and for short- or long-term constant malware activities. *Dark-GLASSO* and *Dark-NMF* are perform poorly when detecting malware activities with small host sizes. Furthermore, *SET1* tends to be weak in detecting long-term persistent activities. *Dark-NTD* tends to be poor at detecting short-term malware activities. *Overall, the results show that an integration of the three proposed modules can mutually complement the weaknesses of each module.*

IV. FEASIBILITY ASSESSMENT OF EARLY DETECTION

In this section, we assess the feasibility of the early detection of malware activities. The details of the dataset, the

TABLE 4. Details of the ground truth to assess the feasibility of early detection of malware activity. It consists of 33 TCP ports and 12 types of threat events for malware activities observed in 17 months from June 2019 to October 2020, wherein the time of infection spread is clearly discernible. (RCE: *Remote Code Execution*, C&C: *Command and Control*, DDoS: *Distributed Denial-of-Service*, CVE: *Common Vulnerabilities and Exposures*, PoC: *Proof of Concept*).

Threat Event	TCP Port	Monitored Period and Scale at NICTER	Reveal Date	Characteristic of Observed Malware Activities
ECHOBOT [33] (Mirai Variant)	*1220,6666, 9080*	*2019-07-11 16:00, 100→200*	*2019-08-06*	A file named Richard is forced to download. More than 50 *exploits* were found, including RCE.
MOOBOT [34] (Mirai Variant)	*60001*	*2019-06-24 09:00, 50→4K*	*2019-09-27*	Moobot has a distinctive encryption method, C&C communication protocol, and infection pathway different from the original Mirai. It has the feature of sharing C&C and download servers. It actively launches zero-day attacks and DDoS attacks targeting various vulnerabilities and ports.
	9527,34567	*2019-07-11 08:00, 50→8K*		
	81,88,8000	*2019-08-30 00:00, 1K→5K*		
	82,83,85, 8081,9090	*2019-09-04 00:00, 1K→10K*		
	1588,8888	*2019-09-19 00:00, 100→16K*		
	84,5984, 8181,9200	*2019-09-21 11:00, 1K→12K*		
BlueKeep [35] (CVE-2019-0708)	*3389*	*2019-12-13 10:00, 3.5K→4.5K*	*2019-05-14*	CVE has been released by Microsoft. *window size* is fixed to *8192*.
Shenzhen TVT [36]	*4567*	*2019-12-28 22:00, 500→3K*	*2018-04-09*	A vulnerability in the Shenzhen TVT product reported in 2017 appears to have recurred. We observed the same string as then in the honeypot: D79E94C5-70F0-46BD-965B-E17497CCB598. We also confirmed the Mirai feature.
MikroTik [28]	*8291,8728*	*2020-02-05 12:00, 50→1K*	*2018-08-01*	It is persistently targeted, and its *window size* is fixed at *8192*. No Mirai features were observed. We observed payloads that targeted WinBox and the API of MikroTik routers in the honeypot.
Xiongmai [37]	*9530*	*2020-02-11 12:00, 50→8K*	*2020-02-04*	A backdoor vulnerability exists in Xiongmai video recorder devices that listen on port *9530*. We confirmed the Mirai feature.
Hoaxcalls Botnet I [38] (CVE-2020-5722, CVE-2020-8515)	*8089*	*2020-03-30 07:00, 400→700*	*2020-03-24*	As for the *User-Agent* included in attack communications, XTC Botnet was frequently found; thus, it is assumed to be related to Hoaxcalls Botnet. PoC code for the vulnerabilities in Grandstream UCM6200 on 2020-03-24 and Vigor, a router manufactured by Draytek, on 2020-03-30 was disclosed.
Hoaxcalls Botnet II [39]	*9673*	*2020-04-21 04:00, 50→700*	*2020-03-09*	Similar to the above Hoaxcalls Botnet I, XTC Botnet was frequently included as a *User-Agent* in attack communications. On 2020-03-09, various vulnerabilities were disclosed in Cloud CNM SecuManager, a network management software product of Zyxel.
SaltStack [40] (CVE-2020-11651, CVE-2020-11652)	*4505,4506*	*2020-05-13 11:00, 50→300*	*2020-04-30*	On 2020-04-30, the security research organization F-Secure issued a security advisory on vulnerabilities in SaltStack Salt, an open source configuration management framework.
Linksys [41]	*55555*	*2020-06-13 15:00, 300→500*	*2020-06-13*	Communications from Iran and India have caused it and the Mirai feature to be monitored. We observed attack communications that exploited the vulnerability of the Linksys E series routers in the honeypot.
MVPower [42]	*5501*	*2020-07-13 14:00, 50→600*	*2020-07-13*	Communications from Egypt caused it and the Mirai feature to be monitored. Attack communications targeting MVPower DVR in the honeypot.
Oracle [43] (CVE-2020-14882)	*7001,7002*	*2020-10-20 00:00, 800→2.5K*	*2020-10-20*	Since 2020-10-20, we have observed a spike in the number of hosts and packets destined to *7001* and *7002*, which are the default ports of the management console of Oracle's Weblogic server.

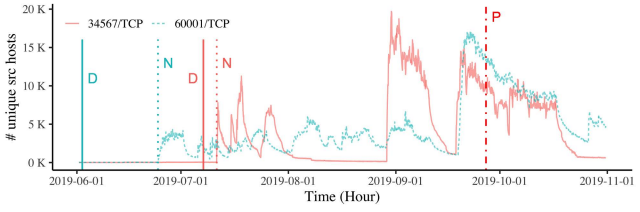

FIGURE 6. The number of unique source hosts per hour on NICTER, where Moobot-related malware activities are observed. Here, the events that do not fluctuate synchronously with the ports in Fig. 5 are shown. D: The earliest time detected by *Dark-TRACER*, N: The time observed by NICTER operators, and P: The time when it was revealed to the public.

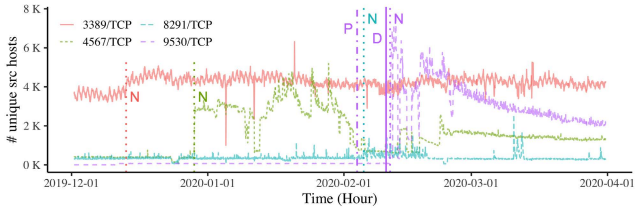

FIGURE 7. The number of unique source hosts per hour on NICTER where the malware activities were observed (December 2019 to March 2020). D: The earliest time detected by *Dark-TRACER*, N: The time observed by NICTER operators, and P: The time when it was revealed to the public.

experimental setup, and the assessment results are described below. This experiment included the general method of cross-validation in time series data; after learning the optimal parameters with past data in section 3, we verified them with future data in this section.

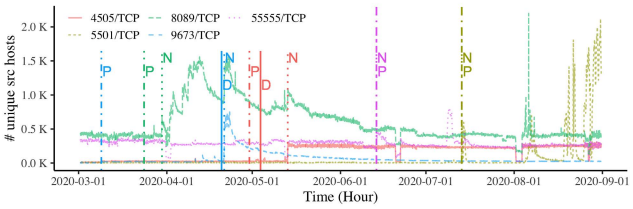

FIGURE 8. Number of unique source hosts per hour on NICTER, where malware activities were observed (March 2020 to August 2020). D: The earliest time detected by *Dark-TRACER*, N: The time observed by NICTER operators, and P: The time when it was revealed to the public.

A. DETAILS OF THE DATASET AND EXPERIMENTAL SETUP

In this experiment, we used data from three sensors A (/17 subnet), B (/18 subnet), and D (/20 subnet), selected by observation scale, instead of all eight sensors. The data period used in the experiment was 17 months, from June 2019 to October 2020. To highlight the observation of unknown malware activity, we excluded known and routinely observed TCP ports by calculating them for each month at each sensor as a preprocessing step.

