Detecting and Interpreting Changes in Scanning Behavior in Large Network Telescopes

Michalis Kallitsis *Member, IEEE*, Rupesh Prajapati *Member, IEEE*,
Vasant Honavar *Senior Member, IEEE*, Dinghao Wu *Member, IEEE, John Yen Fellow, IEEE*

***Abstract*—Network telescopes or "Darknets" received unsolicited Internet-wide traffic, thus providing a unique window into macroscopic Internet activities associated with malware propagation, denial of service attacks, network reconnaissance, misconfigurations and network outages. Analysis of the resulting data can provide actionable insights to security analysts that can be used to prevent or mitigate cyber-threats. Large network telescopes, however, observe millions of nefarious scanning activities on a daily basis which makes the transformation of the captured information into meaningful threat intelligence challenging. To address this challenge, we present a novel framework for characterizing the structure and temporal evolution of scanning behaviors observed in network telescopes. The proposed framework includes four components. It (i) extracts a rich, high-dimensional representation of *scanning profiles* composed of features distilled from network telescope data; (ii) learns, in an unsupervised fashion, information-preserving *succinct representations* of these scanning behaviors using *deep representation learning* that is amenable to clustering; (iii) performs *clustering* of the scanner profiles in the resulting latent representation space on daily Darknet data, and (iv) *detects temporal changes* in scanning behavior using techniques from *optimal mass transport*. We robustly evaluate the proposed system using both synthetic data and real-world Darknet data. We demonstrate its ability to detect real-world, high-impact cybersecurity incidents such as the onset of the Mirai botnet in late 2016 and several interesting cluster formations in early 2022 (e.g., heavy scanners, evolved Mirai variants, Darknet "backscatter" activities, etc.). Comparisons with state-of-the-art methods showcase that the integration of the proposed features with the deep representation learning scheme leads to better classification performance of Darknet scanners.**

***Index Terms*—Network telescope, Internet-wide measurements, anomaly detection, deep learning, autoencoders, clustering.**

I. Introduction

Cyber-attacks present one of the most severe threats to the safety of citizenry and the security of the nation's critical infrastructure (e.g., the energy grid, transportation networks, health systems, food and water supply networks). A critical phase in a cyber-attack is "network reconnaissance", which often involves "scanning" for potentially vulnerable machines or devices on the Internet so that these vulnerabilities may be exploited during later phases of the cyber-attack. Similarly, malware that attempt to propagate from one compromised machine to other unsecured devices are also engaged in malicious scanning activities. Such actions are difficult to be identified in an operational network because they are oftentimes low-volume and interwoven with normal network traffic.

Early detection of these scanning behaviors can provide germane information to network security analysts to detect ongoing cyber threats and to more effectively mitigate them. Network telescopes [1], [2], also known as "Darknets", provide a unique opportunity for characterizing and detecting Internet-wide malicious activities. A network telescope receives and records unsolicited traffic—known as Internet Background Radiation (IBR)—destined to an *unused* but *routed* address space. This "dark IP space" hosts no services or devices, and therefore any traffic arriving to it is inherently malicious. No regular user traffic reaches the Darknet. Thus, network telescopes have been frequently used by the networking and security communities to shed light into dubious malware propagation and interminable network scanning activities [3]–[7]. For instance, a large network telescope operated by Merit Network [2] had been employed in late 2016 to understand the outset and spread of the Mirai botnet [3]. Mirai is among the first botnets that compromised millions of Internet-of-Things (IoT) devices and was responsible for some of the largest Distributed Denial of Service (DDoS) attacks (including attacks against critical DNS infrastructure) ever recorded [3], [8]. However, this analysis was conducted *post-mortem*, i.e., after the catastrophic consequences had been realized. A systematic way to detect changes of ongoing scanning behaviors observed in the Darknet is yet to be developed.

Automated detection of changes in scanning behavior in Darknets is a challenging task due to the vast amounts of senders targeting any combination of UDP/TCP ports and/or other protocols. In this paper, we aim to address this challenge by proposing a framework that continuously monitors and detects changes of scanning activities observed in the Darknet.

An important task in this context has to do with *clustering the different Darknet scanners*, based on their traffic profile, their port scanning patterns, etc., and then employing these "groupings" as signatures to *detect temporal changes in the evolution of the Darknet* (i.e., detect changes between groupings of two time points, e.g., two adjacent days). This problem can be reformulated as a problem of change detection through unsupervised clustering. However, solving this problem presents several non-trivial challenges: (i) The number of ports being scanned in a day can be in the order of tens of thousands, resulting in an extremely high-dimensional feature space. Distance calculations are known to be inherently unreliable in high-dimensional settings [9], making it challenging to apply standard clustering methods that rely on

Michalis Kallitsis is with Merit Network, Inc. in Ann Arbor, Michigan, USA. His e-mail address is mgkallit@merit.edu.

Rupesh Prajapati, Vasant Honavar, Dinghao Wu and John Yen are with the Pennsylvania State University, University Park, Pennsylvania, U.S.A.; their e-mail addresses are [rxp338, vuh14, duw12, juy1]@psu.edu.

measuring distance between data samples to cluster them; (ii) Linear dimensionality reduction techniques such as Principal Component Analysis (PCA) [10] fail to cope with nonlinear interactions between the observed variables; (iii) Detecting shifts of the Darknet structure between two time points is a non-trivial task because it involves hundreds of clusters with high-dimensional features that may differ in their size and/or their cluster structure.

Against this background, this paper explores a novel unsupervised approach for detecting changes in scanning behavior that overcomes the aforementioned challenges through the use of (i) information-preserving, low-dimensional *embeddings* of daily scanning profiles acquired via *deep autoencoders*, and (ii) ideas from *optimal mass transport* [11] to measure changes in scanning behavior between two time points. The proposed framework is applied on Merit's large network telescope [2], and we demonstrate its potential to extract high-impact Darknet events in an automated manner.

The *key contributions* of the paper are as follows: We leverage the recent advances in deep neural networks and employ powerful *embedding* or *representation learning* methods (see Sec. V-A) to automate the construction of an information-preserving *low-dimensional* latent space to represent the heterogeneous, complex and high-dimensional features of scanning profiles (described in Sec. IV). We then apply standard *clustering* methods, e.g., K-means, to the resulting latent representation the scanning profiles (Sec. V-B). To enable the detection of changes in the scanning behavior, we propose the use of a *Wasserstein metric* from optimal mass transport theory [11], [12] to assess the dissimilarity or "distance" between Darknet clusters from two adjacent monitoring windows (e.g., two adjacent days for daily comparison, see Sec. V-C). Further, the outcome of the *optimal transport plan* employed to calculate the Wasserstein distance is utilized to interpret the "change" identified. Figure 1 illustrates an application of the proposed approach on Merit's Darknet during the emergence of the Mirai botnet in September 2016 [3]. Our methodology is able to uncover two important changes in the Darknet's observed behavior: one associated with changes in Mirai's scanning patterns on September 14th, 2016; and another indicative of heavy DNS scans that first occurred on September 24th, 2016. We elaborate further on these results in Sec. VI which includes the evaluation of our methodology using real as well as synthetically-generated data, and comparisons with related work.

II. Related Work

Internet measurement studies. Darknets provide a unique perspective into Internet-wide scanning activities and several studies focused on network telescope data to understand malware propagation, network reconnaissance, botnets and misconfigurations [4], [6], [7], [13]–[16]. Darknet data have been also utilized to study DDoS attacks [17]–[19] and Darknet *backscatter* [20], [21], IPv6 routing instabilities [22], long-term cyber attacks [23], [24] and network outages [25]–[27].

Of particular interest in this context is the use of Darknet data for detecting and characterizing new malware. The Mirai botnet, for instance, is known to have started its malware propagation activity by first scanning port TCP/23 (Telnet) for potential victims in the Internet [3]. Over time, and as Internet-of-Things (IoT) devices had proven to be very susceptible in getting compromised by malware infection, its scanning behavior changed as well. It proceeded to scan port TCP/2323, and eventually 10 other ports [3]. Other studies have employed Darknet data to obtain insights on the IoT ecosystem and its vulnerabilities [28]–[30]. Hence, reliably detecting and responding to such attacks calls for effective methods for rapid identification of novel signatures of malware behavior.

Clustering Darknet data. Clustering offers a powerful approach to the analyses of Darknet data to identify novel attack patterns, victims of attacks, novel network scanners, etc. [31]–[33]. For example, Nishikaze et al. [31] encode Darknet traffic using 27 network features associated with blocks / subnets of the IP space and cluster the resulting data to group the subnets according to their traffic profiles. Ban et al. [32] have shown how clustering of Darknet data, followed by frequent pattern mining and visualization can be used to detect novel attack patterns. Iglesias and Szeby [33] have shown how to cluster IBR data from Darknet based on a novel representation of network traffic to identify network traffic patterns that are characteristic of activities such as long term scanning, as well as bursty events from targeted attacks and short term incidents. Sarabi and Liu [34] employ deep learning for obtaining lightweight embeddings to characterize the population of Internet hosts as observed by scanning services such as `Censys.io` [35].

Methods based on embedding features into a lower-dimensional space using ideas from natural language processing and "word embeddings" appear in [36]–[38]. Closest to ours are the DarkVec [36] and DANTE [37] works in which scanning IPs and destination ports, respectively, are employed to construct the embeddings and then perform clustering. DANTE also utilizes the clustering outcomes for novelty detection based on using Jaccard similarity scores between clusters of adjacent time windows. The work in [36] found the training times of the neural networks employed in DANTE [37] and IP2VEC [38] to be exceedingly high (see Table 3, [36]), even for a small Darknet like the one employed in their study (i.e., a Darknet observing only 543,900 unique IPs in a whole month; contrast this with the 35 million unique source IPs present in our monthly dataset, see Table I). Thus, these methods are impractical to be used in large network telescopes. DarkVec exhibits good training performance, however, as its authors acknowledge, it performs best when one utilizes *"proper service definitions"* (i.e., defining the group of ports that a service commonly uses, such as TCP/80 and TCP/8080 for HTTP). This information is not always readily available, and in fact it is infeasible to have *a priori* knowledge for the set of ports that a new malware, potentially exploiting "zero-day" vulnerabilities, might use for scanning. We compare our approach with DarkVec in Sec. VI-B. We note also that DarkVec does not employ the results of clustering for change-point detection, which is one of the main themes of this paper.

