DarkSim: A Similarity-Based Time Series Analytic Framework for Darknet Traffic

Max Gao
UC San Diego
magao@ucsd.edu

Ricky Mok
CAIDA/UC San Diego
cskpmok@caida.org

Esteban Carisimo
Northwestern University
esteban.carisimo@northwestern.edu

kc claffy
CAIDA/UC San Diego
kc@caida.org

Eric Li
UC San Diego
jul108@ucsd.edu

Shubham Kulkarni
UC San Diego
skulkarn@ucsd.edu

ABSTRACT

Network Telescopes, often referred to as *darknets*, capture unsolicited traffic directed toward advertised but unused IP spaces, enabling researchers and operators to monitor malicious, Internet-wide network phenomena such as vulnerability scanning, botnet propagation, and DoS backscatter. Detecting these events, however, has become increasingly challenging due to the growing traffic volumes that telescopes receive. To address this, we introduce *DarkSim*, a novel analytic framework that utilizes Dynamic Time Warping to measure similarities within the high-dimensional time series of network traffic. *DarkSim* combines traditional raw packet processing with statistical approaches, identifying traffic anomalies and enabling rapid *time-to-insight*. We evaluate our framework against *DarkGLASSO*, an existing method based on the Graphical LASSO algorithm, using data from the UCSD Network Telescope. Based on our manually classified detections, *DarkSim* showcased perfect precision and an overlap of up to 91% of *DarkGLASSO*'s detections in contrast to *DarkGLASSO*'s maximum of 73.3% precision and detection overlap of 37.5% with the former. We further demonstrate *DarkSim*'s capability to detect two real-world events in our case studies: (1) an increase in scanning activities surrounding CVE public disclosures, and (2) shifts in country- and network-level scanning patterns that indicate aggressive scanning. *DarkSim* provides a detailed and interpretable analysis framework for time-series anomalies, representing a new contribution to network security analytics.

CCS CONCEPTS

• **Networks** → **Network measurement**; **Network security**.

KEYWORDS

Network telescope; Internet scanning; network traffic analysis

IMC '24, November 4–6, 2024, Madrid, Spain.

ACM ISBN 979-8-4007-0592-2/24/11…$15.00
https://doi.org/10.1145/3646547.3688426

ACM Reference Format:
Max Gao, Ricky Mok, Esteban Carisimo, kc claffy, Eric Li, and Shubham Kulkarni. 2024. DarkSim: A Similarity-Based Time Series Analytic Framework for Darknet Traffic. In *Proceedings of the 2024 ACM Internet Measurement Conference (IMC '24), November 4–6, 2024, Madrid, Spain.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3646547.3688426

1 INTRODUCTION

Internet Background Radiation (IBR) consists of unsolicited network traffic emitted by globally-distributed devices connected over the Internet. Unused address spaces occupied by network telescopes, often referred to as *darknets*, capture this traffic and thus serve as vital observatories for monitoring its network activities which include malicious scanning campaigns [20], denial-of-service (DoS) attacks [43, 44, 58, 73], and outages [21, 61]. While these activities account for the majority of changes in IBR traffic, their prompt detection and accurate identification prove challenging though critical due to their security implications.

Rapid evolution in the network threat landscape, driven by widespread use of rapid scanning tools (*e.g.*,[24, 29]) and ever-evolving attack vectors, has increased the volume and complexity of IBR which in turn challenges the effectiveness of traditional detection methods. While signature-based packet filtering approaches are capable of processing large traffic volumes in real-time, the fixed nature of their detection mechanism does not proactively adapt to unseen traffic events. Classical statistical methods, such as change point detection [1, 2, 39], often make statistical assumptions about traffic characteristics that, when incorrect, translate to poor detection accuracy. Although recent applications of representation learning techniques [28, 45] show empirical promise, they require substantial computational resources and expertise to implement, and their resulting models often lack interpretability to non-experts.

Motivated by these challenges, we develop *DarkSim*, a novel analytics framework designed to detect both *unseen* and *recurring* traffic anomalies. Utilizing the Dynamic Time Warping (DTW) [7, 10, 23, 62] in its core, *DarkSim* analyzes IBR telemetry data from network telescopes to identify unusual traffic patterns that merit further investigation at a significantly lower "time to insight" compared to traditional methods reliant on high-resolution data such as packet traces and flows. We design our framework based on these three principles:

(1) *No model assumptions.* It does not rely on *a priori* knowledge and makes minimal statistical assumptions about network traffic characteristics.

(2) *Computationally scalable.* The computation time of our analysis is relatively constant because DTW is independent of the volume of traffic and the complexity of the time series segments. Moreover, our implementation leverages the parallelism of high-performance computing clusters for DTW score computation, thereby reducing the time required to detect anomalies.
(3) *Explainable.* Its results are easily interpreted as we can identify key characteristics in time series segments that explain similarity scores. Identification of anomalous segments' timeframes and traffic properties facilitates further investigation of traffic events.

In this paper, we detail our design of *DarkSim* (§3), extended from our prior work [27], and its implementation which leverages open-source tools, such as DTAIDistance [52], to compute DTW scores, and integrates parallel computing technologies, such as Dask [22], to enable rapid processing of time series. We demonstrate *DarkSim*'s capabilities by analyzing time series data of IBR traffic statistics collected from the UCSD Network Telescope (UCSD-NT) [15], the world's largest darknet encompassing approximately 12 million IPv4 addresses, with our implementation deployed to Expanse [71], a high-performance computing (HPC) cluster at the San Diego Supercomputing Center. Our evaluation benchmarks *DarkSim*'s detection capabilities against *DarkGLASSO* [31, 32], a recent framework based on the Graphical LASSO (GLASSO) algorithm [25] (§4). We further evaluate *DarkSim*'s practical application in two case studies, detecting (1) an increase in scanning activities surrounding CVE public disclosures (§5.1); and (2) shifts in country- and network-level scanning patterns that suggest aggressive scanning (§5.2).

To summarize, our contributions are as follows:

- We implement *DarkSim* using open-source tools for offline batch processing of IBR time series in parallel computation environments. We release its source code[1] and experimental artifacts produced in this work.
- We benchmark *DarkSim*'s detection capabilities against *DarkGLASSO* [31, 32], a recent framework based on the GLASSO algorithm [25]. Our framework achieved perfection precision and a maximum overlap of 91% with *DarkGLASSO*'s detections, in contrast to *DarkGLASSO*'s maximum of 73.3% precision and 37.5% overlap.
- We demonstrate *DarkSim*'s practical utility in two case studies:
  - Our first case study detected signatures of scanning activities that targeted TCP ports of applications reported to contain Remote Code Execution (RCE) vulnerabilities, identified in Microsoft's February 2023 Patch Tuesday. Further analysis revealed the sources of these activities as 9 Autonomous Systems (ASes) based in China. In addition, we used the discovered anomalous signatures to trace historical incidents from 2022 to 2023, matching on over 1,600 ports, including those related to recent vulnerabilities in Kubernetes.
  - Our second case study detected anomalies in country-level packet count time series. We identified scanning campaigns originating from a networks geolocated to the Netherlands and US, which corresponded with significant drops in traffic from over hundreds of other countries.

[1] https://github.com/CAIDA/DarkSim

Table 1: Trade-offs between resolution and data sizes of file formats collected by UCSD-NT. The order of magnitude in file size reductions depends on the number of unique time-series analyzed.

File Format	Daily Avg. File Size (2022-08)
Packet Capture (.pcap.gz)	3.86 TB
One-way Flow (.avro)	0.076 TB
2$\hat{1}$6 time series (.parquet.gzip)	0.001 TB

Traffic Resolution ↓ / Data Size ↑

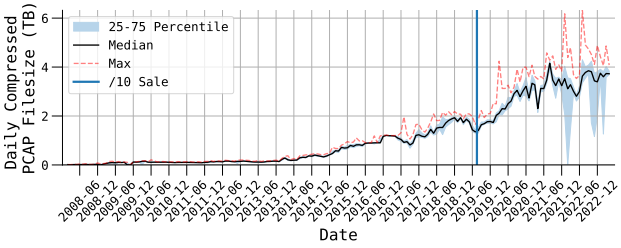


Figure 1: Traffic volume growth over the past 20 years challenges the timely analysis of darknet traffic.

2 BACKGROUND

2.1 Challenges to IBR event detection

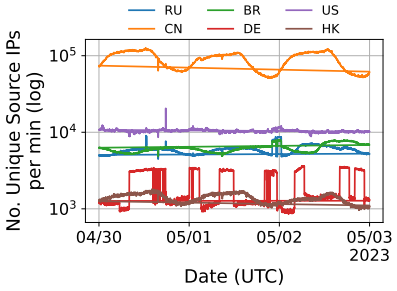
IBR's growth poses practical challenges to its timely analysis. Over the past two decades, IBR volumes captured at UCSD-NT, the world's largest network telescope, have grown by more than three orders of magnitude to a daily median filesize of over 3.5 terabytes (Figure 1) despite a 25% reduction in address space in 2019, pacing the growth of estimated global traffic [41].

Flow aggregation techniques similar to those used for traffic in production networks, *e.g.*, Netflow [35], have been developed for unidirectional darknet traffic. Custom representations such as FlowTuple [12] aggregate 5-minutes of raw packets of IBR into compressed file formats, *e.g.*, Avro [6]. To analyze events using this intermediary data representation, researchers must still ingest large volumes, albeit up to two orders of magnitude less than raw packets (Table 1), and know the precise times an event of interest occurs. On the other hand, event-specific representations, *e.g.*, Merit's ORION project [65] and CAIDA's RSDoS [13], filter traffic that matches event definitions based on inference heuristics, such as Masscan [29] packet fingerprints and TCP headers inferred to be randomly-spoofed denial-of-service [58]. However, these fixed heuristics do not flexibly adapt to other (including new) activities observed in IBR.

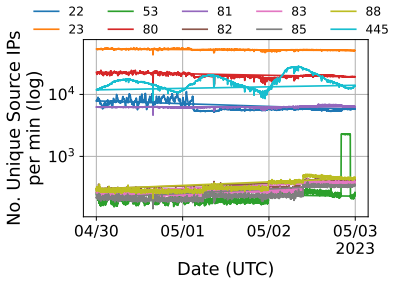
These challenges motivate us to develop a generalized framework capable of detecting a wide range of activities without the

Table 2: Traffic properties, metrics, and filters whose combinations produce over 4.9M (327K × 5 × 3) time series as potential inputs to *DarkSim*.

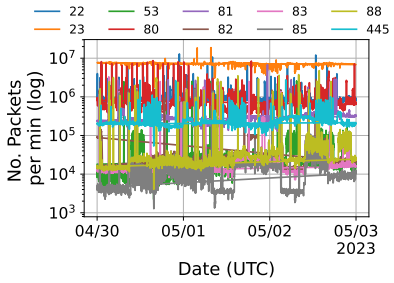
Traffic Properties		Metrics	Filters
Property Class	Unique Count		
Origin ASN	130K+	# packets (PPM)	Unfiltered
Geolocation	255	# bytes (BPM)	Inferred Non-Spoofed
Protocol Number	256	# unique source IPs	Inferred Spoofed
TCP/UDP Dest. Port	131072	# unique source ASNs	
ICMP Type & Code	65536	# unique dest. IPs	



(a) Unique source IPs per minute for 6 countries, 4 of which show stable diurnal patterns of different amplitudes and phases. Germany (DE) and US instead show irregular square/horizontal patterns.