Next, we describe the details of the ground truth used in the evaluation. In the experiment described in Section 3, the ground truth included many stationary threats whose infection spread period of malware activities was unclear; thus, it was not easy to assess early detection. In this experiment, we manually generated a new ground truth for malware activities observed from June 2019 to October 2020, which represented a set of threats with a clearly identifiable infection spread period. The newly prepared ground truth was based on reports and blog posts published by NICTER's expert operators.[5] Among the malware activities observed by NICTER, we selected malware activities whose origin and characteristics were clear and for which there were references by third parties. As a result, we collected 12 types of threat events on 33 TCP ports. The breakdown of the ground truth is shown in Table 4. The following information was accurately recorded:

- the initial period when NICTER began to observe a rapid increase in the number of packets and hosts on TCP ports related to threats
- the change in the scale of the number of hosts at that time
- the period in which threats were revealed to the public due to references issued by reputable third-party security research organizations (i.e., reveal date)
- characteristics of the threats

Third-party references included recurring activities, such as BlueKeep, ShenZhen TVT, and MikroTik, which are attacks on previously known vulnerabilities.

This ground truth considers not only the type of threat but also its variations, such as the observed infected host size and the persistence/stationarity of threats. For clarity, hourly time-series graphs of the number of unique source hosts observed by NICTER are shown in Figures 5, 6, 7, and 8 for each TCP port. The solid vertical line labeled "D" represents the earliest period detected by Dark-TRACER, the dashed line labeled "N" represents the period observed by NICTER operators, and the dashed line labeled "P" represents the period when was is revealed to the public by reputable third-party security research organizations.

[5]https://blog.nicter.jp/

TABLE 5. The number of early detected ports, late detected ports, overlooked ports (#FNs) and their average numbers of days to detection based on the initial period of observation by NICTER.

Modules	#TPs				#FNs
	Early Detection		Delayed Detection		
	#Ports	#Days	#Ports	#Days	
ChangeFinder	15	82.9	10	-5.1	8
Dark-GLASSO	18	89.3	9	-29.5	6
Dark-NMF	28	122.6	3	-79.1	2
Dark-NTD	29	139.9	2	-25.7	2
Dark-TRACER	**33**	**126.4**	**0**	**NaN**	**0**

TABLE 6. Average number of unique ports by period.

Modules	Number of Unique Ports			
	1 Day	1 Week	1 Month	Entire Period
Dark-GLASSO	6.21	10.87	17.24	66
Dark-NMF	21.40	85.42	250.53	2042
Dark-NTD	39.12	193.39	565.00	3969
Dark-TRACER	**58.49**	**252.41**	**718.71**	**5271**

Figures 5 and 6 are time-series graphs summarizing the partial TCP ports where Moobot-related threats were observed. In Fig. 5, port groups with synchronized fluctuations in the number of unique hosts can be confirmed as belonging to one group on September 19 and one group on September 21, indicating that large-scale Moobot activity was observed during this period. Figure 6 shows events where Moobot features were observed but did not show synchronized fluctuations with the ports in Fig. 5. These ports are related not only to Moobot, but also to the activities of other Mirai variants such as Fbot and Estella. As shown above, Moobot's malware activity is an orchestrated threat that combines multiple activities. Next, Figs. 7 and 8 show time-series graphs of partial TCP ports among threats other than Moobot. Of these partial TCP ports, we found several threats that were difficult to detect by conventional methods or by human efforts, such as threats with no spikes, constant threats, and small-scale threats.

B. ASSESSMENT RESULTS

We now describe the processing results for each parameter with their best parameters, which were the same as in the experimental setup described in Section III-C. We also applied the aforementioned simple rule to exclude alerts caused by investigative scanners. Here, *Dark-NMF* was computed with the parameter SET1. Table 5 shows the number of ports that were detected early, late, or falsely negative, and their average number of days, based on the initial period of NICTER observations. The results show that although there were a few overlooked ports (#FNs) and late

detected ports when considered by the module, *all 33 TCP ports could be detected at an early stage when integrated with Dark-TRACER. In addition, Dark-TRACER was able to detect threats on average 126.4 days earlier than the initial period when threats were first observed by NICTER, and 153.6 days earlier than the period when threats were announced to the public.*

We also investigated how many ports were alerted for each module in this experiment. Table 6 shows the average number of unique ports per period for each module. For the entire 17-month period, *Dark-GLASSO*, *Dark-NMF*, and *Dark-NTD* produced alerts for 66, 2,042, and 3,969 unique ports, respectively. When the proposed modules were integrated into *Dark-TRACER*, the number of unique ports was 5,271. We counted the number of unique ports for each day, week, and month, and the averages are shown in Table 6. For example, *Dark-TRACER* issued alerts for an average of 58.49 ports per day. *Assuming that one analyst requires 15 min of analysis time per port (refer to Section. V-E), two analysts could perform these daily operations in approximately 7.3 h (roughly 14.6 h for a single analyst). It would require approximately 31.5 h for a week and 89.8 h for a month with two analysts.*

From the above two experiments, we found that Dark-TRACER could tune the parameters of each module so that the number of `FNs` *was almost non-existent and could also detect malware activities at a fairly early stage.* As a future challenge, the cost of analysis would be lower if the number of `FPs` could be reduced more precisely. In addition, there is a possibility that expert analysis would disclose other unknown activities, in addition to the malware activities that were selected for the ground truth.

V. DISCUSSION

In this section, we provide a comprehensive discussion and insight into the performance of our framework. First, we demonstrate the advantages of *Dark-TRACER* and provide a comprehensive comparison of each proposed module. Then, we discuss the potential concerns of our approach, such as adversarial attacks and the reduction of false-positive alerts. Finally, we present guidelines for the practical application of *Dark-TRACER*.

A. ADVANTAGES OF DARK-TRACER

As mentioned in the introduction, by focusing on the synchronization of spatiotemporal patterns in darknet traffic, we have the following advantages.

1) TRIMMING UNSYNCHRONIZED AND NOISY COMMUNICATIONS

Distinguishing between non-attack-related and attack-related communications from darknet traffic is a difficult task. Misconfigured or unexplained communications are nuisances that interfere with the interpretation of darknet traffic analysis. In this paper, we focused on the fact that hosts infected with similar malware tend to compromise and scan in a synchronized spatiotemporal pattern. By estimating the synchronicity of spatiotemporal patterns in the darknet traffic and eliminating communications that do not show synchronization from the scope of analysis, *noisy communications are expected to be scraped off, and malicious communications can be highlighted.*

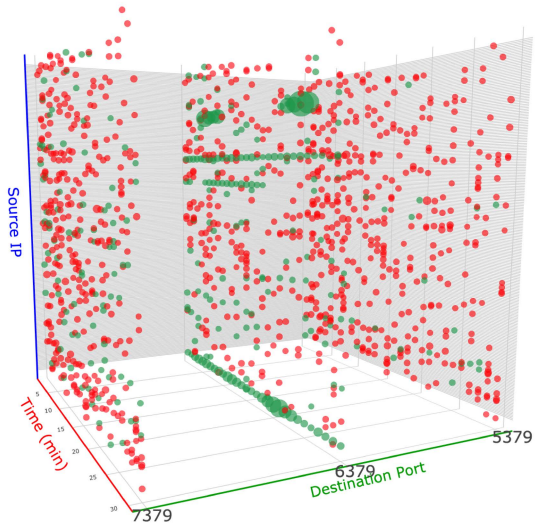

FIGURE 9. A 3D graph visualizing a case of anomalous synchronization of the spatiotemporal patterns detected from the experimental results in Section III. A scatter plot of partial V_{hp} during 18:30–19:00 on October 31, 2018, is visualized. Each of the three axes is a time axis in minutes, a source-host spatial axis, and a destination-port spatial axis, and the plots represent the observed packets (element values in V_{hp}). For the destination-port spatial axis, there are only three points at which anomalous synchronicity was detected—5379, 6379, and 7379. Host IPs are plotted in red if they match on multiple ports within one-minute increments and in green if they do not. The red points are considered to be synchronized communications caused by factors such as malware activities, while the green points are considered to be noise communications.

For understanding, a visualization of the synchronization of the spatiotemporal patterns is shown in Fig. 9. This is an alert of malware activity detected by *Dark-NMF* at `sensor A` during 18:30-19:00 on October 31, 2018, visualizing V_{hp} at that time. The number of packets is plotted in three dimensions: time, source host, and destination port in one-minute increments. Figure 9 shows that the number of red dots indicates the number of communications from the same host to the same destination ports (5379, 6379, 7379/TCP) during that time period. As indicated in Table 1, we observed a scanning attack on the same service `Redis` at these ports. Thus, many red dots appear when the spatiotemporal pattern has anomalous synchronization. In contrast, the green dots can be regarded as noisy communication. It is assumed that *Dark-TRACER* detects anomalies by highlighting the red dots (e.g., malware activity) while eliminating the green dots (noise communication). The red dots (synchronization between spatiotemporal features) do not appear as abundantly as they appeared in Fig. 9 between arbitrary destination ports of ordinary darknet traffic.