Reactive Darknets and honeypots. Network telescopes can be considered as a special case of network honeypots [39]. While network telescopes are completely "passive" (i.e., never

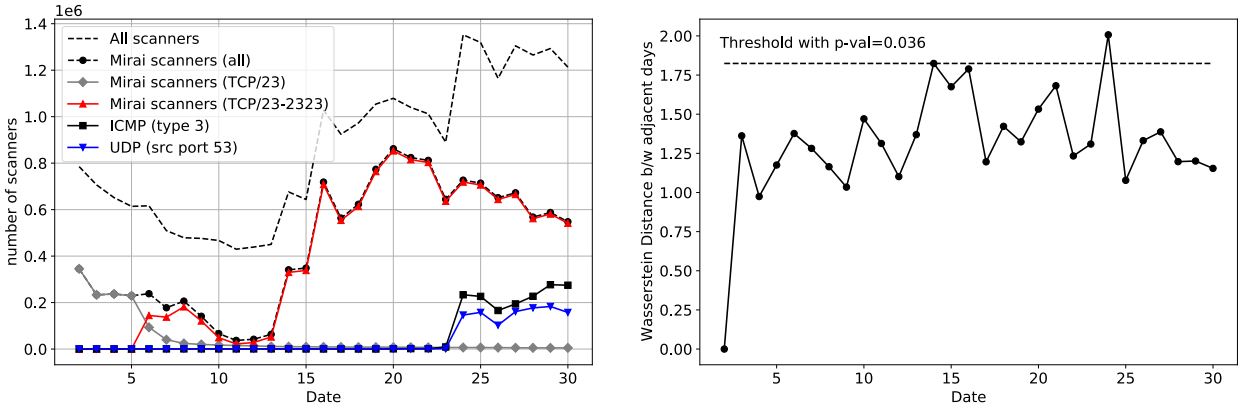

Fig. 1: (Left panel) Scanning traffic at Merit's Darknet (a /10 Darknet, back then) for September 2016. Notice the expansion of the Mirai botnet, namely the addition of TCP/2323 in the set of ports scanned. The figure considers scanners emitting at least 50 packets per day. (Right panel) Detection of temporal changes in the Darknet using the Wasserstein distance.

responding to traffic), honeypots can be reactive and *interact* with the scanners at various level of sophistication. The network security community has been employing honeypots for *situational awareness* for decades [40], oftentimes customizing them to specific goals, e.g., to understand DDoS amplification attacks [41], to study recent malware trends in the IoT ecosystem [42], [43], to understand SCADA systems [44], to target special types of scanning [45], etc. However, employing a large cluster of honeypots can be expensive since large compute and memory resources might be needed. Further, high-interaction honeypots could be hard to maintain and/or adapt to the changing malware landscape. Nevertheless, while we henceforth focus our attention in passive network telescopes, we note that our framework is applicable to data collected from honeypots, *mutatis mutandis*.

III. Problem Formulation

Network telescopes provide the unique opportunity to observe Internet-wide inconspicuous events. Our objective is to offer cyber-security analysts a framework that gleans useful insights in *near-real-time* from the vast amount of Darknet events that are captured in large network telescopes, hence enhancing their situational awareness regarding ongoing cyber-threats. To achieve this, we tackle the following problems.

Problem 1: Darknet Clustering. Consider N Darknet senders (i.e., scanners), each characterized by a high-dimensional feature vector $\boldsymbol{x} \in \mathbb{R}^P$. In this paper, we consider features compiled on a *daily* basis (e.g., total number of packets a scanner has sent within a given day, see Sec. IV). The goal is to assign the scanners into K groups such that "similar" scanners are classified in the same group. The notion of similarity is based on the "loss function" employed to solve the clustering problem and will be defined in the next sections.

Problem 2: Temporal Change-point Detection. Consider the clustering assignment matrices M_0 and M_1, denoting the clustering outcomes for day-0 and day-1, respectively. Here, $M_t \in \{0,1\}^{N\times K}$ is a binary matrix that denotes the cluster assignment for all N scanners, i.e., $M_t\mathbf{1}_K = \mathbf{1}_N$ for $t \in \{0,1\}$, where $\mathbf{1}_K$ and $\mathbf{1}_N$ are column vectors of ones of dimension K and N, respectively[1]. The task here is to detect *significant* changes between the clustering outcomes M_0 and M_1 that would denote that the Darknet structure changed between day-0 and day-1. As we will see later, we will cast this problem as the problem of comparing two multi-variate distributions, using ideas from optimal mass transport.

Henceforth, we assume that day-0 and day-1 are adjacent days and thus we are focusing our interest in detecting significant *temporal* Darknet structure shifts amongst consecutive daily intervals. Notably, the same approach could be utilized to compare Darknets across "space", namely to assess how dissimilar two Darknets that monitor different dark IP spaces might be. As shown in [45], there is evidence that the traffic that a Darknet receives is affected by the monitored IP space and the locality of the scanner. We leave these *spatial* Darknet comparisons as part of our future work.

IV. Data Description and Darknet Features

A. Darknet Data Description

Table I tabulates basic descriptive statistics for the Darknet datasets we employ in this study. Our main dataset spans the month of September 2016 and plays a critical role in demonstrating the benefits of the proposed approach. During September 2016, the Mirai botnet epidemic [3] gained significant speed, and thousands of compromised IoT devices got infected. The Mirai growth is evident in Figure 1 (left panel), where we see that hundreds of thousands of new victims started getting infected on September 14th. This coincides with a modification in Mirai's scanning strategy that involved the use of an additional port, namely TCP/2323, in Mirai's scanning campaigns. Albeit this strategy shift first occurred on September 6th, it was not implemented widely until eight days later. Should the change-point on September 14th got detected on an automated manner, Mirai's mitigation efforts

[1]The number of scanners N varies across different days. As shown next, this does not affect the generality of our approach. For notational convenience, we kept the numbers of scanners fixed in this problem description.

TABLE I: Basic statistics for our Darknet datasets.

Dates	Darknet Size	Sources	Packets	Ports	Top-3 ports		
					Port	Traffic (%)	Sources
[2016-09-02, 2016-09-30]	/10	35M	49B	65536	23	60.34	20.5M
					80	13.55	963K
					2323	4.00	13.5M
2016-09-14	/10	1.8M	1.5B	65536	23	53.30	808K
					2323	11.39	527K
					80	6.83	96K
2016-09-24	/10	3.3M	1.4B	65536	23	69.45	1.8M
					2323	7.00	1.3M
					80	3.73	84K
2022-02-20	/13	845K	3.1B	65536	6379	6.67	2.5K
					23	5.10	122K
					22	2.17	10.4K

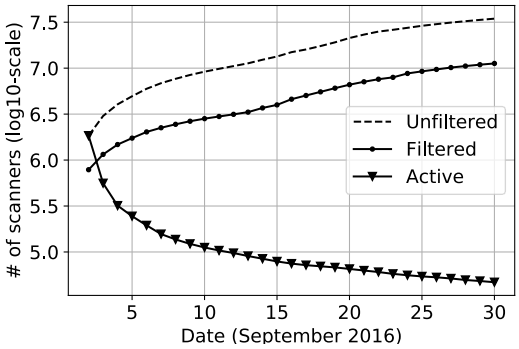

Fig. 2: Evolution of scanning activity over time.

would have started sooner and several attacks launched by the Mirai botnet in late September 2016 [3] could have been avoided. Our clustering and change-point detection framework aims to fill this gap.

The table illustrates that during the course of the month the Darknet observed about 35 million unique source IPs (two orders of magnitude more than the number of unique scanners involved in related works such as DANTE [37] and DarkVec [36]), and all possible (i.e., 65536) port destinations were targeted. One observes that Mirai-related ports (i.e., TCP/23 and TCP/2323) contribute close to 65% of the total traffic. The total number of source IPs associated with these ports and port TCP/80 is also illustrated in Table I. Figure 2 shows the evolution of scanners for the month of September 2016. We show the cumulative number of distinct source IPs seen in the Darknet over time both with and without *filtering*. Filtering is a "data pre-processing" step, employed to discard all scanning IPs that transmitted less than 50 packets within a given day. Filtering low-traffic scanners is a common technique used in Darknet analysis (e.g., applied also in [4], [5], [36]) and aims at reducing "noise" in the data, such as noise attributed to misconfigurations or randomly-spoofed source addresses. Further, keeping scanners with at least 50 packets transmitted is necessary for accurate estimation of some of the features we use to characterize the Darknet probers (e.g., for estimating the average packet inter-arrival times). Figure 2 demonstrates also the number of "active" scanners over time, i.e., scanners whose activity persisted throughout the month.

B. Darknet Features

We utilize an array of numerical and categorical features to characterize Darknet scanners. Figure 3 shows exemplar empirical CDFs for the numerical features used in our study. The features shown are compiled for the *filtered* scanners of September 14th, 2016 (see Table I). The CDFs illustrate the richness and complexity of the Darknet ecosystem in terms of traffic volume received from senders (e.g., see *packets*, *bytes* and *average inter-arrival time*), scanning strategy (e.g., see *number of distinct destination ports* and *number of distinct destination addresses* scanned), etc. We now briefly describe each of the features utilized in our study.

Traffic volume. A series of features characterize the volume and frequency of scanning, namely total number of *packets* transmitted within the observation window (i.e., a day), total *bytes* and *average inter-arrival* time between sent packets. Observe the large spectrum of values that these features exhibit; for instance, Figure 3 shows that some scanners send only a few packets (i.e., as low as 50 packets, our lower bound for filtered traffic) while some emit tens of millions of packets in the Darknet, aggressively foraging for Internet victims.

Scan strategy. Features such as *number of distinct destination ports* and *number of distinct destination addresses* scanned within a day, *prefix density*, *destination strategy*, *IPID strategy* and *IPID options* reveal information about one's scanning strategy. For instance, we see some senders to only focus on a small set of ports (about 90% of the scanners on September 14th targeted up to two ports) while others target all possible ports. Prefix density is defined as the ratio of the number of scanners within a routing prefix over the total IPs covered by the prefix (we use CAIDA's *pf2as* dataset for mapping IPs to their routing prefix [46]), and can provide information about coordinated scanning within a network. *Destination strategy* and *IPID strategy* are features that show 1) whether the scanner kept the associated fields (i.e., destination IP and IPID) constant, 2) with fixed increments or 3) were kept random. These could provide useful insights about the scanning intentions and/or tools used for scanning (for instance, the ZMap tool [47] uses a constant IPID of 54321). *TCP options* is a binary feature that illustrates whether any TCP options have been set in TCP-related scanning; the authors of [45] associate the lack of TCP options with "irregular scanning" (usually associated with heavy, oftentimes nefarious, scanning) and thus we decided to track this as part of our features.