(b) Unique source IPs per minute for 10 well-known TCP destination ports. Some ports (*e.g.*, 23 and 80) show similar patterns with synchronized changes. Some events impact only one port (*e.g.*, 53 on May 2).



(c) Packets per minute for 10 well-known TCP destination ports. This metric exhibits noisier characteristics than that of Fig. 2b for the same ports.

Figure 2: Time series of traffic captured by UCSD-NT (April 30 - May 3, 2023) whose characteristics (*e.g.*, magnitude, pattern, and periodicity) vary across different traffic properties and metrics.

processing costs of analysis using higher resolution formats, *i.e.*, raw packet or flow representations.

2.2 UCSD-NT Time Series Data

UCSD-NT computes metrics per set of traffic properties (Table 2) to provide a glimpse into IBR dynamics in near real-time. In addition to the traditional network 5-tuple (src/dst IPs, src/dst ports, and protocol), properties include inference of spoofed source address [19] and labels from other datasets, such as ASNs and geolocation. UCSD-NT stores the time series data into InfluxDB [38], visible through publicly accessible Grafana dashboards [14].

2.3 Dynamics of IBR in UCSD-NT

By applying the properties to traffic metrics listed in Table 2, we derive over 200k unique time series from raw IBR packets. Figure 2 shows several such time series selected from a week in May 2023, highlighting differences in scale, characteristics (*e.g.*, wave-like, square-like, and step-wise), variance, and periodicity. To monitor and detect events in IBR, we desire a general approach that accounts for this spatiotemporal variety and reliably discerns anomalies. We identify three properties that such an approach may leverage:

(1) *High similarity within the same time series.* Recurring scanning activity of malware-infected end-hosts produces diurnality in some metrics (*e.g.*, number of unique source IPs). Deviations from these known patterns indicate potential new events.
(2) *Malicious activities trigger synchronized changes.* Botnet commands may induce simultaneous and overlapping scanning campaigns to exploit new vulnerabilities. Synchronized anomalies across different traffic sources can indicate these activities.
(3) *Time series patterns may reveal the use of probing tools/logic.* Algorithms used by measurement tools select packet-sending rates and destinations, yielding distinct traffic patterns. Comparing time series with known patterns can reveal the nature of events.

2.4 Measuring time series similarity

Algorithms that measure time series similarity differ in their trade-offs between semantic accuracy, time complexity, and simplicity of parameterization. While straightforward measures such as L_p-norms, *e.g.*, the Manhattan Distance (L_1), and correlation/covariance, are inexpensive to compute and require no additional parameters, their accuracy drops for noisy and lagged time series. More recent algorithms [56] yield higher benchmarked accuracy and performance but require tuning numerous parameters. We seek an algorithm that produces semantically accurate measures, has low time complexity, and requires few or no parameters.

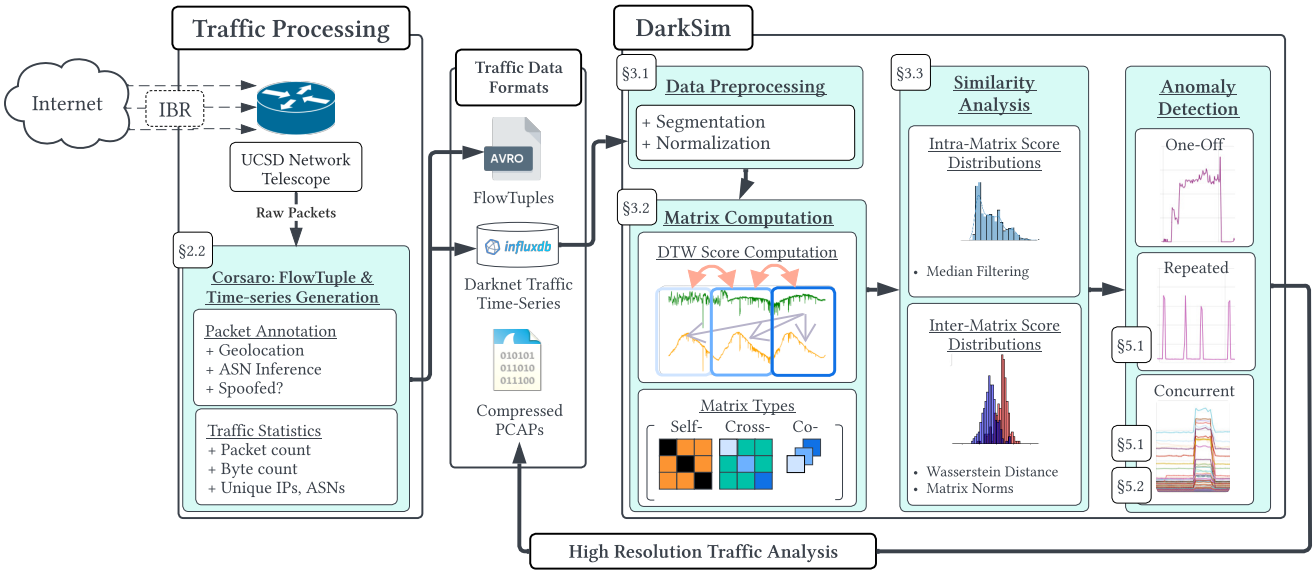


Figure 3: *DarkSim* System Architecture.

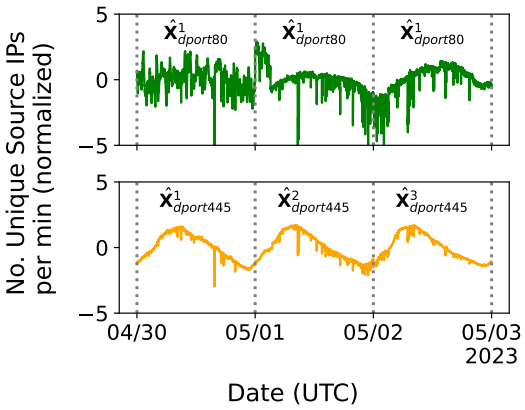


Figure 4: Unique source IPs per minute for TCP destination ports 80 and 445 after pre-processing their raw time series presented in Fig. 2b.

2.4.1 Dynamic Time Warping. We apply the classic DTW algorithm, a popular choice for analysis across a wide range of domains [50, 51, 78], to darknet traffic time series. By temporally aligning observations to account for potential time-shifts across two time series, DTW produces *semi-metric*[2] scores that accurately quantify pattern similarities. This attribute is fundamental for IBR signal analysis, especially in cases of coordinated attacks or scanning where suspicious source behavior is unlikely to be perfectly synchronized. Moreover, its computational complexity is controlled by a single parameter, the warp-width [68], which adjusts the maximum temporal alignment width as a fraction of the time series length. Appendix B.1 provides in-depth details of DTW.

3 METHODOLOGY

DarkSim consists of three main steps: (1) preprocessing raw time series data (§3.1), (2) computing similarity scores between time series segments (§3.2), (3) identifying anomalous segment patterns from statistical properties of score distributions (§3.3). In the following paragraphs, we explain the details of each step (summarized in Figure 3) using notation from Table 3.

3.1 Data preparation

DarkSim preprocesses raw time series inputs by segmenting and normalizing their observations. Separated according to their traffic properties $\mathbf{P}$ (*e.g.*, source port, destination port, etc.), each time series X_p^T consists of T observations sampled at a rate f, with X_p^1 and X_p^T representing an analysis interval's first and last sampling windows, respectively.

From a single time series, the segmentation step uses a prespecified segment length parameter b to generate N equal-length segments, where $N = \lceil T/b \rceil$. For $|\mathbf{P}|$ unique time series, this step produces a total of $|\mathbf{P}|N$ segments. We provide a sensitivty analysis of b in §4. *DarkSim* then applies Z-score normalization[3] on a per-segment basis to address scaling issues in the scoring process. We denote these processed segments as $\hat{\mathbf{X}}_p^n$, with n ranging from 1 to N. To illustrate this step, we utilize two time series from Figure 2 representing the unique counts of IP source addresses to TCP destination ports 80 and 445. Figure 4 presents the results of

[2]DTW's scores do not formally abide by the triangle inequality, though empirically they seldom violate it [59], which may impact the integrity of results for algorithms that depend on *metric* inputs.

[3]Z-score normalization transforms data to a standard scale with a mean of zero and a standard deviation of one.

the process of dividing these time series into daily segments and applying Z-score normalization.

Table 3: Summary of notation.

Symbol	Definition
$\mathbf{P}$	a selected set of traffic properties
f	sampling rate
T	Length of analysis interval (total number of observations)
N	number of time series segments
b	Segment length (observations per segment)
$\mathbf{X_p^T}$	raw time series observations
$\hat{\mathbf{X}}_p^n$	n^{th} normalized segment derived from time series of traffic property p
w	DTW warp-width (as a fraction of b)
$\mathcal{D}(\hat{\mathbf{X}}_p^i, \hat{\mathbf{X}}_p^j, w)$	DTW function computed over segments $\hat{\mathbf{X}}_p^i$,$\hat{\mathbf{X}}_p^j$ with warp-width w

3.2 Computing *DarkSim* matrices

After preprocessing segments, the analytic core of *DarkSim* compares them using DTW. For each pair of segments from $|\mathbf{P}|$ unique time series, *DarkSim* computes a dissimilarity score (a zero score indicates identical segments[4]) and arranges scores into a **Full-Dissimilarity Matrix (Full-DM)**. This matrix ($\mathbf{M}$ in Equation 1) is further partitioned into submatrices (m_{p_a,p_b} in Equation 2) that contain scores between all of the segments belonging to two time series ($\mathbf{X_{p_a}^T}$, $\mathbf{X_{p_b}^T}$).