TABLE 7. A comprehensive comparison of the proposed modules *Dark-GLASSO, Dark-NMF, Dark-NTD*.

Category		*Dark-GLASSO*	*Dark-NMF*	*Dark-NTD*
Accuracy	#FPs	**Almost None**	Many	Numerous
	#FNs	A Few	**Almost None**	**Almost None**
	Ability of Early Detection	Detection Slightly Delayed	**Detection Almost Early**	**Detection Almost Early**
	Ability of Detection for Small-scale Threats	Very Poor	Poor	**Robust**
	Ability of Detection for Constant Threats	**Robust**	Poor	**Robust**
	Ability of Detection for Short-term Threats	**Robust**	**Robust**	Poor
	Impact of Darknet Sensor Scale	Bigger is Better	**Not Dependent on Scale**	Bigger is Better
Cost	Computational Cost	High	**Low**	High
	Preprocessing	Required	**Not Required**	Required
	Alert Analysis	**Low**	High	Very High
Anomaly Detection	Pattern Decomposition	Not Performed	Performed	Performed
	Necessity of Past Data	Required	**Not Required**	**Not Required**
Spatial Feature	Host Spatial Feature	**Computable**	**Host/Port Spatial Features can be Computed Separately**	**Host/Port Spatial Features can be Computed Simultaneously**
	Port Spatial Feature	Computationally Difficult		

2) DETECTING MALWARE ACTIVITIES THAT ARE CONVENTIONALLY DIFFICULT TO DETECT

Traditionally, malware activities have been detected based on changes in time-series data, such as the number of packets and the number of hosts, using change-point detection algorithms such as *ChangeFinder*, or manually by operators. Until recently, many malware activities were relatively easy to detect by operators, as they were threats that had severe and obvious changes in time-series data, threats with clear spikes, or threats that were simple and had a large scale of infection. However, in recent years, the amount of communication observed in the darknet has increased exponentially and cyberattacks have become more diverse and sophisticated, such as the `Moobot` described in Section IV-A. Such orchestrated threats, which intertwine multiple activities, small-scale threats, threats without explicit spikes, and constant threats, are malware activities that are difficult to detect manually. *However, there is a significant possibility that Dark-TRACER can detect such traditionally hard-to-detect malware activities. Its feasibility is well demonstrated by the evaluation results of early detection in Section IV, wherein various types of malware activities were detected.*

3) EARLY DETECTION OF MALWARE ACTIVITIES IN REAL-TIME

Dark-TRACER is not only capable of detecting traditionally hard-to-detect threats, but also of detecting them early and in real-time. Even when the scale of malware activity is small, if there is overlap in the spatial features (e.g., the distribution of hosts and ports) with another malware activity that has already been detected, and if there is synchronicity in the pattern of the number of packets, there is a high probability that they will be detected together. *This implies that Dark-TRACER can capture the signs of infection before it spreads in earnest. In this way, by checking the overlapping degree of host spatial feature variables between alerts from the same period, it is possible to identify threats that at first glance appear to be different events, but are actually caused by the same malware.* In fact, as shown in Figs. 5 and 6, *Dark-TRACER* can detect orchestrated threats such as *Moobot*, in which multiple activities are intertwined, at an early stage by detecting signs of infection as they spread, even at a small scale.

B. COMPREHENSIVE COMPARISON OF PROPOSED MODULES

In this section, we comprehensively compare the proposed modules *Dark-GLASSO, Dark-NMF, Dark-NTD* in terms of accuracy, cost, anomaly detection method, and spatial features. An overview is given in Table 7, and detailed explanations are provided in order from the top of the list.

1) ACCURACY

First, we discuss the accuracy aspect. In general, there was a trade-off between the number of `FPs` and `FNs`. *Dark-GLASSO* had almost no `FPs` and *Dark-NMF, Dark-NTD* had almost no `FNs`. As for the performance of early detection, Table 5 shows that *Dark-GLASSO* tended to make detections slightly later, but the other modules almost always detected threats early. Next, as mentioned in Section III-C, we examined the characteristics of each module's number of `FNs` ports and considered the types of threats that each module overlooked. *Dark-GLASSO* tended to miss small-scale threats, *Dark-NMF* tended to miss small-scale and constant threats, whereas *Dark-NTD* tended to be weak at detecting short-term threats. The same tendency was confirmed by the experimental results described in Section IV. *Because the modules are complementary to each other, it is recommended to use them in an integrated manner, rather than using only one in isolation.* Table 5 shows that by integrating the modules into *Dark-TRACER*, we can avoid problems of missed or delayed detection. Finally, the accuracies of *Dark-GLASSO* and *Dark-NTD* were low, unless the observation scale of the darknet sensor was large. In contrast, *Dark-NMF* recorded the same level of accuracy for all eight sensors of different sizes used in Section III.

2) COST

Dark-NMF is very computationally inexpensive and does not require any particular preprocessing. In contrast, *Dark-GLASSO* and *Dark-NTD* are computationally expensive and require specific preprocessing. Given the spatial feature sizes N_h and N_p, *graphical lasso* has a cubic time complexity of $O(N_h^3)$, *NTD* has a quadratic time complexity of $O(N_h \cdot N_p)$, and *NMF* has a linear time complexity of $O(N_h)$ or $O(N_p)$.

In this study, *Dark-TRACER* was processed online sequentially at 10-min intervals. Therefore, we adjusted the parameters and preprocessed each module to finish the processing within 10 min for one data slot. *Dark-NMF* did not require any preprocessing. Next, as described in Section III-C, *Dark-GLASSO* performs random sampling preprocessing when the number of hosts N_h exceeds γ. In a previous paper [7], we reported that the output alerts were quite unstable when γ was lower than the expected average value of N_h. *Dark-NTD*, as described in Section II-C, applies *FSTD* [20] to preprocess the tensor $\mathcal{V}$ to a low-rank approximation and preliminarily samples only the essential parts. The larger the number of bases $\tilde{R}_n$ in *FSTD*, the better the low-rank approximation of the original tensor. However, as shown in Table 2, the results of tuning $\tilde{R}_n$ demonstrated that increasing the value of $\tilde{R}_n$ only worsened the accuracy. In all experiments, the processing time of each module was no longer than 10 min (CPU: AMD RYZEN TR 2990WX). For one data slot, *Dark-NMF* took approximately 1 min, *Dark-NTD* approximately 5 min, and *Dark-GLASSO* approximately 7 min.

An important factor in the cost of alert analysis is the number of ports that must be analyzed per unit period. As shown in Table 6, *Dark-GLASSO* has the lowest cost. For *Dark-GLASSO*, only 66 ports needed to be analyzed during the entire experiment in Section IV, whereas the other modules required 30 to 60 times more cost.

3) ANOMALY DETECTION

Next, we discuss methodological differences in anomaly detection. *Dark-NMF* and *Dark-NTD* decompose spatiotemporal features into latent frequent patterns and then perform anomaly detection for each group of decomposed spatial features. This decomposition can be regarded as a favorable condition for detecting local events, but it is also one of the reasons why the number of detected anomalous events (alerts) becomes very large, although it should be noted that the number of alerts can be adjusted by tuning the parameters. In contrast, *Dark-GLASSO* detects anomalies from all spatial features without decomposing the spatiotemporal features. This is a favorable condition for detecting global events and is one of the reasons that the number of anomalous events (alerts) detected is small.

In addition, *Dark-GLASSO* requires K of the past data to perform anomaly detection. Each time we change the value of the parameters or introduce a new sensor, *Dark-GLASSO* has to wait for K periods to obtain the detection results. Other modules, however, do not require waiting in such cases because they do not require past data.