Targeted applications. We employ the features *set of ports* and *set of protocol request types* scanned to glean information about the services being targeted. Since there are 2^{16} distinct ports, we encode—using the *one-hot-encoding* scheme—the set of ports scanned using the top-500 ports identified on September 2nd, 2016. I.e., if a scanner had scanned ports outside the top-500 set, its one-hot-encoded feature for ports would be all zeros. Table II shows some of the protocol types we consider (e.g., TCP-SYN request, UDP, etc.). The table shows the top-5 combinations scanned for September 2nd, 2016; when compiling our features vector, this information

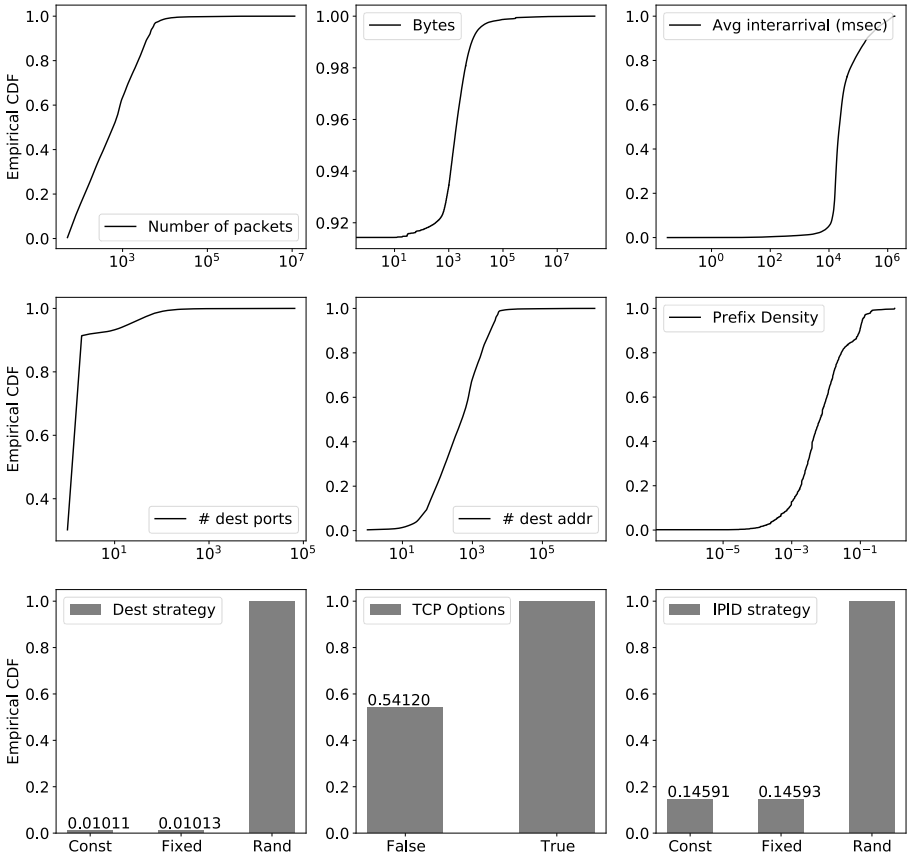

Fig. 3: CDFs for our numerical features that characterize scanning activity. Data source: Merit's Darknet, 2016-09-14.

TABLE II: Traffic types.

Traffic Type	**Fraction of Scanners** (%)
TCP-SYN	91.17
TCP-SYN, UDP	4.04
UDP	2.48
ICMP Echo Request	0.61
TCP-SYN, UDP, ICMP Dest. Unreachable	0.47

is also encoded using an *one-hot-encoding* scheme.

Device or scanner type. We use the *set of TTL values* seen per scanner as an indicator for "irregular scan traffic" [45], [48] and/or the device OS type [49]. For instance, IoT devices that usually run on Linux/Unix-based OSes are seen with TTL values within the range 40–60 (the starting TTL value for Linux/Unix OSes is 64). On the other hand, devices with Windows are seen scanning the Darknet with values in the range 100–120 (starting value for Windows OSes is 128) [49].

V. Methodology

In this section we present the proposed approach that tackles the problems introduced in Sec. III.

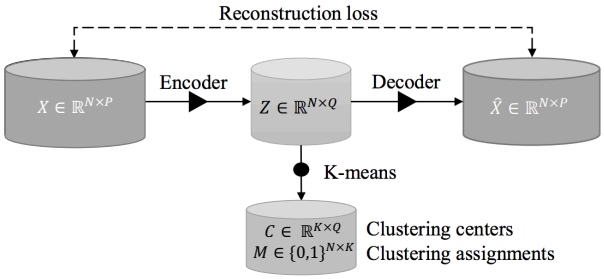

Fig. 4: Autoencoder for dimensionality reduction.

A. Dimensionality Reduction via Representation Learning

A key first step in our methodology is to project the high-dimensional input features onto a low-dimensional vector space of *embeddings* that preserves the information from the input signal and is amenable to clustering. The heterogeneity of the input data features, their high dimensionality, and the need to cope with potentially nonlinear interactions between features motivated us to employ *nonlinear autoencoders* to

address these challenges.

Inspired by recent advances in deep representation learning [50], [51], *deep autoencoders* [34], [52]–[54] receive the input signal $\boldsymbol{x} \in \mathbb{R}^P$ and try to "compress" it in a lower-dimensional space of embeddings $\boldsymbol{z} \in \mathbb{R}^Q, Q \ll P$, while also preserving as much information as possible from the input signal (see Figure 4).

A typical nonlinear autoencoder construction is described next. Let $e_\theta(\cdot)$ be a nonlinear encoder function parameterized by θ that maps the input data to a representation space of embeddings, and $d_\mu(\cdot)$ be a nonlinear decoder function parameterized by μ that maps the data points from the representation space to the input space, such that:

$$e_\theta(\boldsymbol{x}_i) = f(\boldsymbol{x}_i;\theta) =: \boldsymbol{z}_i, \qquad f(\cdot;\theta) : \mathbb{R}^P \to \mathbb{R}^Q$$
$$d_\mu(\boldsymbol{z}_i) = g(\boldsymbol{z}_i;\mu) =: \hat{\boldsymbol{x}}_i, \qquad g(\cdot;\mu) : \mathbb{R}^Q \to \mathbb{R}^P$$

We employ the fully-connected *multilayer perceptron* (MLP) neural network for the implementation of both mapping functions $f(\cdot;\theta)$ and $g(\cdot;\mu)$. For instance, a 4-layer MLP network realizing the encoding function $f(\cdot;\theta)$ is defined as:

$$\boldsymbol{h}^{(1)} = \phi\big(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\big)$$
$$\boldsymbol{h}^{(2)} = \phi\big(\boldsymbol{W}^{(2)}\boldsymbol{h}^{(1)} + \boldsymbol{b}^{(2)}\big)$$
$$\boldsymbol{y} = \phi\big(\boldsymbol{W}^{(3)}\boldsymbol{h}^{(2)} + \boldsymbol{b}^{(3)}\big) =: f(\boldsymbol{x};\theta),$$

where $\phi(\cdot)$ is the activation function, applied element-wise to all elements of its input vector. We elected the ReLU activation function to avoid the "gradient vanishing" problem that other nonlinear functions exhibit (e.g., the sigmoid function) [55], [56]. In the above formulation, the parameters θ of the neural network that need to be "learned" are the weight matrices $\boldsymbol{W}^{(i)}$ and the bias vectors $\boldsymbol{b}^{(i)}$, $i = 1, 2, 3$. The vectors $\boldsymbol{h}^{(\ell)}$ denote the output of the l-th hidden layer.

The decoder function $g(\cdot;\mu)$ uses the same number of hidden layers and layer sizes as the encoding function. The ReLU activation functions are also employed in all layers except the last one; a linear activation function is used at the output layer so that the output vectors take values in $\mathbb{R}^P$.

To learn representations that preserve the information of input data, we consider minimizing the reconstruction loss:

$$\min_{\theta,\mu} \sum_{i=1}^{N} \left(\ell(g \circ f(\boldsymbol{x}_i), \boldsymbol{x}_i)\right) + \lambda(R(\theta) + R(\mu)) \tag{1}$$

where the $\ell(\cdot) : \mathbb{R}^P \to \mathbb{R}$ is a loss function that quantifies the reconstruction error. For simplicity, we choose the Euclidean distance $\ell(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$. $R(\cdot)$ is a regularization term for the model parameters to help us avoid "overfitting" the data. In this work, we adopt the squared ℓ_2 norm, such that $R(\theta) = \|\theta\|_2^2$. $\lambda \geq 0$ is the regularization coefficient. All model parameters, i.e., $\{\theta, \mu\}$, can be jointly learned using standard stochastic gradient-based optimization methods, such as Adam [57].

Autoencoder tuning. Tuning is discussed in Appendix A. As shown there, the optimal MLP parameters are as follows: For the encoder function, we use a 3-layer network with an input layer equal to the dimension P of the features, a second (hidden) layer of size 1000, and an output layer of size $Q = 50$ neurons. The decoding function is also a 3-layer network: the first layer is the layer of embeddings of size $Q = 50$, the second layer is a hidden one of size 1000 and finally we have the output layer of size P. The regularization weight is chosen to be $\lambda = 0.001$ (see Figure 10). Unless otherwise noted, these are the MLP parameters we have employed in the evaluation sections of the paper (see Section VI).

B. Clustering Darknet scanners

After the representation learning step is done, the "trained" encoding function $f(\cdot;\theta)$ is employed to yield the *embeddings* of the scanners to be utilized for clustering (see Figure 4). We perform standard *K-means* clustering on the low-dimensional representation of the data. Formally, in this step, we aim to minimize the following clustering loss:

$$\min_{C,M} \sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i), MC) \tag{2}$$
$$\text{s.t. } m_{i,j} \in \{0,1\},\ M\mathbf{1}_K = \mathbf{1}_N,$$

where $M = (m_{i,j})_{N\times K}$ is the clustering assignment matrix, the entries of which are all binary. $C = (c_{i,j})_{K\times Q}$ is the matrix of clustering centers that lie in the representation space. $\mathbf{1}_K$ and $\mathbf{1}_N$ are column vectors of ones of dimension K and N. We employ the Euclidean distance as the loss function $\ell(\cdot)$.