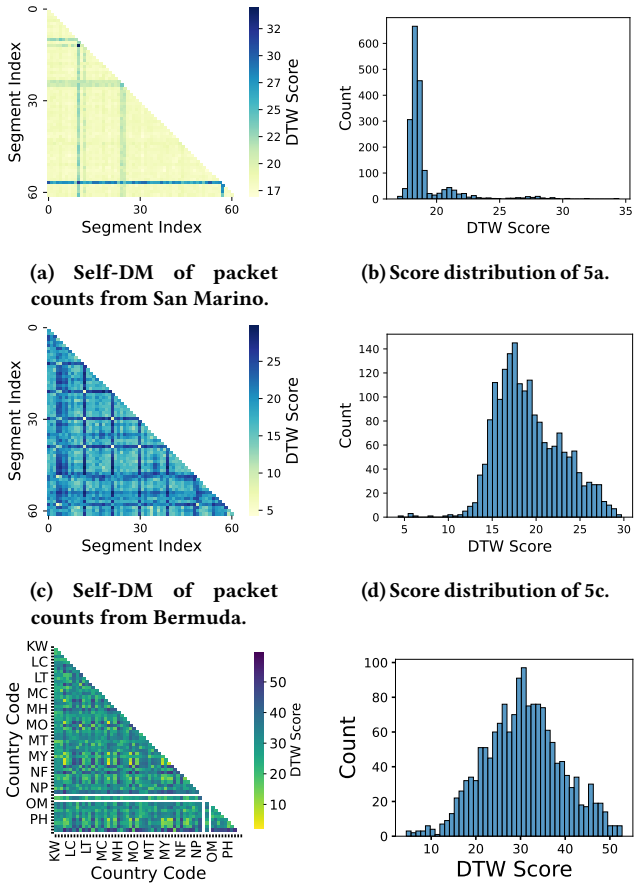
$$\mathbf{M} = \begin{bmatrix} m_{p_1,p_1} & \dots & m_{p_1,p_{|\mathbf{P}|}} \\ \vdots & \ddots & \vdots \\ m_{p_{|\mathbf{P}|},p_1} & \dots & m_{p_{|\mathbf{P}|},p_{|\mathbf{P}|}} \end{bmatrix} \tag{1}$$

$$m_{p_a,p_b} = \begin{bmatrix} \mathcal{D}(\mathbf{X}_{p_a}^1, \mathbf{X}_{p_b}^1, w) & \dots & \mathcal{D}(\mathbf{X}_{p_a}^1, \mathbf{X}_{p_b}^N, w) \\ \vdots & \ddots & \vdots \\ \mathcal{D}(\mathbf{X}_{p_a}^N, \mathbf{X}_{p_b}^1, w) & \dots & \mathcal{D}(\mathbf{X}_{p_a}^N, \mathbf{X}_{p_b}^N, w) \end{bmatrix} \tag{2}$$

We explain the significance of different types of submatrices in the following lines.

- **Self-Dissimilarity Submatrix (Self-DM):** $\mathbf{M}$'s diagonal submatrices (Equation 1, highlighted in pink) contain comparisons between segments of the same traffic property.
- **Cross-Dissimilarity Submatrix (Cross-DM):** $\mathbf{M}$'s non-diagonal submatrices (Equation 1, highlighted in blue) contain comparisons between segments of different traffic properties.
- **Co-Dissimilarity Submatrix (Co-DM):** Assembled by selecting the i-th diagonal score across all cross-DMs, these submatrices contain comparisons between contemporaneous segments from $|P|$ unique time series.
- **Search Matrix:** Consists of scores that compare one segment to many segments across time and traffic properties.

In practice, we compute only a portion of matrix scores. In self-DMs, we skip comparisons between a segment and itself.[5] As DTW is commutative[6] and submatrices m are symmetric, we only compute lower-half scores. We present only these scores in the remainder of this paper for visual conciseness.



(a) Self-DM of packet counts from San Marino.

(b) Score distribution of 5a.

(c) Self-DM of packet counts from Bermuda.

(d) Score distribution of 5c.

(e) Co-DM of 60 countries' packet counts.

(f) Score distribution of 5e.

Figure 5: Heatmaps of various submatrices and their corresponding score distributions that show the characteristics of one-off anomalies (5a, 5b), repeated anomalies (5c, 5d), and concurrent anomalies (5e, 5f).

[4] Though both are theoretically unbounded from above, the Euclidean Distance (ED) between two segments empirically bounds their DTW score.

[5] $\mathcal{D}(\mathbf{S}_{p_a}^i, \mathbf{S}_{p_a}^j, w) = 0, \forall i = j$

[6] $\mathcal{D}(\hat{\mathbf{X}}_{p_a}^i, \hat{\mathbf{X}}_{p_b}^j, w) = \mathcal{D}(\hat{\mathbf{X}}_{p_b}^j, \hat{\mathbf{X}}_{p_a}^i, w))$, $\mathbf{M}$

3.3 Detecting anomalies from *DarkSim* matrices

DarkSim leverages empirical distributions of dissimilarity scores from submatrices to identify various anomalies. The primary types of anomalies detected include (1) One-off Anomalies, (2) Repeated Anomalies, and (3) Concurrent Anomalies, as illustrated in Figure 5. We detail specific characteristics of each type and demonstrate how they manifest within our score matrices.

(1) **One-off Anomalies** occur as singular, unusual events within the dataset and produce *high* scores due to their high dissimilarity with common segment patterns. Figure 5a depicts examples of this anomaly (blue continuous lines) in San Marino's Self-DM. These anomalies correspond to scores greater than 25 in Figure 5b.
(2) **Repeated Anomalies** occur multiple times across the dataset and produce low scores with irregular patterns among themselves. Figure 5c shows these anomalies (bright yellow cells) in Bermuda's Self-DM. These anomalies correspond to the tail of low-scores in Figure 5d.
(3) **Concurrent Anomalies** appear in contemporaneous segments from time series of different traffic properties and can surface as either *one-off* or *repeated* anomalies. Figure 5e shows an example of the latter case as spatially clustered low scores (dull yellow), representing rare pattern occurences at the same time across multiple data streams. Though not as distinct as Bermuda's, scores below 10 correspond to these anomalies in Figure 5f.

DarkSim applies median filtering to identify one-off, repeated, and concurrent anomalies in potentially non-normal score distributions. We use a conventional threshold of 3x the median absolute deviation (MAD) of a distribution to extract outlying scores, marking their segments for further investigation. In §5.1.3, we use a score of an expected pattern match as a threshold to detect repeated anomalies in search matrices. We adopt different approaches in our benchmarks with *DarkGLASSO* (§4).

To detect concurrent anomalies across time, *DarkSim* computes the Wasserstein Distance (WD) [46, 63, 77] between score distributions of two co-DMs. The WD provides an aggregated measure of dissimilarity between all pairwise DTW scores represented in these distributions, enabling identification of time periods corresponding to significant shifts in scores due to the appearance of anomalies. In §5.1 and §5.2, we identify outlying WDs using a specific type of median filter, the Hampel Filter [30], parameterized using a 3x MAD threshold over sliding windows of size 7.

4 EVALUATION

We evaluate *DarkSim*'s ability to detect scanning events under unsupervised settings. Our evaluation compares and contrasts *DarkSim* with recently published work, *DarkGLASSO* [31, 32], a recent framework based on the GLASSO algorithm. Our rationale behind benchmarking *DarkSim* against this framework is that both methodologies aim to identify similarities from the time series of IBR that indicate potential anomalous activity.

Table 4: Parameter values whose combinations we evaluated in our sensitivity analysis.

Parameter	Values
b	15, 15, 30, 60, 180, 720, 1440
λ	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
w	0, 0.5, 1.0

4.1 Implementation

To ensure a fair performance assessment, we align *DarkSim*'s implementation and our replication of *DarkGLASSO* to use a standard programming language and deploy their implementations in the same computational environment. We implement *DarkSim* using Cython bindings from DTAIDistance [52] to compute DTW scores from segments stored in memory. We re-implement *DarkGLASSO*, originally written in R [33], in Python using skggm [49]. skggm implements the QUIC algorithm [36], a more performant version of GLASSO. Our evaluation of both frameworks entail offline batch processing of time series on San Diego Supercomputer Center's Expanse [71] cluster. We leverage a single node on the cluster, utilizing 128 physical cores of its AMD EPYC 7742 processor to schedule application-level tasks. Although these frameworks do not strictly require parallelism, the high number of individual time series analyzed in our evaluation justifies the use of a high-performance computing (HPC) environment.

4.2 Evaluation setup

We perform a sensitivity analysis over both methods' parameters and compare their outputs to assess individual and relative detection efficacy. In this section, we provide details on the input dataset, outputs we compared, and each method's varied parameterizations.

Darknet time series dataset. We select 128 unique time series ($|\mathbf{P}| = 128$) from one month of darknet data collected between June 1 to June 30, 2023. Each time series consists of per-minute unique sender IP address counts observed for a single port among the 128 TCP destination ports with the highest packet counts over the month.

Method Outputs. Both methods produce matrices that require transformations to ensure their comparability. *DarkSim* computes *co-DMs* which we transform into co-similarity (co-SM) matrices by converting dissimilarity to similarity scores (see Appendix B.2) before comparing against *DarkGLASSO*'s outputs. Conversely, *DarkGLASSO* first computes *covariance* (Σ) and *correlation* (R) matrices. From these matrices, it then estimates *inverse covariance* (Σ^{-1}) and *inverse correlation* (R^{-1}) matrices that identify conditionally dependent segment pairs. Our analysis focuses exclusively on the negative elements ($\Sigma^{-1}_{-}$and R^{-1}_{-}) of inverse matrices as they reflect positive conditional dependence, akin to high similarity.

Method Parameters. We conduct a grid search over the parameters (Table 4) of *DarkSim* and *DarkGLASSO*. Both frameworks share the segment length parameter, b, which we evaluate using eight different lengths (denoted as the number of per-minute samples). For

DarkSim, we also examine the sensitivity of its results to the warp-width parameter (§2.4.1) by testing three values, $w \in \{0, 0.5, 1\}$ that cover the entire range of possible values $[0, 1]$. Conversely, *DarkGLASSO* applies a regularization penalty, λ, to empirical covariance Σ and correlation R matrices. We test nine different values of λ, consistent with those used by Han et al. [31, 32].

4.3 Comparing detection discriminability

To compare the detection discriminability of both frameworks, we assess the variability in L_1-norms of each method's output matrices. This variability implies a method's ability to broadly discriminate, absent of labels, between time periods presumably containing outliers versus typical cases.

We first execute both approaches over June's time series using combinations of parameters b, w, or λ, to obtain sets of matrices. For each matrix within a set, we calculate its L_1-norm, which summarizes all pairwise relationships for its corresponding time period. For a set of norms, we compute its Inter-Quartile Range (IQR), further normalized by its range to account for scale differences between both framework's matrices. Thus for a specific method and its choice of parameter values, a higher normalized IQR indicates stronger discriminability between windowed time periods. We note that the scope of our assessment does not assess variability within individual matrices.

DarkSim's results (Figure 6a) revealed two trends: (1) longer segments produced more variable L_1-norms; (2) maximal variability for most segment lengths occurred under a zero warp-width (*i.e.*, $w = 0$). The latter trend suggests that applying DTW's temporal alignment to short segments ($b < 1440$) can hamper discriminability. However for day-long segments specifically, a full-width alignment ($b = 1440$, $w = 1$) enabled *DarkSim* to outperform *DarkGLASSO*'s parameterizations.

DarkGLASSO's results (Figure 6b,6c) indicated an unclear relationship between parameters and variability. For Σ_-^{-1} (Figure 6b), notable variabilities occurred at lengths $b \in \{5, 15\}$ combined with weak penalizers $\lambda \in \{0.1\}$. For R_-^{-1} (Figure 6c), notable lengths were $b \in \{60, 360\}$ combined with mid-range $\lambda > 0.5$ penalizers. Sensitivity to temporal noise could explain decreased discriminability for weaker performing lengths.