4) SPATIAL FEATURE

Finally, we discuss differences in the application of spatial features. In *Dark-TRACER*, two types of spatial features can be handled: host space and port space. For darknet traffic in a short unit time, the size of the port space N_p tends to be larger than the size of the host space N_h. In *Dark-GLASSO*, the port spatiotemporal feature matrix V_p is not employed because it becomes computationally intractable when the size of the spatial features becomes large. *Dark-NMF* can handle not only the host spatiotemporal feature matrix V_h but also the port spatiotemporal feature matrix V_p due to its low computational complexity. Finally, because *Dark-NTD* is designed to handle a three-dimensional spatiotemporal feature tensor V_{hp} from the beginning of the proposal, it can calculate the host/port space simultaneously.

C. CONSIDERATIONS FOR ADVERSARIAL ATTACKS

This section discusses adversarial attacks that an attacker might implement to evade detection by *Dark-TRACER*. Possible attempts include multiplying noise, distributing spatial features, and reducing the frequency of temporal features, which might prevent the malware from capturing spatiotemporal features when the framework performs scanning compromises.

- ***The case wherein dummy scans, which are unrelated to an attack, are attached to a true attack to confuse the detection framework.***
  In this case, more data will be observed, and synchronization of the spatiotemporal features will be captured more strongly, resulting in better detection of true attacks. However, some of the detected events may contain dummy scan information, which may be troublesome for analysis.
- ***The case wherein multiple true attacks from many attack groups are distributed and executed simultaneously.***
  The advantage of *Dark-NMF* and *Dark-NTD* is that they can be decomposed into several patterns with similar spatiotemporal features. Therefore, *Dark-TRACER* can detect anomalies by dividing potential attack groups into several groups, even when orchestrated attacks are conducted simultaneously.
- ***The case of a slow stealth scan attack.***
  Depending on the degree of stealthiness, if a stealth scan attack is too slow, the synchronization of the observed spatiotemporal features becomes weak, making the attack difficult to detect. However, slow stealth scans are not efficient for an attacker who wants to spread the damage of their malware as quickly as possible, because the speed of spreading the malware infection is significantly slower. Slow stealth scans are generally considered to have purposes other than the spread of the

malware infection directly, and thus are not the target of detection in this study.

D. REDUCTION OF FALSE-POSITIVE ALERTS

As mentioned in Section III-C, the primary cause of false-positive alerts (#FPs) issued by *Dark-TRACER* is synchronized scanning by organizations for investigative purposes. Organizations such as Shodan and Censys, which deploy various cyberspace intelligence information as search engines, regularly scan the entire Internet space at a high frequency. Because such Internet-wide scans are fast and large-scale, they are observed in our darknet sensor networks and are represented as synchronized spatiotemporal patterns. Consequently, *Dark-TRACER* achieved a low number of FNs, whereas the number of FPs from investigative scanners is very high.

We believe that if *Dark-TRACER* can distinguish between alerts caused by investigative scanners and alerts caused by malware activities in a secondary manner, after detecting anomalous synchronous spatiotemporal features without missing them, the inefficient situation wherein there are many FPs can be significantly improved. In this study, to temporarily solve this challenge, we applied a simple rule that excluded alerts when a large number, or a sequential number, of TCP ports were seen simultaneously from the same source hosts in the alerts. In the first experiment described in Section III, we found that *Dark-NMF* halved the number of FPs while maintaining a high number of TPs, whereas *Dark-NTD* did not. In the second experiment in Section IV, by applying our simple rule, we were able to reduce the number of unique ports from 64,103 to 5,271 for the entire period alerted by *Dark-TRACER*. *In this way, we have demonstrated the feasibility of significantly improving malware detection by secondarily examining alerts. In future work, we would like to develop a model for classifying or clustering scanners for investigative purposes and automatically create a blacklist so that we can eliminate alerts caused by investigative scanners with better accuracy than the simple rule used in the present study.*

E. TOWARD THE PRACTICAL OPERATION OF DARK-TRACER

Each of the three independent proposed modules has its own strengths and weaknesses, and they complement each other through their collaboration into a single framework *Dark-TRACER*. From the two experiments presented in this paper, it was found that *Dark-TRACER* can achieve a 100% recall rate in the detection accuracy of malware activities and can also accomplish early detection. In this section, we discuss how *Dark-TRACER* can be operated in a practical manner.

First, we consider each module separately. *Dark-GLASSO* has a small number of FNs, but because there are few FPs, the precision rate #TPs / (#TPs+#FPs) is high. When it is not possible to spend much time on the analysis of the detection results, or when the analysis of global malware activities is sufficient, it is practical to employ only the detection results of *Dark-GLASSO*. Next, *Dark-NMF* and *Dark-NTD*, which use nonnegative tensor decomposition methods, are beneficial for detecting local malware activities because they can detect many FPs while incurring almost no FNs at an early stage. In particular, *Dark-NMF* is effective in detecting anomalous synchronization because it does not require preprocessing, has a very low computational cost, and shows good detection accuracy, even for darknet sensors with small observation scales. In contrast, *Dark-NTD* has a very high potential for detecting small-scale threats that are typically considered difficult to detect with other modules and is useful for capturing fairly localized events. *As described above, each module has different characteristics and can be utilized according to nature of the precise situation, or all modules can be fully leveraged into an integrated framework as in Dark-TRACER, taking advantage of their complementary relationship.*

Finally, we discuss a secondary analysis method for the detection results of *Dark-TRACER*. The alerts issued by *Dark-TRACER* contain information on IP addresses, targeted ports, and the timestamps of the hosts that are identified as abnormal. However, this information alone is often not enough to accurately determine malware activity. As mentioned in Section III-A, some malware activities, such as Mirai and Hajime, are known to have fingerprints in their initial scan packets. In large-scale scans such as malware and scanners which operate for investigative purposes, packet headers are often designed to have fingerprints in order to scan faster [44], [45]. Previous research has also reported that scanners use fingerprints to distinguish their scan results from backscatters [46].

The question arises as to what specific information should be checked. The following steps are considered useful for secondary analysis of *Dark-TRACER* alerts:

1) Computing the statistics of packet headers of detected alerts and find characteristic header information (including known fingerprints such as Mirai and Hajime).
2) Checking whether honeypots in an interactive observation network have observed any communication related to the detected alerts, and if so, analyzing what type of communication occurred interactively.
3) Collating and analyzing the presence of information related to the detected alert in third-party threat intelligence information (e.g., CVEs, vulnerabilities, and reports).

This is the actual workflow of the security operations center at NICTER. NICTER operations experts are expected to analyze the aforedescribed collation process in approximately 15 min per port of an alert. However, this does not necessarily imply that the causes and details of all events can be clarified. In order to increase the number of events that can be clarified as much as possible, it is necessary to collate more abundant information. In the future, we intend to extend *Dark-TRACER* by considering a wide range of applications, such as a mechanism to reduce false positives, improve both recall

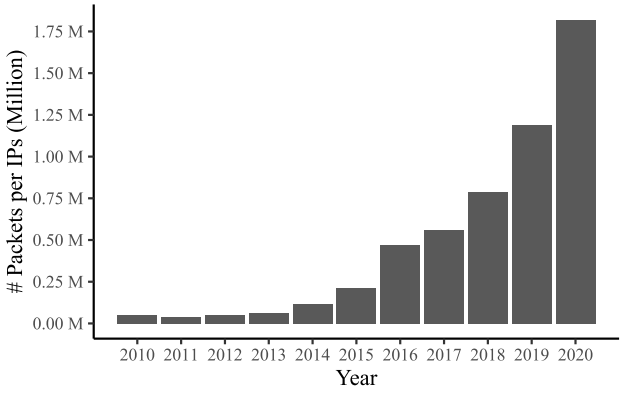

FIGURE 10. Total annual number of observed packets per IP address on NICTER.

and precision, and automatically associate threat intelligence from third parties [47].

VI. RELATED WORK

We describe related work on how darknets are leveraged in measurement analysis and malware activity detection. We also provide related studies and directions for identifying investigative scanners on darknet traffic, which will inevitably become necessary in the future.

A. DARKNET MEASUREMENT ANALYSIS

The darknet has attracted extensive attention in the field of network security, and many researchers are actively engaged in research on its development, analysis, and visualization [55]. Previous research [2], [56]–[58] has discussed the fundamentals of various darknet configurations, deployment techniques, and sensor placement techniques, and clarified the effectiveness of darknets. In addition, profiling, filtering, and classification have been intensively studied for the measurement of darknets. In the rest of this subsection, we present related work on IoT malware analysis and general darknet measurement analysis. A summary of the related studies is shown in Table 8.