C. Change-point Detection via Optimal Mass Transport

The clustering outcomes obtained are utilized both for characterizing the Darknet activities within a monitoring window (e.g., a full day) and for detecting temporal changes in the Darknet's structure (e.g., the appearance of a new cluster associated with previously unseen scanning activities). To accomplish the latter, we employ techniques from the theory of *optimal transport* also known as *Earth mover's distance*. We describe our change-point detection approach next, after first introducing the necessary mathematical formulations.

Optimal Transport: Background. We base our exposition and notation on [11]. Optimal transport serves several applications in image retrieval, image representation, image restoration, etc. [11]. Its ability to "compare distributions" (e.g., comparing two images) fits our need to "compare clustering outcomes" between days.

Let I_0 and I_1 denote *probability density functions* (PDFs) defined over spaces Ω_0 and Ω_1, respectively. Typically, Ω_0 and Ω_1 are subspaces in $\mathbb{R}^d$. In the *Kantorovich formulation* of the optimal transport problem[2] we are interested in finding a *transport plan* that "transforms" I_0 to I_1. The plan, denoted with function γ, can be seen as a joint probability distribution of I_0 and I_1 and the quantity $\gamma(A \times B)$ describes how much mass in set $A \in \Omega_0$ is transported to set $B \in \Omega_1$. In the Kantorovich formulation, the transport plan γ must (i) meet the *constraints* $\gamma(\Omega_0 \times B) = I_1(B)$ and $\gamma(A \times \Omega_1) = I_0(A)$,

[2]There is also the *Monge formulation*, which does not suit our needs.

where $I_0(A) = \int_A I_0(x)dx$ and $I_1(B) = \int_B I_1(x)dx$ and (ii) *minimize* the following quantity:

$$\min_{\gamma} \int_{\Omega_0 \times \Omega_1} c(x,y)d\gamma(x,y),$$

for some *cost function* $c : \Omega_0 \times \Omega_1 \to \mathbb{R}^+$ that represents the cost of moving a unit of mass from x to y.

Application to Darknet clustering. In the Darknet clustering setting, we consider the discrete version of the Kantorovich formulation. The PDFs I_0 and I_1 can now be expressed as $I_0 = \sum_{i=1}^{K} p_i \delta(x - x_i)$ and $I_1 = \sum_{j=1}^{K} q_j \delta(y - y_j)$, both defined over the same space Ω, where $\delta(x)$ is the Dirac delta function. The optimal transport plan problem now becomes

$$K(I_0, I_1) = \min_{\gamma} \sum_i \sum_j c(x_i, y_j)\gamma_{ij} \tag{3}$$
$$\text{s.t.} \sum_j \gamma_{ij} = p_i, \sum_i \gamma_{ij} = q_j$$
$$\gamma_{ij} \geq 0, i, j = 1 \ldots, K.$$

Solutions to this problem can be obtained using standard *linear programming* methods. Further, when the cost function is $c(x, y) = |x - y|^p, p \geq 1$, the optimal solution of (3) defines a *metric* on $P(\Omega)$, i.e., the set of probability densities supported on space Ω. This metric is known as *p-Wasserstein distance* and is defined as

$$W_p(I_0, I_1) = \Big(\sum_i \sum_j |x_i - y_j|^p \gamma^*_{ij} \Big)^{\frac{1}{p}}, \tag{4}$$

where γ^* is the optimal transport plan for (3).

Our proposed approach is to employ the 2-Wasserstein distance on the distributions I_0 and I_1 that capture the clustering outcomes M_0 and M_1, where $M_u, u = 0, 1$, are the clustering assignment matrices for two adjacent days (see Sec. III). Let X_0 and X_1 denote the $N \times P$ matrices that represent the scanner features (see Figure 4) for the two monitoring windows. Define

$$D_u = M_u^\top \mathbf{1}_N \tag{5}$$
$$C_u = \big(X_u^\top M_u\big)\text{diag}(D_u^{-1}), \quad u = 0, 1.$$

Namely, the i-th entry of vector D_u denotes the cluster size of the i-th cluster of scanners identified for day-u, and the i-th row of matrix C_u represents the clustering center of cluster i. Hence, the weights and Dirac locations for the discrete distributions $I_0 = \sum_{i=1}^{K} p_i \delta(x - x_i)$ and $I_1 = \sum_{j=1}^{K} q_j \delta(y - y_j)$ are readily available; e.g., the weight p_i for cluster-i of day-0 corresponds to the size of that cluster normalized by the total number of scanners for that day, and location x_i corresponds to the center of cluster i. Thus, one can obtain the distance $W_2(I_0, I_1)$ and optimal plan γ^* by solving the minimization shown in (3).

As we demonstrate in Sec. VI, one can utilize distance $W_2(I_0, I_1)$ and the associated optimal plan γ^* to (i) *detect* and (ii) *interpret* clustering changes between consecutive monitoring windows. Specifically, an alert that signifies a change in the clustering structure can be triggered when the distance $W_2(I_0, I_1)$ is "large enough". To the best of our knowledge, there is no *test statistic* for the multivariate "goodness-of-fit" problem we are interested in (unlike the univariate case [58]). Thus, we resort to detecting anomalies via the use of historical / empirical values of the $W_2(I_0, I_1)$ metric that one can collect (see, e.g., Figure 1). When an alert is flagged, the optimal plan γ^* is leveraged to shed light into the clustering change.

TABLE III: Linear (PCA) vs. nonlinear (MLP) autoencoder: comparisons over 20 Monte Carlo experiments.

	Nonlinear AE (MLP)		**Linear AE (PCA)**	
Q	**Jaccard** (μ/σ)	**Loss** (μ/σ)	**Jaccard** (μ/σ)	**Loss** (μ/σ)
10	0.87/0.07	0.08/0.002	0.56/0.12	0.15/0.001
20	0.88/0.07	0.08/0.001	0.82/0.09	0.14/0.001
50	0.92/0.09	0.07/0.001	0.90/0.09	0.15/0.001

VI. Evaluation

A. Evaluation Using Synthetic Data

Due to the lack of "ground truth", evaluating unsupervised machine learning methods, like clustering, is challenging. To tackle this problem we generate synthetic data, i.e., artificially generated data that mimic real data. The advantage of such data is that we can introduce different "what-if" scenarios to evaluate different aspects of our framework. Appendix B describes our process for generating the synthetic datasets.

Embeddings evaluation: linear vs nonlinear autoencoders. We first leverage the synthetically-generated datasets to compare the performance of our MLP-based encoder against other dimensionality reduction techniques such as Principal Component Analysis (PCA). PCA is considered a "linear autoencoder" whereas our MLP implementation is specifically designed to be "nonlinear" in order to better capture the complex interactions between the employed Darknet features.

Table III tabulates the results. We created 20 distinct synthetic datasets using the procedure outlined in Appendix B. Each dataset corresponds to $K = 50$ clusters. For each dataset, we first execute the representation learning step. In the representation learning step we acquire embeddings based on PCA and the MLP autoencoder described in Sec. V-A. Then, we perform clustering on the learned embeddings using K-means (see Figure 4) with $K = 50$. We record the overall performance in terms of the *Jaccard score*. Performance of the representation learning step is captured via the *reconstruction loss*. Table III reports the average (μ) and standard deviation (σ) across the 20 Monte Carlo experiments realized. In our experiments, we used an MLP autoencoder with an encoding function with 3 layers: input layer, hidden layer of 1000 units and output layer for the embeddings of size Q. For PCA, we used Q principal components, as shown in Table III; in all cases, the variance explained by the top-Q principal components used exceeded 90%.

As shown, the MLP-based autoencoder outperforms the PCA-based linear autoencoder. We attribute the performance benefit of the MLP encoder to its ability to better capture the nonlinearities in the supplied features.

Change-point detection: sensitivity to clustering changes. Next, we utilize the synthetic data to understand the sensitivity of the Wasserstein metric shown in Eq. (4) under two basic

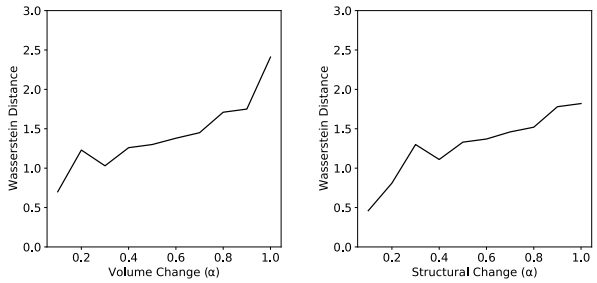

Fig. 5: Wasserstein distance changes significantly to reflect changes in volume (left) and structure (right).

practical scenarios: 1) Darknet changes due to an increase in the volume of scanning attributed to a particular cluster and 2) Darknet shifts due to structural changes (e.g., the cluster population stays the same, but its scanning strategy changes).

Under these considerations, the sensitivity of the Wasserstein distance is explored in Figure 5. The left panel shows the evolution of the $W_2(I_0, I_1)$ metric as the volume of a particular cluster increases. Parameter α shown in the x-axis controls the expansion rate. E.g., $\alpha = 0.1$ signifies a 10% volume increase. The right panel illustrates the $W_2(I_0, I_1)$ sensitivity when structural changes occur; in this case, we model a structural change with the introduction of an additional port being scanned (i.e., a second port). In this scenario, α dignifies the portion of scanners in the changing cluster that scan for the additional port. As we observe, our dissimilarity metric is sensitive enough to detect the simulated Darknet shifts.

B. Comparison with Related Work using Real-World Data

This section juxtaposes our methodology with state-of-the-art related work, namely the DarkVec approach [36]. DarkVec's authors allow researchers to access their code and data [59], and we based our comparisons on the provided data. Specifically, we utilize the last day of the 30-day dataset used in [36] (see Table 1, [36]).