Takeaway: Except at day-long segment lengths, *DarkGLASSO* matrices showcased overall higher variabilities than co-SMs, thus suggesting better discriminability. However, as we show in §4.4, segment pairs detected in these matrices are not necessarily accurate.

4.4 Comparing detection accuracy

To compare detection accuracy, we sample the most notable time series segment pairs detected by each method and inspect them for the presence of shared anomalous patterns. Using these detections, we assess *precision* and *relative overlap* for each framework. Though this does not provide a complete profile of detection accuracy, *i.e.*, we do not evaluate negative classes due to a lack of comprehensive ground truth, the two measures nonetheless reflect real-world framework performance on traffic data consisting of events unknown *a priori*.

4.4.1 Selecting candidate segment pairs. We select candidate segment pairs from *DarkSim* and *DarkGLASSO*'s matrices indicated as high-confidence detections. Per segment length, we identify values of the method-specific parameters that maximized score variability (parameters circled in red in Figure 6). For each matrix type, we select five matrices with the highest norms, totaling 120 candidate matrices (= 3 types × 8 segment lengths × 5). From each candidate matrix, we then identify indices of the top-3 highest valued elements and locate their pairs of original time series segments. This process results in 360 total (= 120 × 3) candidate segment pairs for inspection.

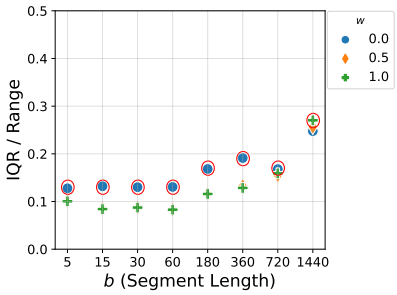
Table 5: Precision and relative overlap of *DarkSim* and *DarkGLASSO*'s detections (computed from Tab. 10, Tab. 11 of Appx. C). *DarkSim* achieved higher degrees of precision and overlap.

b	*DarkSim* Co-SM			*DarkGLASSO* Σ_-^{-1}		*DarkGLASSO* R_-^{-1}	
	Prec.	Overlap Σ_-^{-1}	Overlap R_-^{-1}	Prec.	Overlap Co-SM	Prec.	Overlap Co-SM
5	100	0	25	33.3	66.7	33.3	66.7
15	100	25	85.7	26.7	0	46.7	13.3
30	100	16.7	75	40	13.3	26.7	13.3
60	100	70	80	66.7	0	73.3	26.7
180	100	100	100	40	0	66.7	60
360	100	71.4	72.7	46.7	0	73.3	26.7
720	100	87.5	100	53.3	13.3	53.3	66.7
1440	100	50	66.7	40	0	60	53.3

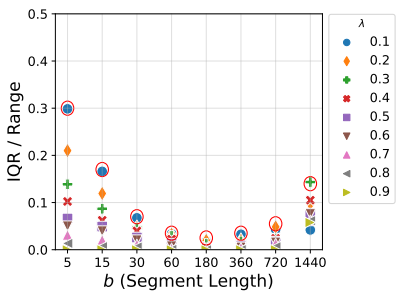
4.4.2 Assessing segment pairs for anomalies. We manually classify times series segment pairs based on their shared pattern characteristics (Figure 7) into three groups—Group A, B and C. Segments in Group A pairs share clear and distinct patterns/trends (Figure 9a). Group B segments likewise share patterns and trends, though of less obvious commonalities (Figure 9b). We count pairs in both of these groups as *true positives*. In contrast, Group C segments lack clear pattern/trend similarities (Figure 7c), which we count as *false positives*.

4.4.3 Comparing precision. From group assignment counts (Table 10 of Appendix C), we compute precision values (Table 5) per segment length for each type of matrix. *DarkSim*'s detections resulted in 100% precision across all segment lengths as we found distinctive shared patterns (Group A) in all segment pairs we examined. In contrast, *DarkGLASSO*'s detections for both types of inverse matrices resulted in overall lower precision (roughly 43.3% and 52.5% averaged across all segment lengths for $\Sigma^{-1}$and R^{-1}, respectively). We note that for both matrices, segment lengths less than an hour produced notably lower precision. We suspect that noisy observations likely affect the accuracy of covariance and correlation measures more for shorter lengths than for longer lengths, translating to lower true positive and higher false positive counts in their inverse matrices.

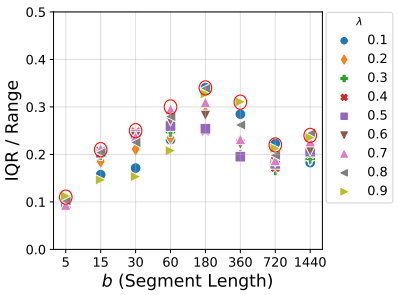
4.4.4 Comparing relative overlap. We calculate the relative overlap in true positives at each choice of segment length for both frameworks (Table 5). In Σ^{-1} and R^{-1}, we consider non-zero elements,



(a) Co-SMs (*DarkSim*): Variability increased with segment length. Non-zero w maximized variability at $b = 1440$.

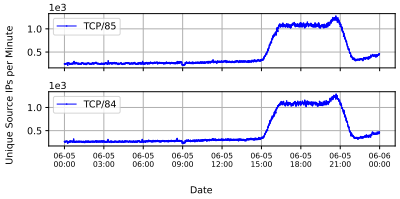


(b) Σ^{-1} (*DarkGLASSO*): Notable variabilities at tail lengths $b \in \{5, 15, 1440\}$ with weak penalizers λ.

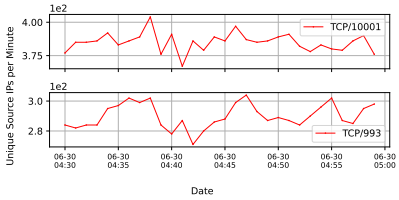


(c) R^{-1} (*DarkGLASSO*): Variabilities increased with segment length up to $b = 360$. Its trend with λ was unclear.

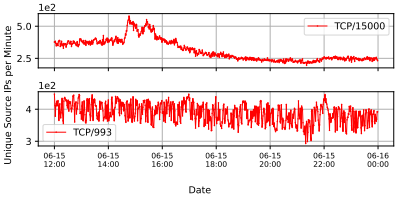
Figure 6: *DarkGLASSO* showcased greater detection discriminability (as measured by variability in matrix L_1-norms) except for day-long $b = 1440$ segments. However, our comparison of detection accuracy (§4.4) indicated not all segment pairs detected in these matrices (produced from parameters circled in red) are true positives.



(a) Group A: TCP/84 and 85's day-long ($b = 1440$) segments identified from *DarkSim* co-SMs share a distinct shape.



(b) Group B: TCP/10001 and 993's 30-minute ($b = 30$) segments identified from R^{-1}matrices share similar overall trends.



(c) Group C: TCP/15000 and 993's 12-hour ($b = 720$) segments identified from R^{-1}matrices lacked notable shared characteristics.

Figure 7: Examples of detected segments' qualitative characteristics used for group assignment.

regardless of scale, as detections. In co-SMs, we consider detections as elements with a similarity score above the 90th percentile for a specific matrix. We also evaluate the 50th percentile as a threshold, *i.e.* we include less similar segment pairs as overlap candidates (Table 11 of Appendix C).

For *DarkGLASSO*, Σ^{-1} missed a majority of co-SM detections (overlap of 4% across all segment lengths). R^{-1}detected a higher proportion (33%), the highest counts at lengths $b \in \{180, 720, 1440\}$. Both missed several key events: (1) a defensive scan conducted by AlphaStrike over ports TCP/27017 and 8085 (Figure 16a of Appendix C); (2) a potential coordinated scan by at least 4 ASes (Figure 9a); (3) a 30-minute outage that saw disconnectivity in nearly 2000 ASes (Figure 16b of Appendix C).

For *DarkSim*, a 90th percentile threshold resulted in an overlap of 31 of 52 (59%) with $\Sigma^{-1}$and 50 of 63 (79%) with R^{-1}. Missed detections each showcased short-lived concurrent drops in sender counts obscured by noisy segment portions. At the 50th percentile threshold, overlaps increased to 44 of 52 (84%) and 61 of 63 (96%), indicating *DarkSim* in fact detected a majority of *DarkGLASSO*'s detections but ranked them weaker in similarity against its other detections.

Takeaway: *DarkSim* showcased perfect precision and detected 70% up to 91% of *DarkGLASSO*'s detections (combined across all trialed segment lengths), missing patterns of short-lived drops. In contrast, *DarkGLASSO* achieved a maximum of 73.3% precision and detected at most 37.5% of *DarkSim*'s detections, missing events including reconnaissance by a defensive scanner, an extended outage, and potential coordinated scan by several ASes.

4.5 Comparing computational performance

To compare computational performance, we measure the wall-times taken by each method's parallel implementation to analyze June's time series (Figure 8). We omitted runtimes for *DarkSim* co-DM to co-SM conversion and *DarkGLASSO*'s computation of Σ and R as they were negligible. Times to generate historical time series from packet captures were equal for both algorithms. Theoretical time complexity of *DarkSim* and *DarkGLASSO* are $O(N \cdot p^2 \cdot b^2)$ and $O(N \cdot K \cdot p^3)$, respectively, further explained in Appendix C.1.

4.5.1 Effects of segment length on runtime. Across all matrix types, longer choices of segments yielded lower wall-times. For *DarkGLASSO*, wall-times decreased at a linear rate with respect to segment length. In contrast, *DarkSim*'s wall-times plateaued beyond 1-hour lengths ($b = 60$).

Times to compute a single 5-minute period's matrix were approximately 0.015, 0.36, and 0.175 seconds (avg. wall-time divided by N

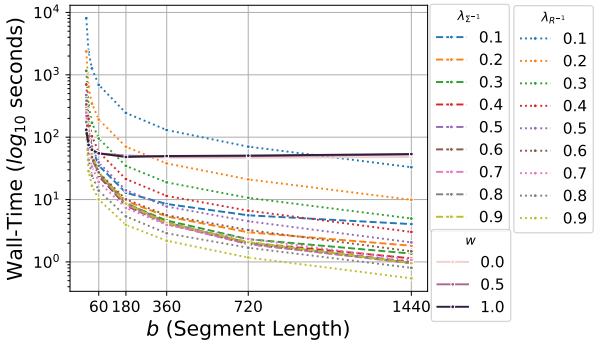


Figure 8: Wall-times to compute (in parallel) method-specific matrices across varied parameterizations.

from Table 6) for co-SM, Σ^{-1}, and R^{-1} respectively. At this length, co-SM computation was roughly 2.4x and 11.6x faster than Σ^{-1} and R^{-1} computation. At $b = 1440$, *DarkGLASSO* outperformed *DarkSim* (respective times for each matrix type were 1.716, 0.048, and 0.21 seconds).

Table 6: Wall-times averaged across method-specific parameters to compute each matrix type.