Fig. 10 shows that the amount of observed traffic in NICTER's darknet observation network with a total of 300,000 IP addresses has increased rapidly over the past few years. The main reason for this is the IoT malware "Mirai", which appeared in 2016 [22]. In contrast to conventional botnets, IoT malware scans multiple ports in parallel to form a large-scale botnet that can spread the infection faster [48]. Moreover, IoT malware variants behave competitively with each other and are repeatedly destroyed and reinfected over a short period [23]. *The emergence of such diverse and sophisticated IoT malware further complicates cyber threats and makes it difficult to examine the actual current state of malware strategies. Therefore, it is essential to have a mechanism to investigate IoT botnets while they are still persistent and to rapidly and precisely detect potential threats.*

Apart from malware activity detection, which is discussed in the next subsection, the following studies were conducted in other areas of darknet measurement analysis. Dainotti *et al.* contributed to a census-like analysis of how the IP address space is used by developing malware and evaluating methods to remove spoofed traffic from darknets and live networks [49]. Durumeric *et al.* analyzed a large-scale darknet to investigate Internet-wide scanning activities and identify patterns of extensive horizontal scanning operations [50]. Fachkha *et al.* devised an inference and characterization module to identify and analyze the probing activities of cyber-physical systems (CPS) by extracting various features from large amounts of darknet data and performing correlational analyses [51]. Jonker *et al.* introduced a framework to protect against DoS attacks based on various data sources, including darknet traffic data [52]. They found that one-third of all /24 networks on the Internet had suffered at least one DoS attack in the past two years. Shaikh *et al.* identified unsolicited IoT devices by collecting IP header information from darknet traffic data and classifying them using several machine learning algorithms [53]. Akiyoshi *et al.* proposed a method to detect emerging scanning activities and their scale by analyzing the correlation between traffic in honeypots and darknets [54]. *Most of the measurement analysis studies using darknets have been applied to understand the general trend of malicious communications observed in darknets. Thus, for detailed analysis, many studies use not only darknet data but also trap-based monitoring systems such as honeypots.*

B. MALWARE ACTIVITY DETECTION ON DARKNETS

A summary of the related works referred to in this subsection is provided in Table 9. According to a survey paper on darknets [55], the technique of filtering misconfigured traffic has not yet been fully explored and is an ongoing challenge that deserves more attention from the research community. We consider that our method can filter out misconfigured traffic by detecting anomalies in the synchronization of spatiotemporal patterns. Furthermore, our method is unique in that it can detect global cyber threats/malware activities in real time in a uniform format by focusing on the synchronization of anomalous spatiotemporal patterns among many indiscriminate suspicious scans that reach large-scale darknets.

Here, we present some prior research that had a similar scope to our problem and used darknet traffic but did not focus on synchronization. There are several methods to detect anomalies by detecting change points in darknet traffic, such as *ChangeFinder* that was introduced as a comparison method in a previous study [10], [59]–[61]. Ahmed*et al.* proposed a sliding window-based adaptive cumulative sum (*CUSUM*) algorithm, which is a sequential analysis method for detecting drastic changes in darknet traffic [59]. Inoue *et al.* [60] employed the *ChangeFinder* algorithm [10] to detect sudden change points in darknet traffic with a low computational cost. Ban *et al.* proposed an abrupt-change detection algorithm that can detect botnet

TABLE 8. Summary of related works on darknet measurement analysis including IoT malware (Mirai).

Category	Year	Publication	Objective/Method/Contribution
IoT Malware Analysis (Mirai)	2017	[22]	Systematically and extensively investigated the Mirai botnet, revealing how it emerged, what devices were affected, and how Mirai variants continue to evolve.
	2020	[48]	Investigated the actual status of botnets caused by IoT malware and found that IoT botnets and their C&C servers were destroyed in the short term and actively infected and disposed of IoT devices.
	2020	[23]	Investigated the interaction and competitive behavior of the latest Mirai variants and found that IoT malware variants repeatedly destroy and reinfect each other in a short period.
Darknet Measurement Analysis	2013	[49]	Conducted a census-like analysis of the Internet space and proposed a method to remove spoofed traffic from darknets and live networks.
	2014	[50]	Analyzed darknet traffic data to investigate Internet-wide scanning activities of the entire Internet. Identified a large-scale horizontal pattern of scanning operations.
	2017	[51]	Extracted various features from darknet traffic data and analyzed probing activities to cyber-physical systems by performing correlation analysis.
	2017	[52]	Introduced a framework to protect against DoS attacks based on various data sources, including darknet and found that one-third of all /24 networks had suffered at least one DoS attack in the past two years.
	2018	[53]	Identified unsolicited IoT devices by collecting IP header information from darknet traffic data and classifying them using several machine learning algorithms.
	2018	[54]	Proposed a method to detect newly emerged scanning activities and their scale by analyzing the correlation between honeypots and darknet traffic.

TABLE 9. Summary of related works on darknet detecting malware activity from darknet traffic.

Category	Year	Publication	Approach/Method/Tools	Compared to *Dark-TRACER* ...
Anomalous Change Detection	2008	[59]	Proposed a sliding window-based adaptive cumulative sum (*CUSUM*) algorithm, which is a sequential analysis method for detecting drastic changes in darknet traffic.	The survey paper [55] pointed out that the filtering of misconfiguration communication is still an open problem. The change detection-based methods detect change points without distinguishing noise communication, such as misconfiguration communication, and thus cannot be expected to achieve high accuracy unless they are focused on specific protocols. However, *Dark-TRACER* focuses on the anomalous synchronization of spatiotemporal patterns to solve this problem. Moreover, it can analyze all darknet traffic data at once without focusing on a specific protocol.
	2008	[60]	Employed the *ChangeFinder* algorithm [10] to detect sudden change points in darknet traffic with a low computational cost.	
	2017	[61]	Proposed an abrupt-change detection algorithm that can detect botnet probe campaigns with a high detection rate by searching for temporal coincidences in botnet activities seen on the darknet.	
Clustering	2020	[62]	Extracted features embedded in unsolicited scan flows using convolutional neural networks and deployed hierarchical agglomerative clustering.	Although clustering methods can identify characteristic hosts or packets as a group, they cannot detect anomalies and thus cannot indicate which clusters are abnormal or characteristic. However, since *Dark-TRACER* raises alerts in real time based on anomaly detection, it is easy to grasp what to analyze first, making it highly practical. In addition, clustering methods are not suitable for the early detection of unknown or small-scale malware activities because clusters are not formed unless the amount of observed features of malware activities is large.
	2020	[63]	Inferred the compromised IoT devices, characterized the generated scanning campaigns, and clustered them.	
	2020	[24]	Proposed a method for sequential clustering of attacking hosts having similar intentions by embedding port sequences for each source host.	
Anomalous Synchronization Detection	2019	[6]	Proposed a method for real-time anomaly detection of spatiotemporal pattern synchronization using *graphical lasso* and graph density.	Since the previous methods have been considered independently, their relationship was not clear. In this paper, we integrated and evaluated the three previous methods as *Dark-TRACER* and clarified that they complement each other. In addition, although the previous methods can detect malware in real time, the feasibility of early detection of malware activity was not evaluated; therefore, in this paper, we evaluated the early detection and the analysis cost.
	2020	[7]	Quantitatively evaluated the accuracy of the method in paper [6] and discussed the results and limitations in depth.	
	2021	[8]	Proposed a method for real-time anomaly detection of spatiotemporal pattern synchronization using *NMF* and evaluated detection accuracy.	
	2019	[9]	Proposed a method for real-time anomaly detection of spatiotemporal pattern synchronization using *NTD* but did not evaluate it quantitatively.	

probe campaigns with a high detection rate by searching for temporal coincidences in botnet activities observed on the darknet [61]. *The aforementioned change detection methods all share the same drawback—they cannot achieve high accuracy without focusing on specific protocol ports because they detect change points without distinguishing between many sources of noisy communications, such as misconfigured traffic.* As shown in the experimental results of *ChangeFinder* herein, the accuracy was low when the algorithm was applied to the entire traffic without focusing on a specific port. In addition, applying the change detection method to a specific port would result in many alerts, which would require considerable computational and analysis costs.