We employ the same *semi-supervised* approach that DarkVec used for its comparisons with other methods. Since no "ground truth" exists for clustering labels when working with real-world Darknet data, the authors in [36] assigned labels based on domain knowledge; e.g., known scans projects (i.e., Censys [35], Shodan [60], etc. [61]) and known signatures such as the Mirai one [3]; an "unknown" label is assigned to the rest of the senders. The complete list of the nine "ground truth" labels utilized can be found in [36] (Table 2).

The semi-supervised approach evaluates the quality of the learned embeddings. Intuitively, the embeddings of all scanners belonging in the same "ground truth" class (e.g., Mirai) should be "near" each other according to some appropriate measure. The semi-supervised approach engaged in [36] involves the usage of a k-Nearest-Neighbor (k-NN) classification algorithm that assigns each scanner to the class of its k-nearest neighbors based on a majority voting rule. Using the *leave-one-out* approach, each scanner is assigned a label, and the overall classification accuracy is evaluated using standard metrics such as *precision* and *recall*.

We construct the autoencoder-based embeddings needed for our approach (see Sec. V-A) on the last day of the 30-day dataset used in [36]. The DarkVec embeddings, which are acquired via word embeddings techniques such as Word2Vec, are readily available (see [59], dataset `embeddings_d1_f30.csv.gz`). Using this dataset, DarkVec was shown to perform better than alternatives such as IP2VEC (see Table 3, [36]) and thus we solely focus our comparisons against DarkVec. Table IV tabulates our results. The semi-supervised approach using our embeddings shows an overall F1-score of 0.98 whereas DarkVec's embeddings lead to a classification accuracy score[3] of 0.90. The higher accuracy of our approach can be attributed to the quality of the embeddings; we believe that the set of features we employ leads to embeddings that more accurately reflect the (dis)-similarities between the scanners than competing approaches.

C. Validation using Real World Darknet Data

In this section, we validate our approach using real-world data (see Table I). First, we evaluate the complete methodology on a month-long dataset that includes the outset of the Mirai botnet (see Figure 1). Then, we apply our clustering approach on a recent dataset (i.e., February 20, 2022) to showcase some important recent Darknet activities that our system diagnoses.

September 2016: The Mirai onset. We applied our full methodology on data for September 2016 (starting on 2016-09-02); i.e., for each day, we obtain the necessary embeddings, cluster the scanners to 200 groups (see Appendix A), and then apply the techniques of Sec. V-C to calculate the Wasserstein metric and associated transport plan between consecutive days.

Figure 1 (right panel) shows the time-series of 2-Wasserstein distances for September 2016. As can be seen, at a significance level of 5%, we *identify two change-points*; one for September 14th (with p-value=0.036) and another for September 24th (with p-value=0). (On September 16th, we have p-value=0.071.) The p-values are calculated using the set of all Wasserstein distances estimated for the whole month[4].

We next utilize the optimal transport plan γ^* to *interpret* the change-points detected. Let $G = (V, E)$ be a *weighted directed graph* with $V := \{A_u\} \cup \{B_u\}, u = 1, \ldots, K$, denoting the graph's nodes, where node A_u corresponds to cluster-u in day-0 and B_u to cluster-u in day-1, respectively. $(u, v) \in E$ if and only if $\gamma^*_{uv} > 0$, i.e., there is some amount of mass transferred from cluster-u of day-0 to cluster-v of day-1 (see Sec. V-C). The edge weights $w_{uv}, (u, v) \in E$ are defined as $w_{uv} := \gamma^*_{uv}$. The graph in Figure 6 shows the graph extracted based on the optimal transport plan γ^* for the clustering outcomes of September 13 and September 14. It can shed light into the clustering changes that occurred between the two days. For instance, Figure 6 and Table VI

[3]In [36] (Table 3) an accuracy of 0.96 is reported. This is due to a subtle error in DarkVec's code for calculating the classification accuracy (see [59], `src/utils.py`, function `get_freqs`, lines 207 and 214). We fixed the error, which does not affect the main conclusions/contributions in [36], and report the updated results here. Without the fix, our accuracy would (incorrectly) be 1.00.

[4]In a real-world implementation of our system, historical Wasserstein values can be used (e.g., values from the previous month).

TABLE IV: Comparison with DarkVec [36].

	DarkVec Embeddings			Autoencoder Embeddings (Section V-A)			
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Support**
Mirai-like	1.00	0.91	0.95	1.00	0.99	0.99	7351
Binaryedge	0.98	0.93	0.95	1.00	0.92	0.96	101
Censys	0.99	0.90	0.94	1.00	0.98	0.99	336
Engin-umich	1.00	1.00	1.00	1.00	1.00	1.00	10
Internet-census	0.99	0.99	0.99	1.00	0.89	0.94	103
Ipip	0.45	0.67	0.54	1.00	0.92	0.96	49
Sharashka	0.83	1.00	0.91	1.00	1.00	1.00	50
Shodan	1.00	0.70	0.82	1.00	0.74	0.85	23
Stretchoid	1.00	0.14	0.25	1.00	1.00	1.00	104
Accuracy			**0.90**			**0.98**	

show that most mass is moved from cluster A_{10} (largest cluster of September 13) to cluster B_{18}. Examining Table VI we observe that these Mirai-like clusters are quite similar with regards to the features that characterize their scanners. The fact that B_{18} is a much smaller Mirai cluster than A_{10} suggests that there was a decreasing trend in the amount of Mirai-related scanners that solely targeted port TCP/23. Indeed, the second largest mass transfer was between A_1 and B_{34}, and in this case we clearly see that cluster B_{34} captures the introduction of port TCP/2323 in the set of ports scanned by Mirai (see Table VI). Similar insights can be obtained by inspecting cluster pairs (A_{25}, B_{56}), (A_{20}, B_{11}), (A_{28}, B_{52}), (A_{38}, B_{96}), and others not shown here for space economy. By inspecting Figure 1 (left) one can validate that the change between the 2 days can actually be attributed to the changing tactics of the Mirai botnet. Note, though, that without the automated methodology proposed here, capturing this change would require monitoring an enormous amount of time series (e.g., the scanning traffic to all ports) which is practically infeasible.

As shown in Figure 1, the most significant clustering change was detected on September 23–24. Surely, in Figure 1 (left) we see a dramatic increase in the amount of Darknet traffic associated with UDP scanning and ICMP messages with Type 3 (Destination Unreachable). Upon closer inspection, we see UDP with `src port 53` and ICMP messages with the message `destination port 53 unreachable`. The payload of these messages point to the conclusion that these are indicators of heavy nefarious DNS scanning, captured in the Darknet as "DNS backscatter" [48]. Within the UDP and ICMP packets we see DNS A-record queries under the domain `xy808.com`, with randomly looking subdomains. This is a common technique that scanners embrace in order to identify open DNS resolvers while at the same time concealing their identity. The list of compiled open DNS resolvers can then be used in volumetric, reflection and amplification DDoS attacks [62]. To put things in perspective, we note that some of the largest Mirai-based DDoS attacks occurred on September 25th (against Krebs on Security) and on October 21st, 2016 (against Dyn) [3]. We thus speculate that the Mirai operators were the ones behind these heavy DNS scanning activities.

Having confirmed that the change-point for September 23–24 is a "true positive" malicious event, we consult the optimal transport plan γ^* to see how one can interpret the alert raised. Table VII tabulates the top-6 pairs of clusters with the largest amount of "mass" transferred. The pair (A_{47}, B_{24}) indicates there was high transfer of mass to cluster B_{24} which is associated with ICMP (type 3) activities. In contrast with the other row-pairs in the table, the fact that mass gets transferred from A_{47} to B_{24} indicates the formation of a novel cluster; the Jaccard similarity between the set of source IPs of the 2 clusters is zero, and their scanning profile varies significantly.

Figure 7 shows the in-degrees for the graph G induced by the optimal transport plan of September 23–24. In the three panels shown, we pruned the edges for which $\gamma^*_{uv} < \tau$, where threshold $\tau \in \{5 \times 10^{-4}, 0.001, 0.003\}$. Notice that cluster B_{123} stands out as the one with the highest in-degree in all three cases. The fact that the "optimal transport plan" includes transferring high amounts of mass from several different clusters (of the previous day) to cluster B_{123} indicates that the latter is a novel cluster. Indeed, the members of B_{123} are associated with UDP messages with `src port 53`, and as illustrated in Figure 1 this activity started on September 24th.

Cluster inspection: 2022-02-20 dataset. Next, we discuss recent activities identified in the Darknet when our clustering approach is applied. We focus our attention on the dataset for February 20th, 2022 (see Table I). In total, Merit's Darknet observed 845,000 scanners for that day; after the filtering step a total of 223,909 senders remain. They are grouped into the categories shown in Table V.

We found 70 *Mirai-related* clusters comprised of 108,912 scanners. We classify them as "Mirai-related" due to the destination ports they target and the fact that their traffic type (see Table II) is TCP-SYN. Note that we do not observe the characteristic Mirai fingerprint in all of them (i.e., setting the scanned destination address equal to the TCP initial sequence number [3]). This implies the existence of several Mirai variants. In fact, we see several combination of ports being scanned, such as "23", "23-2323", "23-80-8080", "5555" and even largest sets like "23-80-2323-5555-8080-8081-8181-8443-37215-49152-52869-60001". The vast majority of these clusters appear with Linux/Unix-like TTL fields, indicating they are likely compromised IoT/embedded devices [63].

The next large category of Darknet scanners is one with unusual activities that we cannot attribute to some known malware or specific actor; we deem these activities as "Unknown". Their basic characteristics are that they involve mostly UDP traffic and target "high-numbered" ports such as port 62675. Upon inspection of the TTL feature, these group of clusters includes both Windows and Linux/Unix OSes. For many of these clusters, the country of origin for these scanners is China.

We identified 20 clusters associated with TCP/445 scanning, i.e., the SMB protocol. Several ransomware-focused malware

(such as WannaCry) are known to be aiming to exploit SMB-related vulnerabilities [64], [65]. Members of these clusters are usually Windows machines.