N	b	*DarkSim* Co-SM	*DarkGLASSO* Σ^{-1}	*DarkGLASSO* R^{-1}
		Time (sec.)	Time (sec.)	Time (sec.)
8640	5	131.13	312.15	1514.74
2880	15	75.57	108.97	426.35
1440	30	61.80	51.94	231.01
720	60	54.71	26.38	127.17
240	180	49.12	9.03	45.77
120	360	48.61	4.82	24.52
60	720	48.94	2.54	13.52
30	1440	51.48	1.46	6.31

4.5.2 Effects of method-specific parameters on runtime. Effects of method-specific parameters on runtimes aligned with our expectations. For *DarkSim*, use of non-zero warp-widths w increased runtimes for each segment length. However, these increases were insignificant as our implementation parallelizes matrix computation. The maximum increase occurred at $b = 1440$, where the runtime for $w = 1$ was 11.3%, or 5.47 seconds, higher than for $w = 0$.

For *DarkGLASSO*, use of larger penalties λ resulted in lower runtimes to estimate Σ^{-1} and R^{-1}. We note that λ's runtime effects were less significant for Σ^{-1} than R^{-1}. The maximum speedup $\lambda = 0.9$ offered over $\lambda = 0.1$ was 4.2x for the former, 71x for the latter due to the conditioning of Σ and R.

Takeaway: Both *DarkSim* and *DarkGLASSO*'s runtimes decreased with larger choices of segment lengths. Despite *DarkGLASSO*'s performance improvements that outpaced *DarkSim*'s, its outputs lacked *DarkSim*'s perfect precision (§4.4.3).

5 CASE STUDIES

We present two case studies to demonstrate the capabilities of *DarkSim*. The first case (§5.1) leveraged fingerprints to quickly identify similar events from many time series. The second case (§5.2) applied *DarkSim* to detect short-lived, high-intensity scanning sourced from a single country.

5.1 Detecting anomalous events after public vulnerability disclosure

Using *DarkSim*, we analyzed traffic across a two-month timeframe around the release of Common Vulnerabilities and Exposures (CVEs) reported in Microsoft's February 2023 Patch Tuesday. Our framework detected patterns of abnormal traffic events in 12/14 ports associated with newly reported vulnerabilities. We further assessed the prevalence of this pattern over an extended 18-month timeframe and identified similar events matched on over 1,600 additional TCP destination ports. Characterization of a subset of these events revealed two groups of networks that repeatedly conducted scans targeting ports of known vulnerabilities.

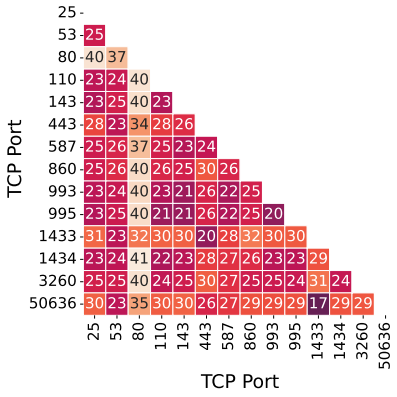
5.1.1 Microsoft's Patch Tuesday. Public disclosure of vulnerabilities as CVEs often triggers Internet-wide scanning observed by network telescopes. For some vulnerabilities, scanning activity related to their exploitation occurs even before their CVE reports and patches are released [67].

On February 14, 2023, as part of its regular software updates, Microsoft released a series of patches to address 75 CVEs [75], including three high-severity Remote Code Execution (RCE) vulnerabilities [53–55]. Our analysis focuses on the 14 TCP destination ports (denoted as TCP/<port>) associated with these three major vulnerabilities listed in Table 7).

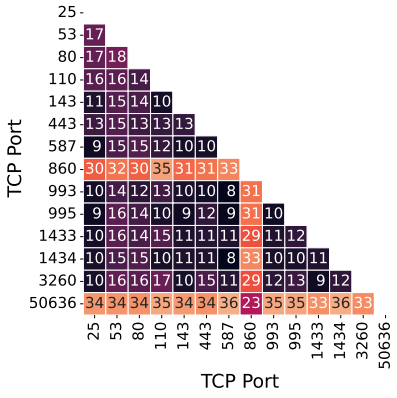
Table 7: Default TCP ports of applications whose RCE vulnerabilities were patched by Microsoft's February 2023 release. We included 80 and 53 due to their occasional use by mail exchanges for web traffic.

Vulnerability	Application	Default TCP Ports
CVE-2023-21803 [55]	Microsoft iSCI Service	860, 3260
CVE-2023-21706 [53]	Microsoft Exchange Server	80, 25, 53, 110, 143, 443, 587, 993, 995, 50636
CVE-2023-21718 [54]	Microsoft SQL Server	1433, 1434

5.1.2 Detecting concurrent anomalies and their pattern templates. To detect possible changes in traffic activity near the CVE release date (February 1 to March 31, 2023), we analyzed the 14 ports' time series of per-minute unique source IP address counts. We assessed their time series for the presence of *concurrent anomalies* in co-DMs (§3.2) by computing DTW scores using a maximal warp-width from contemporaneous day-long segments ($b = 1440, w = 1$). We chose these parameter values since they maximized DTW score variability



(a) Feb. 22



(b) Feb. 23

Figure 9: Global decrease reflected in cross-sectional scores of co-DMs for Feb. 22 and 23, 2023, indicating similar patterns appearing in 12 ports of the latter day's segments.

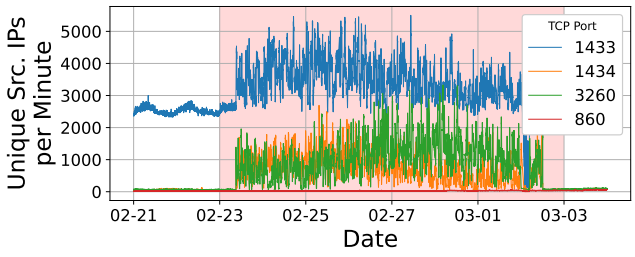


Figure 10: Concurrent anomalies detected in all ports but TCP/860 (for the subset pictured) roughly 9 days after high-severity RCE vulnerabilities were released on Feb. 14, 2023.

in our sensitivity analysis (§4.3). In total, we computed 59 co-DMs (one per day) each of dimension 14×14.

To measure daily changes across cross-sectional DTW scores, we calculated WDs between consecutive co-DM score distributions. Application of a Hampel Filter, using parameter values listed in §3.3, to these distances detected five outliers: February 23 and 24, March 3, 4, and 16. Figure 9 plots scores of the two matrices responsible for February 23's high distance. Further investigation revealed that February 23 and March 3's outliers respectively identified the onset and cessation of *concurrent anomalies* across twelve of the fourteen ports selected for our analysis. Figure 10 depicts the anomalous pattern in four of these ports.

5.1.3 Locating similar events across time with pattern templates. Having detected anomalous patterns in CVE-related ports, we assessed their prevalence across an extended timeframe spanning 18 months (January 1, 2022-June 30, 2023, totaling 546 days). We also expanded our analysis to include all 65,536 TCP destination ports' unique sender IP address count time series. For this event's pattern template, we chose a day-long segment belonging to TCP/3260 that occurred on February 24, 2023. We based our choice on the observation that distinguishing characteristics of the event were unlikely distorted by the low baseline sender IP address count seen for this port prior to the event's occurrence. After scoring our template against daily segments across the 18 months, we computed a 65536×546 *search matrix* (§3.2) to detect *repeated* anomalies.

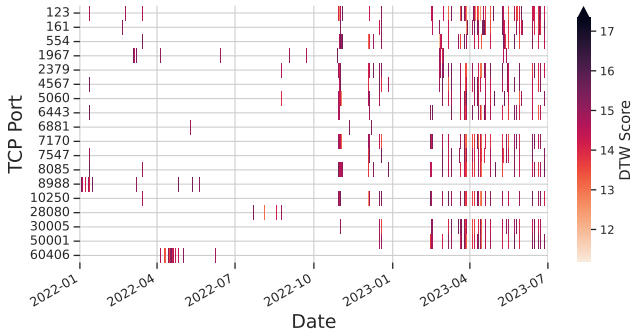


Figure 11: Scores of matched segments for the top-10 ports by match count across our analysis time frame. Notable increases from late 2022 onwards suggest more frequent occurrences of similar events.

We filtered for pattern matches using a cutoff score: the median of scores between segments of TCP/3260 and the 11 ports manually confirmed to also contain anomalous patterns on February 24. Figure 19 of Appendix D depicts the quantity of segment matches by choice of score. Compared to 2022, more matches occurred in 2023, frequently appearing across different ports on the same day (Figure 11), hinting at the increased prevalence of this particular type of scanning behavior.

Table 8 summarizes the ten most frequently targeted ports by their match counts per year. Apart from TCP ports of well-known services, *e.g.*, 123 (NTP), 554 (RSTP), 161 (SNMP over TCP [42]), 5060 (SIP), we identified several others by their continued and newly-trending security concerns, *e.g.*, 7170 (NSRP), 10250, 5060, 7170 (various Kubernetes components) [47], and 50001 (IBM Cloud Orchestrator) [37].

Table 8: Top-10 TCP ports (and their inferred services) by match counts for 2022 (365 days) and 2023 (181 days). 2023 showcases services related to cloud deployment in addition to overall higher match counts.

2022 (Jan. 1 - Dec. 31)			2023 (Jan. 1 - June 31)		
Service	Port	No.	Service	Port	No.
Bittorrent	6881	41	NTP	123	66
	8988	38	NSRP	7170	59
	60406	33	TR-069	30005	58
	1967	22	SNMP	161	56
SIP	5060	22	CPE Mgmt.	8085	55
RSTP	554	22	IBM Cloud Orch.	50001	55
NSRP	7170	20	Kubernetes	2379	54
Thor	28080	20	Kubernetes	10250	52
CPE Mgmt.	8085	20	Kubernetes	6443	52
TR-069	7547	19	Verizon Routers	4567	49

Notably, ports related to known Kubnernetes vulnerabilities [48, 76] appeared in the top-10 ports for 2023. This coincides with reports of increased use of Kubernetes in the cloud [34, 60] and reported campaigns targeting publicly accessible clusters for malware deployment [4, 5, 18].

5.1.4 Characterizing events of matched patterns. To better characterize the events responsible for these patterns, we conducted network traffic analysis of matched segments' time frames and ports using flow-resolution data (FlowTuple [12], traffic aggregated into 5 minute-intervals). For each event, we determined: (1) ASes likely to be responsible for originating abnormal traffic; and (2) scanning behaviors observed for senders from these networks.