Next, we review recent related works on the analysis of malware activity using darknet data (mainly clustering). As mentioned in the previous subsection on darknet mea-

surement analysis, most of the communications reaching the darknet since 2016 have seen a considerable increase in traffic volume due to IoT malware. Therefore, many clustering methods targeting the analysis of IoT malware activities have been studied in recent years. Pour *et al.* learned to classify compromised IoT devices and non-IoT devices using convolutional neural networks [62]. Using the classification training results, they extracted features embedded in unsolicited scan flows and deployed hierarchical agglomerative clustering. As a result, the authors uncovered 440,000 compromised IoT devices and 350 IoT botnets. Torabi *et al.* leveraged the Shodan IoT search engine and darknet traffic data to infer compromised IoT devices and to characterize and cluster the generated scanning campaigns [63]. The authors discovered newly targeted ports and emerging IoT malware/botnets, highlighting their persistence and evolutionary process. Cohen *et al.* proposed a method for sequential clustering of aggressive hosts having similar intentions from scalable darknet traffic by embedding a port sequence for each source host [24]. By tracking the clusters, they detected recurrent or new attacks and found several new campaigns of malicious port sequences.

Such clustering methods are considered to be able to classify malware activities, investigative scans, and noisy communications (e.g., misconfigured traffic) to some extent. However, although the clustering methods can distinguish distinctive hosts or packets as a group, they cannot detect anomalies and thus cannot identify which clusters are anomalous or distinctive. Therefore, it is necessary to prioritize the clusters to be analyzed based on some criteria. In contrast, as *Dark-TRACER* issues alerts in real time upon anomaly detection, it is easy to know what to analyze first, thereby making the framework highly practical. In addition, clustering methods are not suitable for the early detection of unknown or small-scale malware activities because clusters are not formed unless the amount of observed features of malware activities is large. As shown in the experimental results herein, *Dark-TRACER* has good potential to detect small-scale malware activities in an early stage.

BotSniffer [1] and *BotMiner* [64] proposed a framework for detecting `C&C` traffic and malicious activities based on the spatiotemporal correlation method. However, the scope of their problem setting was different from that of ours, because *BotSniffer* and *BotMiner* only target specific protocols of actual network traffic and therefore cannot detect global cyber threats and malware activities in the entire Internet space. To the best of our knowledge, there is no related work that focuses on synchronization in the same scope as the present study. As described in Table 9, prior papers [6] and [7] were published as prototypes of *Dark-GLASSO*, prior paper [8] was published as a prototype of *Dark-NMF*, and, finally, prior paper [9] was published as a prototype of *Dark-NTD*. However, as the previous methods have been considered independently, their relationship has remained unclear. In this study, we integrated and evaluated the three previous methods as *Dark-TRACER* and clarified that they complement each other. In addition, although the previous methods can detect malware in real time, the feasibility of early detection of malware activity was not evaluated; therefore, in this study, we evaluated the early detection performance and the analysis cost.

We also present several other related works that utilized darknet traffic to identify and detect malicious communications, based on the reports of a recent survey paper [65]. Kumar *et al.* proposed a model that learns from darknet data and benign traffic data to test whether it can classify malicious communications [66]. However, the model only classifies whether the traffic flow is malicious or benign, and because it learns all at once, it does not specifically identify what kind of maliciousness it has classified. Bou-Harb *et al.* investigated orchestrated probing campaigns by considering a clustering method for time-series traffic data [67]. However, this method does not detect anomalies and does not distinguish noisy communications. Ali *et al.* classified `DDoS` attacks using Resource Allocating Network with Locality Sensitive Hashing) (*RAN-LSH*), which employs *LSH* to select data for training and achieves fast online learning by training only selected data [68]. However, because this method only analyzes backscatter traffic and targets to classify `DDoS` attacks, it is not suitable for detecting and classifying malware activities, which do not involve backscatter.

C. TOWARDS THE IDENTIFICATION OF INVESTIGATIVE SCANNERS

We conclude this section by sharing related works that have distinguished between investigative scanners, which is an issue that must be considered in future studies. A summary of the related works mentioned in this subsection is presented in Table 10. Many massive Internet-wide scanners are observed on the darknet, including both public scanning activities and malware activities. Recently, high-performance scanning tools such as `ZMap` [44] and `Masscan` [69] have been deployed, and Mazel *et al.* profiled the utilization of such tools [45]. The results revealed that many entities openly engage in scanning activities on a large scale and on a constant basis. Because such harmless and large-scale investigative scanners perform activities with relatively synchronized spatiotemporal patterns, many of these scanners were incorrectly detected in the results of this study. Therefore, we must consider how to distinguish such harmless investigative scanners from malware activities.

As mentioned in the previous subsection, *DANTE* [24] embeds port sequences of darknet traffic by source host in a given time frame and performs clustering. By comparing with the previous time frame and labeling the clusters, it is possible to track campaigns and detect recurrent or new attacks. In large-scale scans using scanning tools, such as `ZMap` or `Masscan`, or large-scale scans using malware, fingerprints are often attached to packet headers to perform faster scans [46]. It has also been reported in [46] that fingerprints are provided to distinguish scan results from backscatters. In contrast, Tanaka *et al.* proposed

TABLE 10. Summary of related works on identifying investigative scanners.

Year	Publication	Objective/Method/Contribution
2017	[45]	Profiled the utilization of scanning tools such as `ZMap` [44] and `Masscan` [69]. Many entities were openly engaged in scanning activities on a large scale and constantly.
2020	[46]	In large-scale scans using scan tools or malware, fingerprints are often assigned to packet headers for faster scanning. Therefore, a technique to identify fingerprints from darknet traffic in a rule-based manner was proposed.
2021	[70]	A method to automatically identify fingerprints embedded in TCP/IP headers from darknet traffic was proposed based on a genetic algorithm. It successfully identified unknown fingerprints from a short period of data.
2020	[71]	It was found that the coverage rate differs depending on the scan source, and that the observed hosts differ depending on the region of the observation network.

a method based on a genetic algorithm to automatically identify fingerprints embedded in TCP/IP headers from darknet traffic [70]. They succeeded in identifying unknown fingerprints from data corresponding to a short period. *Identifying the fingerprints of investigative scanners and tracing the scanners using the DANTE mechanism may enable us to distinguish scanners from malware activities.* Additionally, Wan *et al.* clarified that the coverage rate differs depending on the scan source and that the observed hosts differ depending on the region of the observation network [71]. Analyzing the darknet observation networks in various regions, such as the autonomous system (AS) and in various countries, is a way to obtain a more detailed and precise understanding of the actual scanning situation.

VII. CONCLUSION

In this study, we introduced three independent machine learning methods to automatically estimate the synchronization of the spatiotemporal patterns of darknet traffic in real time and to detect anomalies. Those three methods are: *Dark-GLASSO*, *Dark-NMF*, and *Dark-NTD*. We also proposed *Dark-TRACER*, which integrates all three methods into a single framework. We found that *Dark-TRACER* was able to complement the weaknesses of each module, achieving a 100% recall rate and detecting all malware activities in the experiment. It detected the malware on average 153.6 days earlier than the time when the threats were revealed to the public by reputable third-party security research organizations. In addition, we found that two analysts could perform the daily operations necessary to detect these threats in approximately 7.3 h.

Currently, our most serious challenge is the large number of false positives. In this study, we confirmed that even a simple rule-based approach can effectively reduce the number of false-positive alerts. As described in Sections V-D and VI-C, our future work is to reduce the number of false positives by identifying the fingerprints of investigative scanners and building a model to track them. By reducing the number of false positives, the analysis cost can be lowered. In addition, we intend to automate the secondary collision analysis mentioned in Section V-E to elucidate the causes and details of the alerts detected by *Dark-TRACER*. Finally, we plan to deploy *Dark-TRACER* in the real world and detect threats and malware activities in real-time to aid rapid response.

ACKNOWLEDGMENT

The authors would like to thank an Associate Professor Katsunari Yoshioka from Yokohama National University and Prof. Noboru Murata from Waseda University for their valuable comments.