Further, we detected a plethora of "heavy scanners", some performing scanning for benign purposes (e.g., Censys.io [35], Shodan [60]) and others engaged in nefarious-looking activities. Four clusters comprise of almost exclusively of *acknowledged scanners*[5], i.e. IPs from research and other institutions that are believed to not be hostile [61]. Four other clusters (three from Censys and one from Normshield [66]) are also benign clusters that scan from IPs not yet included in the "acknowledged scanners" list [61]. Some clusters in the "Heavy Scanners" category exhibit interesting behavior; e.g., 1) some scan with extremely high speeds (five clusters have mean packet inter-arrival times less than 10 msecs), 2) ten clusters probe all or (close to all) IPs that our Darknet monitors, 3) two clusters scan almost all 2^{16} ports, 4) one cluster sends an enormous amount of UDP payload to 16 different ports, and 5) two clusters are engaged in heavy SIP scanning activities.

We also identified a cluster associated with TCP/6379 (Redis) scanning comprising of 437 scanners. Interestingly, Table I shows that TCP/6379 is the most scanned port in terms of packets on 2022-02-20. Our clustering procedure grouped this activity within a single cluster which indicates orchestrated and homogeneous actions (indeed, members of that cluster scan extremely frequently, probe almost all Darknet IPs, are Linux/Unix-based, and originate mostly from China). We further uncovered two clusters performing TCP/3389 (RDP) scanning, two clusters targeting UDP/5353 (i.e., DNS) and two clusters that capture "backscatter" activities, i.e., DDoS attacks based on spoofing [13].

Figure 8 demonstrates the average *silhouette score* for each cluster of the 2022-02-20 dataset. The silhouette score [67] takes values between -1 (worst score) and 1 (perfect score), and indicates if a cluster is "compact" and "well separated" from other clusters. We annotate the plot of silhouette scores with some clusters associated with orchestrated scanning activities: the 4 clusters of "Acknowledged Scanners", the 3 "Censys" clusters, the cluster for Normshield, and 18 clusters from the "Heavy Scanners" category (the left-out cluster includes only a single scanner corresponding to NETSCOUT's research scanner [68]; the silhouette score for singleton clusters is undefined). We chose clusters like these since their members (i.e., the senders) are usually engaged in similar behavior (e.g., sending about the same amount of packets, targeting the same number of ports, etc.) and are thus good examples to demonstrate our clustering performance. As expected, the silhouette scores for the vast majority of these clusters are quite good (≥ 0.33). However, for few clusters the silhouette score is close to 0. While we still get meaningful insights from these clusters (e.g., cluster 162, with score -0.01, indicates extreme scanning activity against almost all Darknet IPs with its members scanning an average of 5,753 unique ports), their silhouette score is low because of intra-cluster variability in some of their features (e.g., the TTL values). If necessary, the analyst can resort to hierarchical clustering and re-partition the clusters with low scores.

[5]We label a cluster as "Ack Scanners" if at least 95% of its IPs are in [61].

TABLE V: Cluster Inspection (2022-02-20).

Description	# of Clusters	# of Senders
Mirai-related	70	108,912
Unknown	67	76,525
SMB	20	23,700
Heavy Scanners	19	2,377
ICMP scanning	5	2,619
Ack Scanners	4	795
SSH scanning	4	2,635
censys.io	3	147
TCP/3389 (RDP)	2	1,482
UDP/5353	2	3,212
Backscatter (DDoS)	2	815
TCP/6379 (Redis)	1	437
Normshield	1	253
TOTAL	**200**	**223,909**

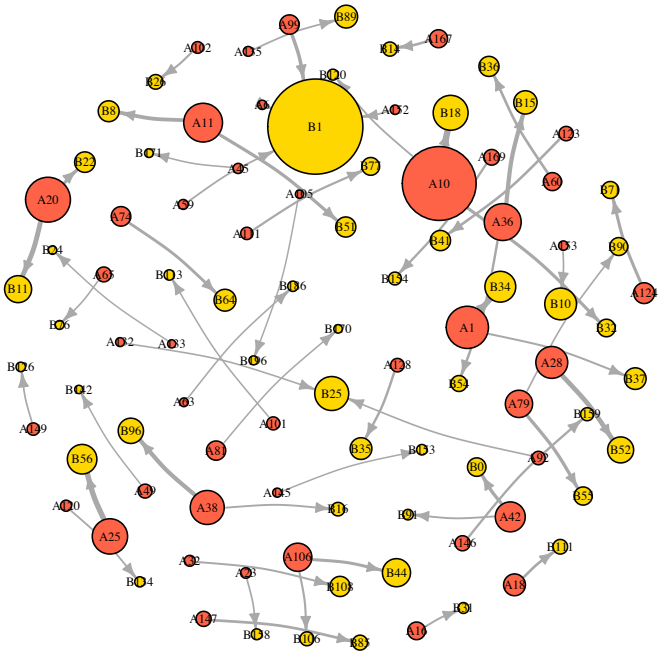

Fig. 6: Optimal transport plans for Sept. 13–14. Only edges with $\gamma^*_{uv} \geq 0.01$ are shown.

Figure 9 shows t-SNE visualizations [69] for some select clusters. We illustrate some clusters of acknowledged / heavy scanners that exhibit high average silhouette scores. We also depict the largest cluster for each of these categories: Mirai, "Unknown", SMB, ICMP scanning and UDP/5353. The t-SNE projections are learned from the 50-dimensional embeddings acquired from our autoencoder step. Thus, the signal is quite compressed; nevertheless, we are still able to observe that similar scanners are represented with similar embeddings.

VII. Conclusion

This paper presents a framework for (i) characterizing network scanners captured in large network telescopes and for (ii) detecting and interpreting temporal changes in the telescope's ever-evolving structure. The proposed approach employs a nonlinear autoencoder, implemented as a multilayer perceptron, to learn a low-dimensional representation of the

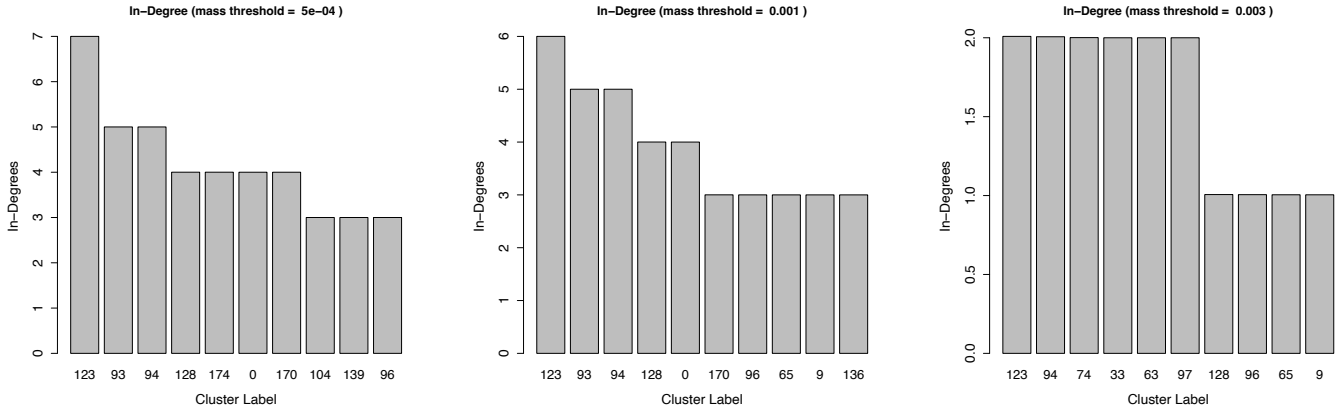

Fig. 7: In-degree distributions of the graph induced by the optimal plan γ^* for Sept. 23–24. The x-axis shows cluster labels for Sept. 24th. Cluster 123, associated with the *DNS scans* that started on September 24th (see Figure 1), exhibits the highest in-degree. High in-degree can be considered an indication of a "novel" cluster, i.e., a cluster not present in the previous day.

.

TABLE VI: Interpretation of clustering changes between September 13 and September 14, 2016. The table includes the pairs of clusters with the largest amount of mass transferred between the two days. It also shows the basic profile for each cluster in terms of its numerical features (as captured by the cluster's center) and the top frequency for the set-valued features (TTLs, protocol/traffic types, ports). The Jaccard column indicates the Jaccard similarity between the scanning IPs of the 2 clusters.

Day	Label	Mass	Jaccard	Size	Packets	Avg. IA (ms)	Bytes	# DstPorts	# DstAddr	TTL	Freq.	Traffic	Freq.	Ports	Freq.
13	10			21247	1513	24586	0	1.1	854	50	19628	TCP-SYN	21213	23	19674
14	18	0.022	0.18	15174	1313	29594	0	1.5	904	50	13300	TCP-SYN	15137	23	8345
13	1			12145	1821	29673	0	1.1	1058	53	11906	TCP-SYN	12139	23	11391
14	34	0.020	0.17	13410	1145	29472	0	1.6	815	53	12834	TCP-SYN	13408	23-2323	6911
13	25			10236	1669	27095	0	1.2	960	49	9762	TCP-SYN	10235	23	9438
14	56	0.019	0.18	12862	1412	27172	0	1.3	975	49	11186	TCP-SYN	12861	23	9113
13	20			12906	2259	29468	0	1.7	1107	47	12193	TCP-SYN	12891	23	11982
14	11	0.017	0.18	11744	1343	33147	0	2	824	47	11233	TCP-SYN	11730	23	7291
13	28			9244	2058	28640	0	1.8	1148	45	8944	TCP-SYN	9235	23	8759
14	52	0.017	0.12	11312	1179	30244	0	2.1	850	45	10842	TCP-SYN	11303	23-2323	6055
13	38			9851	2369	27181	0	1.8	1277	46	9705	TCP-SYN	9730	23	9465
14	96	0.015	0.15	11001	1391	35617	0	2.2	920	46	8559	TCP-SYN	10800	23	7936

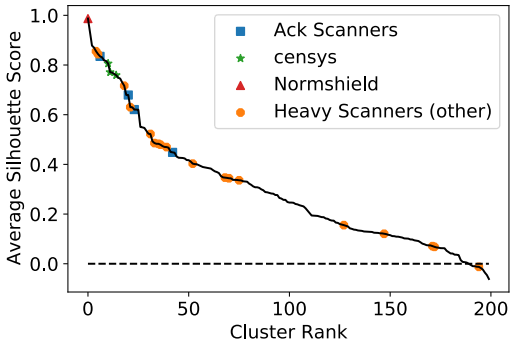

Fig. 8: Average silhouette score for all clusters (2022-02-20).

input space that is amenable to clustering. The resulted clustering outcomes are then utilized to detect Darknet shifts using a metric that arises from optimal mass transport theory. The techniques proposed are robustly evaluated, including using real Darknet data from Merit's Network Telescope.