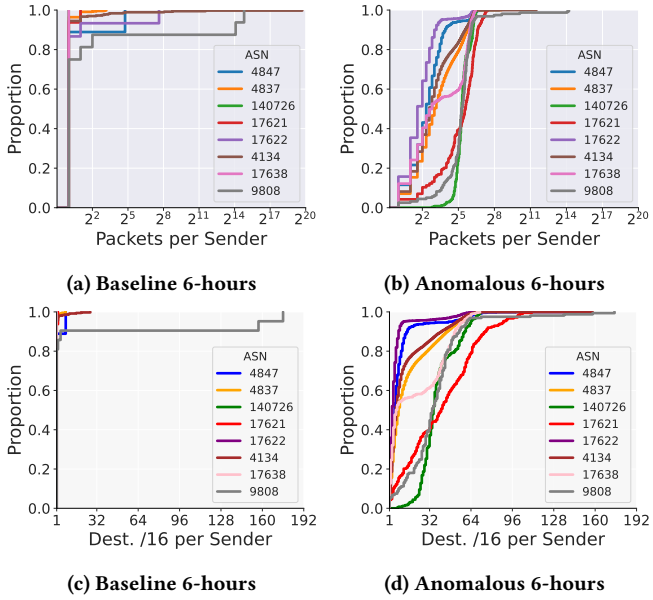
For candidate events, we selected the two lowest-scoring segments, *i.e.,* most-similar matches, per the top-20 ports by match count over both years. We included February 24's segments for the twelve ports from §5.1.2. Due to missing data, we analyzed only 13 out of 52 total port-day pairs.

To determine ASes likely involved in each event, we compared sets of unique senders from two 6-hour windows (one capturing baseline traffic, the other anomalous). We considered ASes whose unique counts increased at least twofold and by a minimum of 100 as likely originators.

Among events, we detected two groups of ASes. The first group consisted of 8 ASes (Table 9) that each appeared in all *strong* matches (4/13) whose pattern characteristics matched TCP/3260's template near-exactly (*i.e.,* rapid onset of new senders, highly variable periodicity and amplitudes). The second group consisted of only one AS (AS135377), a cloud-hosting provider in Hong Kong, that produced *weak* pattern matches (*i.e.,* slow onset of new senders, sub-hourly periodicity but stable amplitudes). TCP ports targeted exclusively by the former group included 25 (SMTP), 2000, 2379 (Kubernetes); by the latter group: 8009 (Apache Tomcat [64]), 7547 (TR-069) and by both, though on different days: 1723 (PTPP). These ports have and continue to be targeted for potential RCE vulnerabilities [17].

Table 9: ASes determined as likely originators of traffic responsible for TCP/3260's anomalous pattern (Fig. 10) and its strong matches (§5.1.4) detected by *DarkSim*.

ASN ↳Sibling ASN(s)	No. Unique Src. IP Address		Δ	
	Baseline	Anomaly	≈%	Abs.
4847	9	907	9977	898
4837	443	15311	3356	14868
↳140726, 17621, 17622	48	2118	4412	2070
4134	362	16667	4504	16305
↳17638	19	435	2189	416
9808	16	158	887	142



(a) Baseline 6-hours (b) Anomalous 6-hours

(c) Baseline 6-hours (d) Anomalous 6-hours

Figure 12: CDFs of packets per sender (12a,12b) and destination subnets (12c,12d) of traffic destined to TCP/3260 for 8 source ASNs identified before and during the anomaly on Feb. 14, 2023. Statistics produced from 5-minute windows using flow-resolution data.

We observed an increase in unique senders, packet rates per sender, and the number of unique telescope /16s targeted per sender for all 8 ASes during strongly matched events. Figure 12 depicts these changes for the 8 ASes during February 14, 2023's event, representative of the other 12 strong matches. We further found that individual senders from these ASes targeted subnets non-sequentially (Figure 17 of Appendix D). The rapid onset of highly similar behaviors from several orgin ASes, resulting in the anomalous pattern signatures *DarkSim* detected, suggests potentially coordinated campaigns directed towards default ports of known vulnerabilities.

Takeaway: We applied *DarkSim* to detect anomalous traffic events targeting TCP ports of vulnerable applications reported in Microsoft's February 2023 Patch Tuesday and further discovered hundreds of historical occurrences. By narrowing the set of candidates, *DarkSim* enabled us to selectively perform flow analysis and characterize these events as potentially coordinated scanning originated repeatedly by a group of the same 8 ASes.

5.2 Needles in a haystack: identifying irregular high-rate scanning events

For our second case study, we employed DarkSim to uncover the most severe shifts across country-level packet count time series. Across a one-year time frame, we discovered repeated scanning campaigns launched by senders from two ASes.

5.2.1 Identifying country-level concurrent anomalies. We applied *DarkSim* with the same parameter values ($b = 1440$, $w = 1$) as the previous case study to packet count time series gelocated for 255 countries across 2022. Here, outputs consisted of 364 co-DMs, each of dimension 255×255. We removed matrices containing invalid comparisons, *i.e.*, scores between segments missing observations due to data loss, before computing a total of 342 WDs. Using a Hampel Filter, parameterized by the same values in §5.1.3, we identified a total of 46 outliers. Appendix D includes a sensitivity analysis of the MAD threshold. We selected two of the top-5 outliers (Figure 13) for flow-resolution traffic analysis to further characterize these events.

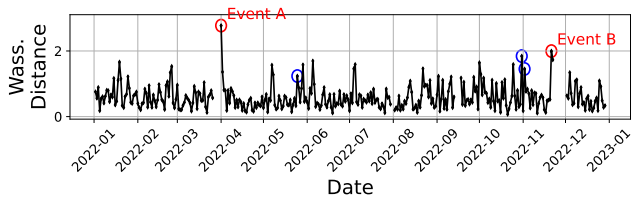


Figure 13: Wasserstein Distances between daily *DarkSim* co-DMs. Gaps indicate absent WDs due to missing time series data. Red encircles Event A and B. Blue encircles 3 additional events detected by *DarkSim*, initiated by AS202425.

5.2.2 Characterizing traffic dynamics of irregular events. Using flow-resolution traffic data, we analyzed events that corresponded to the WDs that occurred on March 31st (Event A) and November 22 (Event B). Event A consisted of two packet surge episodes on the same day from source IPs geolocated to the Netherlands (NL) accompanied by drops from non-NL countries. Event B similarly consisted of a single packet surge, instead from IPs geolocated to the US but accompanied by drops from non-US countries. Table 12 of Appendix D lists the traffic statistics we describe in this section.

As baseline measures, we used traffic metrics of the 20/30-minutes prior to Event A/B. During Event A's two episodes (I, II), the total packet counts increased 132.9% and 96.9% from baseline, a majority sent by NL-geolocated sources (55.5% and 55.6% of total packets, up from 11.5%), specifically by AS202425 (97.7% and 97.3% of NL's packets). In both episodes, the count of unique source IPs from this

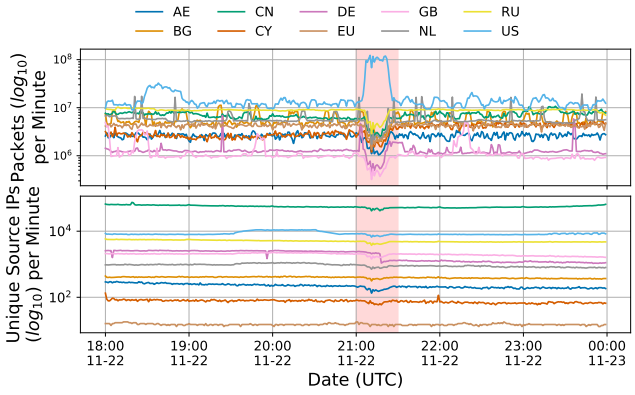


Figure 14: November 22nd's anomaly (Event B) that consisted of an inversion in packet counts between US and non-US countries. Two episodes (I, II) on March 31st (Event A) showed a similar inversion in packet counts but between NL and non-NL countries.

AS did not change significantly, implying an increase in the number of packets sent from each sender. We found that AS202425 was also responsible for an additional 3 events, which also showcased NL packet count surges and non-NL drops, among the top-10 WDs (blue in Figure 13).

During Event B, the total packet counts increased 147.1%, consisting of 75.5% US-originated packets. During the baseline period, a single AS14987 IP address sent only 32 packets to TCP/179. However during the event, 156 AS14987 IPs were responsible for 91.2% and 68.9% of US-originated and of all packet counts, respectively. Targeted port counts increased to 306 TCP and 101 UDP ports with nearly all ports receiving roughly 5.8 million packets. Further, we found that every /16 subnet of UCSD-NT received at least one packet belonging to AS14987-originated traffic.

Since both ASes are hosting companies with reliable Internet connectivity, we believe this activity resulted from their users launching high-rate Internet-wide scanning activities. UCSD Network Telescope operators confirmed that the inversions were likely due to dropped packets caused by high scanning rates of the ASes during events.

Takeaway: By applying *DarkSim* to detect concurrent anomalies from country-level packet count time series, we discovered high-rate scanning campaigns repeatedly originated by two ASes.

6 DISCUSSION AND FUTURE WORK

We discuss limitations of *DarkSim*'s approach and identify areas to improve its evaluation against other methodologies.

Enhancing *DarkSim*'s detection capabilities Time series metrics capturing high amounts of baseline traffic may reduce the efficacy of *DarkSim*'s detection mechanism. This baseline noise can obscure the visibility of low-magnitude anomalies, thus resulting in normative DTW scores. For example, anomalies detected in the unique source IP counts for the three TCP destination ports of Figure 10 would produce negligible change in the counts for a popular port, *e.g.*, TCP/23 from Figure 2b. One potential solution involves

disaggregating time series prior to scoring. Using the previous example, by considering both the source network of traffic in addition to TCP/23 as a destination port, we partition a single time series into its many constituents for comparison. Future work may also consider other distance measures (*e.g.*, [70]) that aim to filter noise algorithmically to produce more accurate time series comparisons.

Improving evaluation robustness Exhaustive labeling of IBR traffic events and their senders remains a resource-intensive task that seldom attains perfect accuracy. For this reason, we evaluated our framework without a comprehensive set of ground truth events. This limited our benchmarks of detection accuracy to only the precision component as we could not evaluate false negative rates. More generally, this limits analysis of the parameterization of statistical techniques used for detection. Future evaluations stand to benefit from high-fidelity event labels that enable researchers to definitively and precisely profile the classes of events that different methodologies can detect.

Expanding evaluation scope Our benchmarks in this work focused on two event detection frameworks that utilize IBR in time series form. As future work, we will expand the scope of our benchmarks to a wider range of approaches to better understand their relative capabilities and tradeoffs. Although results we attained from evaluating historical time series roughly reflect online performance, our future benchmarks will include benchmarks in near-realtime detection settings.

7 RELATED WORK

Detecting Darknet Events Packet header information, *e.g.*, TCP or UDP destination ports, TCP flags, TCP sequence numbers, and TTL values, are commonly used to identify IBR traffic activity. For example, darknet packet signatures that indicate a response (*e.g.*, TCP/ACKs, TCP RSTs, ICMP errors, and DNS responses) result from DoS attacks originated by randomly-spoofed source IP addresses. The fixed nature of these signatures have allowed researchers to apply static rules [58] to infer and characterize *backscatter traffic* [11, 43]. Other packet signatures, though more transient, have identified probing tools and botnet populations [3, 74]. However, despite their efficacy for isolating specific events, filters for these signatures lack the flexibility to adapt to emergent events.