REFERENCES

[1] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting botnet command and control channels in network traffic," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2008, pp. 1–19.
[2] M. Bailey, E. Cooke, F. Jahanian, A. Myrick, and S. Sinha, "Practical darknet measurement," in *Proc. 40th Annu. Conf. Inf. Sci. Syst.*, Mar. 2006, pp. 1496–1501.
[3] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, Dec. 2007.
[4] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2000, pp. 535–541.
[5] Y.-D. Kim and S. Choi, "Nonnegative tucker decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[6] C. Han, J. Shimamura, T. Takahashi, D. Inoue, M. Kawakita, J. Takeuchi, and K. Nakao, "Real-time detection of malware activities by analyzing darknet traffic using graphical lasso," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Aug. 2019, pp. 144–151.
[7] C. Han, J. Shimamura, T. Takahashi, D. Inoue, J. Takeuchi, and K. Nakao, "Real-time detection of global cyberthreat based on darknet by estimating anomalous synchronization using graphical lasso," *IEICE Trans. Inf. Syst.*, vol. 103, no. 10, pp. 2113–2124, Oct. 2020.
[8] C. Han, J. Takeuchi, T. Takahashi, and D. Inoue, "Automated detection of malware activities using nonnegative matrix factorization," in *Proc. IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021.
[9] H. Kanehara, Y. Murakami, J. Shimamura, T. Takahashi, D. Inoue, and N. Murata, "Real-time botnet detection using nonnegative tucker decomposition," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 1337–1344.
[10] J. Takeuchi and K. Yamanishi, "A unifying framework for detecting outliers and change points from time series," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 4, pp. 482–492, Apr. 2006.
[11] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by internet-wide scanning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 542–553.
[12] T. Ide, A. Khandelwal, and J. Kalagnanam, "Sparse Gaussian Markov random field mixtures for anomaly detection," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 955–960.
[13] A. J. Gibberd and J. D. B. Nelson, "High dimensional changepoint detection with a dynamic graphical lasso," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 2684–2688.
[14] T. Idé, A. C. Lozano, N. Abe, and Y. Liu, "Proximity-based anomaly detection using sparse structure learning," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2009, pp. 97–108.
[15] S. Liu, T. Suzuki, and M. Sugiyama, "Support consistency of direct sparse-change learning in Markov networks," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2785–2791.
[16] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[17] Q. Zhao, C. F. Caiafa, D. P. Mandic, L. Zhang, T. Ball, A. Schulze-Bonhage, and A. and Cichocki, "Multilinear subspace regression: An orthogonal tensor decomposition approach," in *Proc. 25th Annu. Conf. Neural Inf. Process. Syst.*, 2011, pp. 1269–1277.

[18] A. H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *IEICE Nonlinear Theory Appl.*, vol. 1, no. 1, pp. 37–68, 2010.

[19] A. Anandkumar, P. Jain, Y. Shi, and U. N. Niranjan, "Tensor vs. matrix methods: Robust tensor decomposition under block sparse perturbations," in *Proc. 19th Int. Conf. Artif. Intell. Statist., (AISTATS)*, vol. 51, 2016, pp. 268–276.

[20] C. F. Caiafa and A. Cichocki, "Generalizing the column–row matrix decomposition to multi-way arrays," *Linear Algebra its Appl.*, vol. 433, no. 3, pp. 557–573, Sep. 2010.

[21] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, "Efficient nonnegative tucker decompositions: Algorithms and uniqueness," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4990–5003, Dec. 2015.

[22] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. and Zhou, "Understanding the Mirai botnet," in *Proc. 26th USENIX Secur. Symp.*, 2017, pp. 1093–1110.

[23] H. Griffioen and C. Doerr, "Examining Mirai's battle over the Internet of Things," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 743–756.

[24] D. Cohen, Y. Mirsky, M. Kamp, T. Martin, Y. Elovici, R. Puzis, and A. Shabtai, "DANTE: A framework for mining and monitoring darknet traffic," in *Proc. 25th Eur. Symp. Res. Comput. Secur. (ESORICS)*. Springer, 2020, pp. 88–109.

[25] FOX-IT. (2016). *Recent Vulnerability in Eir D1000 Router Used to Spread Updated Version of Mirai DDoS Bot*. [Online]. Available: https://blog.fox-it.com/2016/11/28/recent-vulnerability-in-eir-d1000-ro% uter-used-to-spread-updated-version-of-mirai-ddos-bot/

[26] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer IoT botnet," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.

[27] Netlab 360. (2018). *HNS Botnet Recent Activities*. [Online]. Available: https://blog.netlab.360.com/hns-botnet-recent-activities-en/

[28] Netlab 360. (2018). *7,500+ MikroTik Routers are Forwarding Owners' Traffic to the Attackers, How is Yours?*. [Online]. Available: https://blog.netlab.360.com/7500-mikrotik-routers-are-forwarding-owners% -traffic-to-the-attackers-how-is-yours-en/

[29] Huawei. (2017). *Security Notice—Statement on Remote Code Execution Vulnerability in Huawei HG532 Product*. [Online]. Available: https://www.huawei.com/en/psirt/security-notices/huawei-sn-20171130-01-% hg532-en

[30] Tenable Research Blog. (2018). *Tenable Research Advisory: Peekaboo Critical Vulnerability in NUUO Network Video Recorder*. [Online]. Available: https://www.tenable.com/blog/tenable-research-advisory-peekaboo-critica% l-vulnerability-in-nuuo-network-video-recorder

[31] Netlab 360. (2018). *BCMPUPnP_Hunter: A 100k Botnet Turns Home Routers to Email Spammers*. [Online]. Available: https://blog.netlab.360.com/bcmpupnp_hunter-a-100k-botnet-turns-home-ro% uters-to-email-spammers-en/

[32] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[33] I. Ilascu. (2019). *New Echobot Botnet Variant Uses Over 50 Exploits to Propagate*. [Online]. Available: https://www.bleepingcomputer.com/news/security/new-echobot-botnet-varia% nt-uses-over-50-exploits-to-propagate/

[34] Netlab 360. (2019). *The Botnet Cluster on the 185.244.25.0/24*. [Online]. Available: https://blog.netlab.360.com/the-botnet-cluster-on-185-244-25-0-24-en/

[35] Microsoft. (2019). *Remote Desktop Services Remote Code Execution Vulnerability*. [Online]. Available: https://msrc.microsoft.com/update-guide/en-U.S./vulnerability/CVE-2019-07% 08

[36] Bashis. (2018). *Shenzhen TVT Digital Technology co. ltd & oem DVR/NVR/IPC API RCE*. [Online]. Available: https://vulners.com/seebug/SSV:97217

[37] V. Yarmak. (2020). *Full Disclosure: 0day Vulnerability (Backdoor) in Firmware for Xiaongmai-Based DVRs, NVRs and IP Cameras*. [Online]. Available: https://habr.com/en/post/486856/

[38] K. Hsu, H. Zhang, Z. Zhang, and R. Nigam. (2020). *Grandstream and Draytek Devices Exploited to Power New Hoaxcalls DDoS Botnet*. [Online]. Available: https://unit42.paloaltonetworks.com/new-hoaxcalls-ddos-botnet/

[39] Pierre. (2020). *Multiple Vulnerabilities found in Zyxel CNM Secumanager*. [Online]. Available: https://pierrekim.github.io/blog/2020-03-09-zyxel-secumanager-0day-vuln% erabilities.html

[40] F-Secure. (2020). *Saltstack Authorization Bypass*. [Online]. Available: https://labs.f-secure.com/advisories/saltstack-authorization-bypass/

[41] REW. *Linksys e-Series—Remote Code Execution*. [Online]. Available: https://www.exploit-db.com/exploits/31683, 2014.

[42] J. Lands. (2020). *Priority Threat Actors Adopt Mirai Source Code*. [Online]. Available: https://blogs.juniper.net/en-us/security/priority-threat-actors-adopt-mirai-source-code

[43] M. Althibyani. (2020). *Oracle Weblogic Server 10.3.6.0.0/12.1.3.0.0/12.2.1.3.0/12.2.1.4.0/14.1.1.0.0—Unauthenticated rce Via Get Request*. [Online]. Available: https://www.exploit-db.com/exploits/48971

[44] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *Proc. 22th USENIX Secur. Symp.*, 2013, pp. 605–620.

[45] J. Mazel, R. Fontugne, and K. Fukuda, "Profiling internet scanners: Spatiotemporal structures and measurement ethics," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2017, pp. 1–9.