ACKNOWLEDGEMENTS

This material is based upon work partially supported by the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-05-00, and the National Science Foundation under awards CNS-1823192 and CNS-2120400. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government.

APPENDIX A
HYPER-PARAMETER TUNING

The "architecture" of the MLP network (i.e., the dimension Q of the latent space and the number and size of inner layers) and other hyper-parameters (such as the regularization coefficient λ) are tuned using a "grid search" procedure. Figure 10 illustrates the calibrated parameters along with the training times required for each hyper-parameter combination. The *loss* shown in Figure 10 (left panel) represents the average reconstruction loss across all training features based on the *relative mean square error* (RMSE), defined as

$$\mathrm{RMSE}(\boldsymbol{x}) = \frac{\sum_k (\hat{x}_k - x_k)^2}{\sum_k x_k^2}.$$

The training times in Figure 10 (right) correspond to training a sample of 50,000 scanners from September 2nd, 2016. We

TABLE VII: Interpretation of clustering changes between September 23 and September 24, 2016. Notice the rows in gray scale that indicate the formation of a new large cluster (cluster 24), associated with a DDoS attack.

Day	Label	Mass	Jaccard	Size	Packets	Avg. IA (ms)	Bytes	# DstPorts	# DstAddr	TTL	Freq.	Traffic	Freq.	Ports	Freq.
23	13			22294	2890	29704	0	2	1282	45	22096	TCP-SYN	22208	23-2323	22099
24	63	0.025	0.14	37923	1196	53657	10	2	792	45	36858	TCP-SYN	37520	23-2323	37322
23	9			20659	1404	52025	89	2	1011	47	20038	TCP-SYN	20539	23-2323	20430
24	60	0.023	0.16	31479	914	61293	19	2	781	47	25513	TCP-SYN	31195	23-2323	29094
23	28			24152	851	52845	0	2	686	47	23893	TCP-SYN	24141	23-2323	19273
24	25	0.022	0.12	24422	648	58031	0	2	537	47	25423	TCP-SYN	29387	23-2323	21269
23	81			31276	1681	43937	32	2	1036	46	31086	TCP-SYN	31094	23-2323	31028
24	1	0.021	0.18	32827	1228	53974	21	2	881	46	32637	TCP-SYN	32536	23-2323	32437
23	11			23792	759	42032	0	2	663	53	23241	TCP-SYN	23787	23-2323	21545
24	29	0.021	0.11	28586	509	51834	0	2	444	53	28152	TCP-SYN	28583	23-2323	26336
23	47			19833	1477	56862	5	2	1090	48	17331	TCP-SYN	19702	23-2323	19592
24	24	0.017	0.00	23594	145	434328	5971	2	145	48	22803	ICMP (type 3)	23146	0	23204

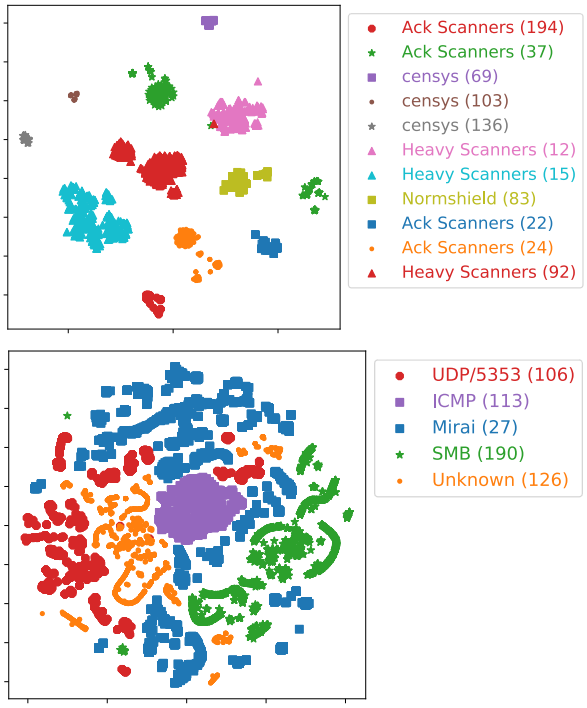

Fig. 9: t-SNE visualizations for various clusters.

.

trained the MLP network on a Tesla P100 NVIDIA GPU with 16GB of memory. We show results for 50 epochs of training and a batch size of 2000. As we observe, the best architecture with regards to loss is one using 3 layers for the encoding function; an input layer for the features, a hidden layer of size 1000 and a third layer for the embeddings output of size $Q = 50$. This architecture balances well computational efficiency with autoencoder accuracy and has been used throughout our experiments.

The "elbow method" [70] was used to select a judicious value for K, a critical parameter for K-means clustering. Figure 11 shows the Jaccard and Silhouette measures as a function of size K. Two different datasets (see Table I) are utilized to calculate the Jaccard score, which is based on around a dozen "semi-ground-truth" scanner labels we assigned to some of the Darknet senders. For instance, the Mirai signature [3] was used to label some scanners as "Mirai", scanners targeting port TCP/445 were labeled as "SMB", etc. By inspecting these plots, we decided to set $K = 200$ in all of our experiments.

Appendix B
Synthetic Data Generation

We have opted to use a generative model based on Bayesian networks to generate synthetic data that capture the causal relationships between the numerical features we employ in our study (i.e., the ones in Sec. IV-B under the "Traffic volume" and "Scan strategy" categories, except the *destination strategy*, *IPID strategy* and *IPID options*). To learn the Bayesian network we employ the *hill-climbing* algorithm implemented in R's `bnlearn` package [71]. We use features from a typical day of our Darknet to learn the structure of the network, which is represented as a *directed acyclic graph* (DAG). The nodes in the DAG represent the features and the edges between pairs of nodes represent the causal relationship between these nodes.

Let $X_1, \ldots, X_n$ denote the nodes of the Bayes network. Their joint distribution can be expressed as $\mathbb{P}(x_1, \ldots, x_n) = \prod_{i=1}^{n} \mathbb{P}(x_i|\text{parents}(X_i))$, where $\text{parents}(X_i)$ denote the parents of node X_i that appear in the DAG. It can be shown that for every variable in the network X_i, we can have

$$\mathbb{P}(X_i|X_{i-1}, \ldots, X_1) = \mathbb{P}(X_i|\text{parents}(X_i)).$$

This relationship can be satisfied if the nodes in the Bayes net are numbered in a *topological order* [72]. Given this specification of the joint distribution, we then proceed with a Monte Carlo randomized sampling algorithm to obtain data points for our synthetic dataset (see [72], Sec. 13.4).

In the Monte Carlo approach, we treat all variables $X_1, \ldots, X_n$ as Gaussian random variables with a joint distribution $\mathcal{N}(\mu, \Sigma)$, and hence we employ the conditional distribution relationships for multivariate Gaussian random variables. The parameters μ and Σ are estimated from the same real Darknet dataset we use to learn the Bayes net.

Once the numerical features are generated with the Monte Carlo approach, we add the feature "set of ports scanned" so that each synthetically generated data point combines both numerical and categorical features. We create K distinct clusters by appropriately spacing the values of the root nodes in the Bayes network.

References

[1] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Network telescopes: Technical report," Cooperative Association for Internet Data Analysis (CAIDA), Tech. Rep., Jul 2004.

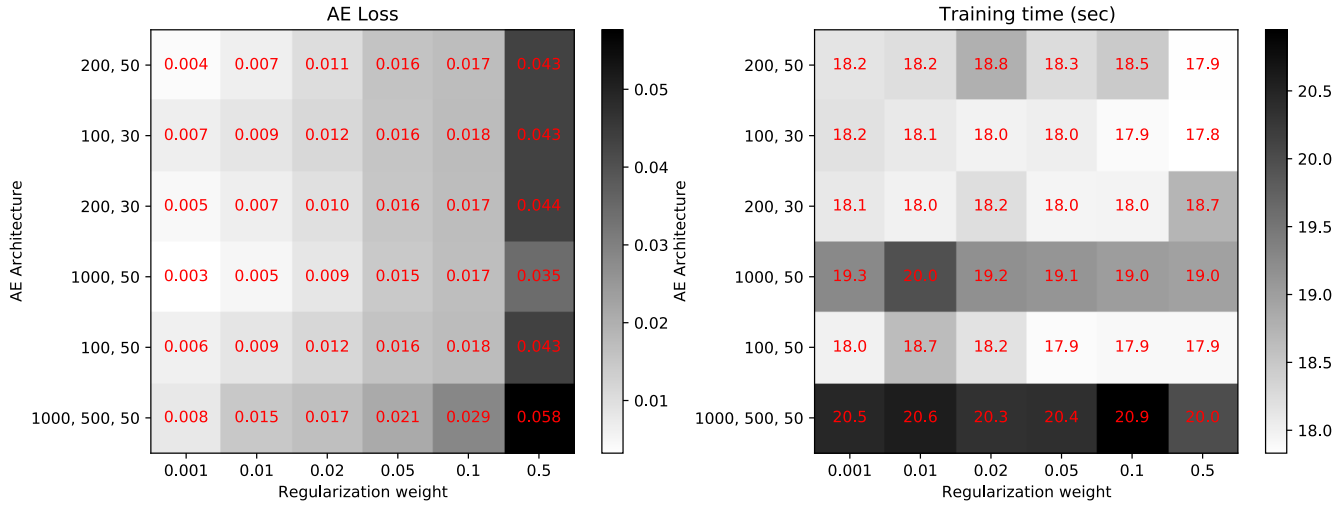

Fig. 10: Autoencoder tuning: Grid search for MLP architecture and regularization weight (λ). (Left) Relative MSE; (Right) Training times. Note that the notation "200, 50" denotes a 3-layer network architecture with an input layer of size P, a hidden layer of size 200 and the output layer of "embeddings" of size 50. To avoid cluttering, we omit denoting the size P of the input layer in these heatmaps.

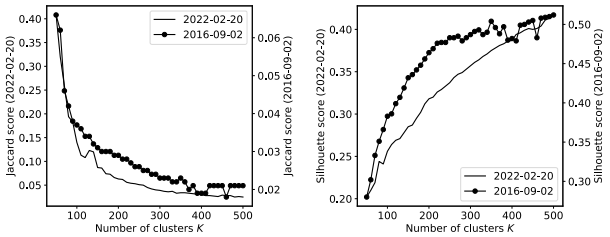

Fig. 11: Elbow plots: Jaccard and Silhouette measures.