[46] H. Griffioen and C. Doerr, "Discovering collaboration: Unveiling slow, distributed scanners based on common header field patterns," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–9.

[47] T. Takahashi, Y. Umemura, C. Han, T. Ban, K. Furumoto, O. Nakamura, K. Yoshioka, J. Takeuchi, N. Murata, and Y. Shiraishi, "Designing comprehensive cyber threat analysis platform: Can we orchestrate analysis engines?" in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops Affiliated Events (PerCom Workshops)*, Mar. 2021, pp. 376–379.

[48] R. Tanabe, T. Tamai, A. Fujita, R. Isawa, K. Yoshioka, T. Matsumoto, C. Gañán, and M. van Eeten, "Disposable botnets: Examining the anatomy of IoT botnet infrastructure," in *Proc. 15th Int. Conf. Availability, Rel. Secur.*, Aug. 2020, pp. 7:1–7:10.

[49] A. Dainotti, K. Benson, A. King, K. Claffy, M. Kallitsis, E. Glatz, and X. Dimitropoulos, "Estimating internet address space usage through passive measurements," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 1, pp. 42–49, Dec. 2013.

[50] Z. Durumeric, M. Bailey, and J. A. Halderman, "An internet-wide view of internet-wide scanning," in *Proc. 23rd USENIX Secur. Symp.*, 2014, pp. 65–78.

[51] C. Fachkha, E. Bou-Harb, A. Keliris, N. Memon, and M. Ahamad, "Internet-scale probing of CPS: Inference, characterization and orchestration analysis," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2017, pp. 1–15.

[52] M. Jonker, A. King, J. Krupp, C. Rossow, A. Sperotto, and A. Dainotti, "Millions of targets under attack," in *Proc. Internet Meas. Conf.*, Nov. 2017, pp. 100–113, doi: 10.1145/3131365.3131383.

[53] F. Shaikh, E. Bou-Harb, J. Crichigno, and N. Ghani, "A machine learning model for classifying unsolicited IoT devices by observing network telescopes," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 938–943.

[54] R. Akiyoshi, D. Kotani, and Y. Okabe, "Detecting emerging large-scale vulnerability scanning activities by correlating low-interaction honeypots with darknet," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2018, pp. 658–663.

[55] C. Fachkha and M. Debbabi, "Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1197–1227, 2nd Quart., 2016.

[56] M. Bailey, E. Cooke, F. Jahanian, and J. Nazario, "The internet motion sensor—A distributed blackhole monitoring system," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2005, pp. 167–179.

[57] V. Yegneswaran, P. Barford, and D. Plonka, "On the design and use of internet sinks for network abuse monitoring," in *Recent Adv. Intrusion Detection, 7th Int. Symp. (RAID)*, vol. 3224, 2004, pp. 146–165.

[58] D. Moore, "Network telescopes: Tracking denial-of-service attacks and internet worms around the globe," in *Proc. 17th Conf. Syst. Admin. (LISA)*, 2003. [Online]. Available: https://www.usenix.org/conference/lisa-03/network-telescopes-tracking-denial-service-attacks-and-internet-worms-around

[59] E. Ahmed, A. Clark, and G. Mohay, "A novel sliding window based change detection algorithm for asymmetric traffic," in *Proc. IFIP Int. Conf. Netw. Parallel Comput.*, Oct. 2008, pp. 168–175.

[60] D. Inoue, K. Yoshioka, M. Eto, M. Yamagata, E. Nishino, J. Takeuchi, K. Ohkouchi, and K. Nakao, "An incident analysis system NICTER and its analysis engines based on data mining techniques," in *Proc. Neural Inf. Process. 15th Int. Conf. (ICONIP)*, vol. 5506, 2008, pp. 579–586.
[61] T. Ban, L. Zhu, J. Shimamura, S. Pang, D. Inoue, and K. Nakao, "Detection of botnet activities through the lens of a large-scale darknet," in *Neural Information Processing*. Cham, Switzerland: Springer, 2017, pp. 442–451, doi: 10.1007/978-3-319-70139-4_45.
[62] M. S. Pour, A. Mangino, K. Friday, M. Rathbun, E. Bou-Harb, F. Iqbal, S. Samtani, J. Crichigno, and N. Ghani, "On data-driven curation, learning, and analysis for inferring evolving Internet-of-Things (IoT) botnets in the wild," *Comput. Secur.*, vol. 91, Apr. 2020, Art. no. 101707.
[63] S. Torabi, E. Bou-Harb, C. Assi, E. B. Karbab, A. Boukhtouta, and M. Debbabi, "Inferring and investigating IoT-generated scanning campaigns targeting a large network telescope," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 402–418, Jan. 2020.
[64] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. 17th USENIX Secur. Symp.*, 2008, pp. 139–154.
[65] P. S. Joshi and H. Dinesha, "Survey on identification of malicious activities by monitoring darknet access," in *Proc. 3rd Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Aug. 2020, pp. 346–350.
[66] S. Kumar, H. Vranken, J. van Dijk, and T. Hamalainen, "Deep in the dark: A novel threat detection system using darknet traffic," in *Proc. IEEE Int. Conf. on Big Data (Big Data)*, Dec. 2019, pp. 4273–4279.
[67] E. Bou-Harb, M. Debbabi, and C. Assi, "A time series approach for inferring orchestrated probing campaigns by analyzing darknet traffic," in *Proc. 10th Int. Conf. Availability, Rel. Secur.*, Aug. 2015, pp. 180–185.
[68] S. H. A. Ali, S. Ozawa, T. Ban, J. Nakazato, and J. Shimamura, "A neural network model for detecting DDoS attacks using darknet traffic features," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 2979–2985.
[69] R. Graham. (2013). *Masscan: The Entire Internet in 3 Minutes, Errata Security Blog*. [Online]. Available: https://blog.erratasec.com/2013/09/masscan-entire-internet-in-3-minutes%.html
[70] A. Tanaka, C. Han, T. Takahashi, and K. Fujisawa, "Internet-wide scanner fingerprint identifier based on TCP/IP header," in *Proc. 6th Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Dec. 2021.
[71] G. Wan, L. Izhikevich, D. Adrian, K. Yoshioka, R. Holz, C. Rossow, and Z. Durumeric, "On the origin of scanning: The impact of location on internet-wide scans," in *Proc. ACM Internet Meas. Conf.*, Oct. 2020, pp. 662–679.

CHANSU HAN (Member, IEEE) received the B.E. degree in computer science and the M.S. and Ph.D. degrees in informatics engineering from Kyushu University, in 2016, 2018, and 2021, respectively. He is currently a Researcher with the National Institute of Information and Communications Technology (NICT), Japan. His research interests include analyzing and solving problems in the cybersecurity field (especially networks and malware) using machine learning.

JUN'ICHI TAKEUCHI (Member, IEEE) received the B.Sc. degree in physics and the Dr.Eng. degree in mathematical engineering from the University of Tokyo, in 1989 and 1996, respectively. From 1989 to 2006, he worked with NEC Corporation, Japan. In 2006, he joined Kyushu University, Fukuoka, Japan, where he is currently a Professor in mathematical engineering. From 1996 to 1997, he was a Visiting Research Scholar with the Department of Statistics, Yale University, New Haven, CT, USA. His research interests include mathematical statistics, information geometry, information theory, data science, and machine learning. He is a member of IEICE and JSIAM.

TAKESHI TAKAHASHI (Member, IEEE) received the Ph.D. degree in telecommunications from Waseda University, in 2005. He was with the Tampere University of Technology as a Researcher, from 2002 to 2004, and Roland Berger Ltd., as a Business Consultant, from 2005 to 2009. Since 2009, he has been with the National Institute of Information and Communications Technology, where he is currently a Research Manager. His research interests include cybersecurity and machine learning.

DAISUKE INOUE (Member, IEEE) received the B.E. and M.E. degrees in electrical and computer engineering and the Ph.D. degree in engineering from Yokohama National University, in 1998, 2000, and 2003, respectively. He is currently the Director of the Cybersecurity Laboratory, National Institute of Information and Communications Technology (NICT). He has received several awards, including the Asia-Pacific Information Security Leadership Achievements (ISLA), in 2014.