[2] Merit Network, Inc., "ORION: Observatory for Cyber-Risk Insights and Outages of Networks," https://www.merit.edu/initiatives/orion-network-telescope/, 2022.

[3] 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. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium*. USENIX Association, 2017.

[4] Z. Durumeric, M. Bailey, and J. A. Halderman, "An internet-wide view of internet-wide scanning," in *Proceedings of the 23rd USENIX Conference on Security Symposium*, ser. SEC'14. Berkeley, CA, USA: USENIX Association, 2014, pp. 65–78.

[5] P. Richter and A. Berger, "Scanning the scanners: Sensing the internet from a massively distributed network telescope," in *Proceedings of the Internet Measurement Conference*, ser. IMC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 144–157.

[6] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston, "Internet background radiation revisited," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, 2010, pp. 62–74.

[7] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of internet background radiation," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 2004, pp. 27–40.

[8] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[9] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.

[10] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, 2016.

[11] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE Signal Processing Magazine*, vol. 34, 2017.

[12] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. IEEE Computer Society, 1998, pp. 59–66.

[13] M. Jonker, A. King, J. Krupp, C. Rossow, A. Sperotto, and A. Dainotti, "Millions of targets under attack: A macroscopic characterization of the dos ecosystem," in *Proceedings of the 2017 Internet Measurement Conference*, ser. IMC '17. New York, NY, USA: ACM, 2017.

[14] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescapé, "Analysis of a "/0" stealth scan from a botnet," *IEEE/ACM Transactions on Networking*, vol. 23, no. 2, pp. 341–354, 2014.

[15] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer iot botnet." in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2019.

[16] O. Cabana, A. M. Youssef, M. Debbabi, B. Lebel, M. Kassouf, and B. L. Agba, "Detecting, fingerprinting and tracking reconnaissance campaigns targeting industrial control systems," in *Detection of Intrusions and Malware, and Vulnerability Assessment*. Cham: Springer International Publishing, 2019, pp. 89–108.

[17] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 pound gorilla: The rise and decline of ntp ddos attacks," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, 2014, pp. 435–448.

[18] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Inferring distributed reflection denial of service attacks from darknet," *Computer Communications*, vol. 62, pp. 59–71, 2015.

[19] A. Wang, W. Chang, S. Chen, and A. Mohaisen, "Delving into internet ddos attacks by botnets: characterization and analysis," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2843–2855, 2018.

[20] M. Jonker, A. Pras, A. Dainotti, and A. Sperotto, "A first joint look at DoS attacks and BGP blackholing in the wild," in *Proceedings of the Internet Measurement Conference 2018*, 2018, pp. 457–463.

[21] D. Moore, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," in *USENIX Security Symposium*, Washington, D.C., Aug 2001.

[22] J. Czyz, K. Lady, S. G. Miller, M. Bailey, M. Kallitsis, and M. Karir, "Understanding IPv6 internet background radiation," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013.

[23] T. Ban, L. Zhu, J. Shimamura, S. Pang, D. Inoue, and K. Nakao, "Behavior analysis of long-term cyber attacks in the darknet," in *International Conference on Neural Information Processing*, 2012.

[24] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescapé, "Analysis of a /0; stealth scan from a botnet," *IEEE/ACM Transactions on Networking*, vol. 23, no. 2, pp. 341–354, April 2015.

[25] A. Guillot, R. Fontugne, P. Winter, P. Merindol, A. King, A. Dainotti, and C. Pelsser, "Chocolatine: Outage detection for internet background radiation," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*, 2019, pp. 1–8.

[26] K. Benson, A. Dainotti, K. Claffy, and E. Aben, "Gaining insight into as-level outages through analysis of internet background radiation," in *2013 IEEE Conference on Computer Communications Workshops*, 2013.

[27] A. Dainotti, R. Amman, E. Aben, and K. C. Claffy, "Extracting benefit from harm: Using malware pollution to analyze the impact of political and geophysical events on the internet," *SIGCOMM CCR*, vol. 42, no. 1, pp. 31–39, Jan. 2012.

[28] S. Torabi, E. Bou-Harb, C. Assi, M. Galluscio, A. Boukhtouta, and M. Debbabi, "Inferring, characterizing, and investigating internet-scale malicious iot device activities: A network telescope perspective," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, pp. 562–573.

[29] S. Ozawa, T. Ban, N. Hashimoto, J. Nakazato, and J. Shimamura, "A study of iot malware activities using association rule learning for darknet sensor data," *International Journal of Information Security*, vol. 19, no. 1, pp. 83–92, 2020.

[30] F. Shaikh, E. Bou-Harb, J. Crichigno, and N. Ghani, "A machine learning model for classifying unsolicited iot devices by observing network telescopes," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 938–943.

[31] H. Nishikaze, S. Ozawa, J. Kitazono, T. Ban, J. Nakazato, and J. Shimamura, "Large-scale monitoring for cyber attacks by using cluster information on darknet traffic features," *Procedia Computer Science*, vol. 53, pp. 175–182, 2015.

[32] T. Ban, S. Pang, M. Eto, D. Inoue, K. Nakao, and R. Huang, "Towards early detection of novel attack patterns through the lens of a large-scale darknet," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing*. IEEE, 2016, pp. 341–349.

[33] F. Iglesias and T. Zseby, "Pattern discovery in internet background radiation," *IEEE Transactions on Big Data*, 2017.

[34] A. Sarabi and M. Liu, "Characterizing the internet host population using deep learning: A universal and lightweight numerical embedding," in *Proceedings of the Internet Measurement Conference 2018*, 2018.

[35] The Censys Team, "Censys.io," https://censys.io.

[36] L. Gioacchini, L. Vassio, M. Mellia, I. Drago, Z. B. Houidi, and D. Rossi, "Darkvec: Automatic analysis of darknet traffic with word embeddings," in *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '21. New York, NY, USA: ACM, 2021, p. 76–89.

[37] 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 *Computer Security – ESORICS 2020: 25th European Symposium on Research in Computer Security*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 88–109.

[38] M. Ring, A. Dallmann, D. Landes, and A. Hotho, "Ip2vec: Learning similarities between ip addresses," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 657–666.

[39] N. Provos *et al.*, "A virtual honeypot framework." in *USENIX Security Symposium*, vol. 173, no. 2004, 2004, pp. 1–14.

[40] P. Barford, Y. Chen, A. Goyal, Z. Li, V. Paxson, and V. Yegneswaran, *Employing Honeynets For Network Situational Awareness*. Boston, MA: Springer US, 2010, pp. 71–102.

[41] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow, "Amppot: Monitoring and defending against amplification ddos attacks," in *International Symposium on Recent Advances in Intrusion Detection*. Springer, 2015, pp. 615–636.

[42] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, *Open for Hire: Attack Trends and Misconfiguration Pitfalls of IoT Devices*. New York, NY, USA: Association for Computing Machinery, 2021, p. 195–215.

[43] M. Wang, J. Santillan, and F. Kuipers, "ThingPot: an interactive Internet-of-Things honeypot," *arXiv e-prints*, p. arXiv:1807.04114, Jul. 2018.

[44] P. Simões, T. Cruz, J. Proença, and E. Monteiro, "Specialized honeypots for scada systems," in *Cyber Security: Analytics, Technology and Automation*. Springer, 2015, pp. 251–269.

[45] R. Hiesgen, M. Nawrocki, A. King, A. Dainotti, T. Schmidt, and M. Wählisch, "Spoki: Unveiling a new wave of scanners through a reactive network telescope," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022.

[46] CAIDA, "Routeviews Prefix to AS mappings Dataset (pfx2as) for IPv4 and IPv6," https://www.caida.org/catalog/datasets/routeviews-prefix2as/.

[47] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *USENIX SEC'13*, 2013, pp. 605–620.

[48] K. Benson, A. Dainotti, k. claffy, A. C. Snoeren, and M. Kallitsis, "Leveraging internet background radiation for opportunistic network analysis," in *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, ser. IMC '15, 2015.

[49] S. Siby, "Default TTL (Time To Live) Values of Different OS, 2014," https://subinsb.com/default-device-ttl-values.

[50] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3861–3870.

[51] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[52] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science NY*, vol. 313, pp. 504–7, 08 2006.

[53] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 3861–3870.

[54] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv e-prints*, p. arXiv:1312.6114, Dec. 2013.

[55] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[58] A. Ramdas, N. G. Trillos, and M. Cuturi, "On wasserstein two-sample testing and related families of nonparametric tests," *Entropy*, vol. 19, no. 2, 2017.

[59] L. Gioacchini, L. Vassio, M. Mellia, I. Drago, Z. B. Houidi, and D. Rossi, "Darkvec: Automatic analysis of darknet traffic with word embeddings (code and data)," https://github.com/SmartData-Polito/darkvec.

[60] Shodan, "Shodan Search Engine," www.shodan.io.

[61] M. Collins, "Acknowledged Scanners (Version 1.0 – September 23, 2021)," https://gitlab.com/mcollins_at_isi/acknowledged_scanners.

[62] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*, February 2014.

[63] O. Alrawi, C. Lever, K. Valakuzhy, R. Court, K. Snow, F. Monrose, and M. Antonakakis, "The circle of life: A Large-Scale study of the IoT malware lifecycle," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 3505–3522.

[64] C. Young, R. McArdle, N.-A. Le-Khac, and K.-K. R. Choo, *Forensic Investigation of Ransomware Activities—Part 1*. Cham: Springer International Publishing, 2020, pp. 51–77.

[65] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, "Ransomware detection and mitigation using software-defined networking: The case of wannacry," *Computers & Electrical Engineering*, vol. 76, pp. 111–121, 2019.

[66] tchelebi, "Normshield," https://tchelebi.io/.

[67] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[68] NETSCOUT, "Arbor Networks Research Scanner," https://www.arbor-observatory.com/.

[69] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[70] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, Dec. 2001.

[71] S. Højsgaard, D. Edwards, and S. Lauritzen, *Graphical Models with R*. USA: Springer, Boston, MA, 2012.

[72] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 4th ed.* USA: Pearson, 2020.