Prior works have trained machine-learning models using *features* of darknet traffic to classify events. By applying both supervised and unsupervised algorithms to traffic data, researchers aim to produce models that identify events by features' statistical relationships [8, 9, 26]. Recent approaches combine neural-networks with unsupervised clustering techniques to detect coordinated scanning in IBR. DarkVec [28] and its predecessors [16, 66] infer malicious darknet sender IP addresses by clustering senders in low-dimensional representations of Word2Vec [57] generated based on sender packet co-occurrence. Kallitsis et al. [45] instead use autoencoders to reduce the dimensionality of their original feature set before clustering senders.

8 CONCLUSION

We designed and implemented *DarkSim*, a top-down analytic framework for detecting anomalies in IBR. Instead of directly processing raw packets, our framework identifies changes in time series of traffic metrics by comparing similarities within and across different time series. To achieve this, *DarkSim* employs a similarity measure, Dynamic Time Warping (DTW), and statistical techniques to identify specific time periods and traffic properties that warrant further analysis.

We showcased the effectiveness of our framework in our benchmark against *DarkGLASSO*, a recent framework that applies the GLASSO algorithm to darknet time series. Whereas *DarkGLASSO* achieved only a maximum precision of 73.3% and a 37.5% overlap with our framework's detections, *DarkSim* achieved perfection precision and a maximum overlap of 91%.

We highlighted *DarkSim*'s practical utility in two case studies. Our analyses successfully captured events that targeted ports associated with new critical vulnerabilities addressed by Microsoft Patch Tuesday, as well as aggressive scanning activities originating from hosting companies in the United States and Netherlands.

ACKNOWLEDGMENTS

The authors thank our anonymous reviewers and shepherd for their constructive feedback on improving the paper. We thank Thy Nguyen for her keen analysis performed during the exploratory stages of this project. This work is based on research sponsored by U.S. NSF grants OAC-2319959, OAC-2131987, CNS-2120399. The views and conclusions are those of the authors and do not necessarily represent endorsements, either expressed or implied, of NSF.

REFERENCES

[1] Ejaz Ahmed, Andrew Clark, and George Mohay. 2009. Characterising anomalous events using change-Point correlation on unsolicited network traffic. In *Proceedings of the Nordic Conference on Secure IT Systems.* https://doi.org/10.5555/3089844.3089855

[2] Ejaz Ahmed, Andrew Clark, and George Mohay. 2009. Effective Change Detection in Large Repositories of Unsolicited Traffic. In *Proceedings of the International Conference on Internet Monitoring and Protection.* https://doi.org/10.1109/ICIMP.2009.8

[3] Manos Antonakakis, Tim April, Michael Bailey, Matthew Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *Proceedings of the USENIX Security Symposium.* https://doi.org/10.5555/3241189.3241275

[4] Aqua. 2023. First-Ever Attack Leveraging Kubernetes RBAC to Backdoor Clusters. Retrieved December 3, 2023 from https://blog.aquasec.com/leveraging-kubernetes-rbac-to-backdoor-clusters

[5] Aqua. 2023. TeamTNT Reemerged with New Aggressive Cloud Campaign. Retrieved December 3, 2023 from https://blog.aquasec.com/teamtnt-reemerged-with-new-aggressive-cloud-campaign

[6] Apache Avro. 2023. Apache Avro - a data serialization system. Retrieved December 3, 2023 from https://avro.apache.org/

[7] Anthony J. Bagnall, Aaron Bostrom, James Large, and Jason Lines. 2016. The Great Time Series Classification Bake Off: An Experimental Evaluation of Recently Proposed Algorithms. Extended Version. (2016). arXiv:1602.01711 http://arxiv.org/abs/1602.01711

[8] Eray Balkanli, Jander Alves, and A. Nur Zincir-Heywood. 2014. Supervised learning to detect DDoS attacks. In *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security.* https://doi.org/10.1109/CICYBS.2014.7013367

[9] Eray Balkanli, A. Nur Zincir-Heywood, and Malcolm I. Heywood. 2015. Feature selection for robust backscatter DDoS detection. In *Proceedings of the IEEE Local Computer Networks Conference Workshops.* https://doi.org/10.1109/LCNW.2015.7365905

[10] Donald J. Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining.* https://doi.org/10.5555/3000850.3000887

[11] Norbert Blenn, Vincent Ghiëtte, and Christian Doerr. 2017. Quantifying the Spectrum of Denial-of-Service Attacks through Internet Backscatter. In *Proceedings of the International Conference on Availability, Reliability and Security.*

https://doi.org/10.1145/3098954.3098985
[12] CAIDA. 2021. FlowTuple. Retrieved September 4, 2024 from https://stardust.caida.org/docs/data/flowtuple/
[13] CAIDA. 2021. RSDoS. Retrieved September 4, 2024 from https://stardust.caida.org/docs/data/dos/
[14] CAIDA. 2021. STARDUST Grafana dashboard. Retrieved September 4, 2024 from https://explore.stardust.caida.org
[15] CAIDA. 2021. STARDUST. UCSD network telescope. Retrieved September 4, 2024 from https://stardust.caida.org
[16] Dvir Cohen, Yisroel Mirsky, Manuel Kamp, Tobias Martin, Yuval Elovici, Rami Puzis, and Asaf Shabtai. 2020. DANTE: A Framework for Mining and Monitoring Darknet Traffic. In *Proceedings of the European Symposium on Research in Computer Security*. https://doi.org/10.1007/978-3-030-58951-6_5
[17] Corelight. 2022. Detecting CVE-2022-23270 in PPTP. Retrieved December 3, 2023 from https://corelight.com/blog/detecting-cve-2022-23270-in-pptp
[18] Crowdstrike. 2023. CrowdStrike Discovers First-Ever Dero Cryptojacking Campaign Targeting Kubernetes. Retrieved December 3, 2023 from https://www.crowdstrike.com/blog/crowdstrike-discovers-first-ever-dero-cryptojacking-campaign-targeting-kubernetes/
[19] Alberto Dainotti, Karyn Benson, Alistair King, kc claffy, Michael Kallitsis, Eduard Glatz, and Xenofontas Dimitropoulos. 2014. Estimating Internet Address Space Usage through Passive Measurements. *SIGCOMM Comput. Commun. Rev.* 44, 1 (2014). https://doi.org/10.1145/2567561.2567568
[20] Alberto Dainotti, Alistair King, Kimberly Claffy, Ferdinando Papale, and Antonio Pescape. 2015. Analysis of a "/0" Stealth Scan From a Botnet. *IEEE/ACM Transactions on Networking* 23, 2 (2015). https://doi.org/10.1109/tnet.2013.2297678
[21] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. 2014. Analysis of Country-Wide Internet Outages Caused by Censorship. *IEEE/ACM Trans. Netw.* 22, 6 (2014). https://doi.org/10.1109/TNET.2013.2291244
[22] Dask Development Team. 2016. Dask: Library for dynamic task scheduling. Retrieved September 4, 2024 from https://dask.org
[23] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. 2008. Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. *Proceedings of the VLDB Endowment* 1, 2 (2008). https://doi.org/10.14778/1454159.1454226
[24] Zakir Durumeric, Michael Bailey, and J. Alex Halderman. 2014. An Internet-Wide View of Internet-Wide Scanning. In *Proceedings of the USENIX Security Symposium*. https://doi.org/10.5555/2671225.2671230
[25] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2007. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9, 3 (2007). https://doi.org/10.1093/biostatistics/kxm045
[26] Nobuaki Furutani, Jun Kitazono, Seiichi Ozawa, Tao Ban, Junji Nakazato, and Jumpei Shimamura. 2015. Adaptive DDoS-Event Detection from Big Darknet Traffic Data. In *Proceedings of the International Conference on Neural Information Processing*. https://doi.org/10.1007/978-3-319-26561-2_45
[27] Max Gao and Ricky K. P. Mok. 2022. A scalable network event detection framework for darknet traffic. In *Proceedings of the ACM Internet Measurement Conference*. https://doi.org/10.1145/3517745.3563015
[28] Luca Gioacchini, Luca Vassio, Marco Mellia, Idilio Drago, Zied Ben Houidi, and Dario Rossi. 2021. DarkVec: Automatic Analysis of Darknet Traffic with Word Embeddings. In *Proceedings of the International Conference on Emerging Networking EXperiments and Technologies*. https://doi.org/10.1145/3485983.3494863
[29] Robert Graham. 2021. MASSCAN: Mass IP port scanner. Retrieved September 4, 2024 from https://github.com/robertdavidgraham/masscan
[30] Frank R. Hampel. 1974. The Influence Curve and Its Role in Robust Estimation. *J. Amer. Statist. Assoc.* 69, 346 (1974). https://doi.org/10.1080/01621459.1974.10482962
[31] Chansu Han, Jumpei Shimamura, Takeshi Takahashi, Daisuke Inoue, Masanori Kawakita, Jun'ichi Takeuchi, and Koji Nakao. 2019. Real-Time Detection of Malware Activities by Analyzing Darknet Traffic Using Graphical Lasso. In *Proceedings of the IEEE International Conference On Trust, Security And Privacy In Computing And Communications/IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. https://doi.org/10.1109/TrustCom/BigDataSE.2019.00028
[32] Chansu Han, Jun'ichi Takeuchi, Takeshi Takahashi, and Daisuke Inoue. 2022. Dark-TRACER: Early Detection Framework for Malware Activity Based on Anomalous Spatiotemporal Patterns. *IEEE Access* 10 (2022). https://doi.org/10.1109/access.2022.3145966
[33] Chansu et al. Han. 2023. Dark-TRACER. Retrieved September 4, 2024 from https://github.com/Gotchance/Dark-TRACER
[34] Red Hat. 2023. Kubernetes adoption, security, and market trends report 2023. Retrieved December 3, 2023 from https://www.redhat.com/en/resources/kubernetes-adoption-security-market-trends-overview
[35] Rick Hofstede, Pavel Čeleda, Brian Trammell, Idilio Drago, Ramin Sadre, Anna Sperotto, and Aiko Pras. 2014. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys & Tutorials* 16, 4 (2014). https://doi.org/10.1109/COMST.2014.2321898
[36] Cho-Jui Hsieh, M. Sustik, I. Dhillon, and P. Ravikumar. 2014. QUIC: Quadratic Approximation for Sparse Inverse Covariance Estimation. *Journal of Machine Learning Research* 15 (2014). https://doi.org/10.5555/2627435.2697058
[37] IBM. 2021. Ports used by IBM Cloud Orchestrator. Retrieved December, 3, 2023 from https://www.ibm.com/docs/en/cloud-orchestrator/2.5.0.3?topic=reference-ports-used-by-cloud-orchestrator
[38] influxdata. 2023. InfluxDB. Retrieved September 4, 2024 from https://www.influxdata.com
[39] Daisuke Inoue, Katsunari Yoshioka, Masashi Eto, Masaya Yamagata, Eisuke Nishino, Jun'ichi Takeuchi, Kazuya Ohkouchi, and Koji Nakao. 2009. An incident analysis system NICTER and its analysis engines based on data mining techniques. In *Proceedings of the Advances in Neuro-Information Processing: International Conference*. https://doi.org/10.1007/978-3-642-02490-0_71
[40] Fumitada Itakura. 1975. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23, 1 (1975). https://doi.org/10.1109/TASSP.1975.1162641
[41] ITU. 2024. Internet Facts and Figures 2023. Retrieved January 10, 2024 from https://www.itu.int/itu-d/reports/statistics/2023/10/10/ff23-internet-traffic/
[42] TU Braunschweig J. Schoenwaelder. 2002. Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping. Retrieved December 3, 2023 from https://www.rfc-editor.org/rfc/rfc3430.html
[43] Mattijs Jonker, Alistair King, Johannes Krupp, Christian Rossow, Anna Sperotto, and Alberto Dainotti. 2017. Millions of Targets under Attack: A Macroscopic Characterization of the DoS Ecosystem. In *Proceedings of the ACM Internet Measurement Conference*. https://doi.org/10.1145/3131365.3131383
[44] Mattijs Jonker, Aiko Pras, Alberto Dainotti, and Anna Sperotto. 2018. A first joint look at DOS attacks and BGP blackholing in the wild. In *Proceedings of the ACM Internet Measurement Conference*. https://doi.org/10.1145/3278532.3278571
[45] Michalis Kallitsis, Rupesh Prajapati, Vasant Honavar, Dinghao Wu, and John Yen. 2022. Detecting and Interpreting Changes in Scanning Behavior in Large Network Telescopes. *IEEE Transactions on Information Forensics and Security* 17 (2022). https://doi.org/10.1109/tifs.2022.3211644
[46] Leonid V Kantorovich. 1960. Mathematical methods of organizing and planning production. *Management science* 6, 4 (1960). https://doi.org/10.1287/mnsc.6.4.366
[47] Kubernetes. 2022. Ports and Protocols. Retrieved December 3, 2023 from https://kubernetes.io/docs/reference/networking/ports-and-protocols/
[48] RedHunt Labs. 2023. Thousands of Unsecured Kubernetes Clusters Exposed. Retrieved December 3, 2023 from https://redhuntlabs.com/blog/unsecured-kubernetes-clusters-exposed/
[49] Jason Laska and Manjari Narayan. 2017. skggm 0.2.7: A scikit-learn compatible package for Gaussian and related Graphical Models. Retrieved September 4, 2024 from https://doi.org/10.5281/zenodo.830033
[50] Bo Liu, Wenpeng Luan, and Yixin Yu. 2017. Dynamic time warping based non-intrusive load transient identification. *Applied Energy* 195 (2017). https://doi.org/10.1016/j.apenergy.2017.03.010
[51] Victor Maus, Gilberto Câmara, Marius Appel, and Edzer Pebesma. 2019. dtwSat: Time-Weighted Dynamic Time Warping for Satellite Image Time Series Analysis in R. *Journal of Statistical Software* 88, 5 (2019). https://doi.org/10.18637/jss.v088.i05
[52] et al. Meert, Wannes. 2020. DTAIDistance (v2.3.10). Retrieved September 4, 2024 from https://github.com/wannesm/dtaidistance
[53] Microsoft. 2023. CVE-2023-21706. Retrieved December 3, 2023 from https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-21706
[54] Microsoft. 2023. CVE-2023-21718. Retrieved December 3, 2023 from https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-21718
[55] Microsoft. 2023. CVE-2023-21803. Retrieved December 3, 2023 from https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-21803
[56] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. 2023. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. (2023). arXiv:2304.13029 https://arxiv.org/abs/2304.13029
[57] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Advances in Neural Information Processing Systems*. https://doi.org/10.5555/2999792.2999959
[58] David Moore, Colleen Shannon, Douglas J. Brown, Geoffrey M. Voelker, and Stefan Savage. 2006. Inferring Internet Denial-of-Service Activity. *ACM Trans. Comput. Syst.* 24, 2 (2006). https://doi.org/10.1145/1132026.1132027
[59] Abdullah Mueen and Eamonn Keogh. 2016. Extracting Optimal Performance from Dynamic Time Warping. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2945383
[60] Help net Security. 2023. Kubernetes adoption creates new cybersecurity challenges. Retrieved December 3, 2023 from https://www.helpnetsecurity.com/2023/11/13/cloud-native-environments-risks/
[61] Ramakrishna Padmanabhan, Arturo Filastò, Maria Xynou, Ram Sundara Raman, Kennedy Middleton, Mingwei Zhang, Doug Madory, Molly Roberts, and Alberto Dainotti. 2021. A Multi-Perspective View of Internet Censorship in Myanmar. In

Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet. 27–36. https://doi.org/10.1145/3473604.3474562

[62] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 262–270. https://doi.org/10.1145/2339530.2339576

[63] Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. 2015. On Wasserstein Two Sample Testing and Related Families of Nonparametric Tests. arXiv:1509.02237 [math.ST]

[64] RedHat. 2024. AJP File Read/Inclusion in Apache Tomcat (CVE-2020-1938) and Undertow (CVE-2020-1745). Retrieved December 3, 2023 from https://access.redhat.com/solutions/4851251

[65] Merit Research. 2021. Darknet Processing. Retrieved September 4, 2024 from https://github.com/Merit-Research/darknet-events

[66] Markus Ring, Alexander Dallmann, Dieter Landes, and Andreas Hotho. 2017. IP2Vec: Learning Similarities Between IP Addresses. In *Proceedings of the IEEE International Conference on Data Mining Workshops*. https://doi.org/10.1109/ICDMW.2017.93

[67] Jukka Ruohonen. 2019. A look at the time delays in CVSS vulnerability scoring. *Applied Computing and Informatics* 15, 2 (2019). https://doi.org/10.1016/j.aci.2017.12.002

[68] H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1 (1978). https://doi.org/10.1109/TASSP.1978.1163055

[69] Bertil Schmidt and Christian Hundt. 2020. CuDTW++: Ultra-Fast Dynamic Time Warping on CUDA-Enabled GPUs. In *Proceedings of Euro-Par: Parallel Processing*. https://doi.org/10.1007/978-3-030-57675-2_37

[70] Patrick Schäfer. 2015. Scalable time series classification. *Data Mining and Knowledge Discovery* 30, 5 (2015). https://doi.org/10.1007/s10618-015-0441-y

[71] SDSC. 2020. Expanse. Retrieved September 4, 2024 from https://www.sdsc.edu/services/hpc/expanse/

[72] Diego Silva, Rafael Giusti, Eamonn Keogh, and Gustavo Batista. 2018. Speeding up similarity search under dynamic time warping by pruning unpromising alignments. *Data Mining and Knowledge Discovery* 32 (2018). https://doi.org/10.1007/s10618-018-0557-y

[73] Raffaele Sommese, KC Claffy, Roland van Rijswijk-Deij, Arnab Chattopadhyay, Alberto Dainotti, Anna Sperotto, and Mattijs Jonker. 2022. Investigating the Impact of DDoS Attacks on DNS Infrastructure. In *Proceedings of the ACM Internet Measurement Conference*. ACM. https://doi.org/10.1145/3517745.3561458

[74] Akira Tanaka, Chansu Han, and Takeshi Takahashi. 2023. Detecting Coordinated Internet-Wide Scanning by TCP/IP Header Fingerprint. *IEEE Access* 11 (2023). https://doi.org/10.1109/ACCESS.2023.3249474

[75] Tenable. 2023. Microsoft's February 2023 Patch Tuesday Addresses 75 CVEs (CVE-2023-23376). Retrieved December 3, 2023 from https://www.tenable.com/blog/microsofts-february-2023-patch-tuesday-addresses-75-cves-cve-2023-23376

[76] Unit42. 2023. Unsecured Kubernetes Instances Could Be Vulnerable to Exploitation. Retrieved December 3, 2023 from https://unit42.paloaltonetworks.com/unsecured-kubernetes-instances/

[77] Leonid Nisonovich Vaserstein. 1969. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii* 5, 3 (1969).

[78] Gang-Jin Wang, Chi Xie, Feng Han, and Bo Sun. 2012. Similarity measure and topology evolution of foreign exchange markets using dynamic time warping method: Evidence from minimal spanning tree. *Physica A: Statistical Mechanics and its Applications* 391, 16 (2012). https://doi.org/10.1016/j.physa.2012.03.036

A ETHICS

We do not disclose IPv4 addresses responsible for sending the traffic that we analyzed in our work.

B METHOD ADDENDUMS

B.1 The Dynamic Time Warping Algorithm

The Dynamic Time Warping (DTW) algorithm employs a dynamic programming approach to quantify the dissimilarity, *i.e.*, optimal alignment cost, between observations of two time series segments. A single parameter parameterizes DTW: the warp-width, a constraint on the maximal temporal distance between time series observations used to calculate a score. Two well-known variants exist for defining the region produced using a specified warp-width: the Sakoe-Chiba band [68] and Itakura parallelogram [40]. In this paper we use the former.

	(-)	S_0	S_1	S_2	S_3
(-)	0	∞	∞	∞	∞
Q_0	∞	0	16	17	17
Q_1	∞	0	16	17	17
Q_2	∞	16	0	9	25
Q_3	∞	17	9	0	1

Q_0	Q_1	Q_2	Q_3
1	1	5	2

S_0	S_1	S_2	S_3
1	5	2	1

Figure 15: A fully computed DTW cost-matrix (shortest cost path marked in red) between two time series. Right: Sample values for Q and S.

While its theoretical computation complexity possesses a quadratic upper-bound, optimized implementations designed for modern hardware architectures can dramatically reduce its empirical runtimes [69].

Input: $Q_0 \dots Q_{M-1}, S_0 \dots S_{N-1}$
Output: Dissimilarity Score

$D \in \mathbb{R}^{(M+1)\times(N+1)}$ ←cost matrix
$D_{0,0} \leftarrow 0$
for $i \leftarrow 1$ to N: $D_{0,i} = \infty$
for $j \leftarrow 1$ to M: $D_{j,0} = \infty$
for $i \leftarrow 1$ to M **do**
  for $j \leftarrow 1$ to N **do**

$$D_{i,j} = dist(Q_i, S_j) + min \begin{cases} D_{i,j-1} \\ D_{i-1,j} \\ D_{i-1,j-1} \end{cases}$$

  end for
end for
return $\sqrt{D_{M,N}}$

B.2 Converting Dissimilarity to Similarity

While optional for most cases, we detail these steps to convert dissimilarity to similarity scores for analysis that depend on the semantic meaning of score directionality. Our analysis in §4.2 requires that we use *similarity matrices* to compare *DarkSim* against *DarkGLASSO*.

$$s_{ij} = \frac{max(\hat{m}_i) - m_{ij}}{max(\hat{m}_i)} \quad , \forall i, j \in \{1, \dots, N\} \tag{3}$$

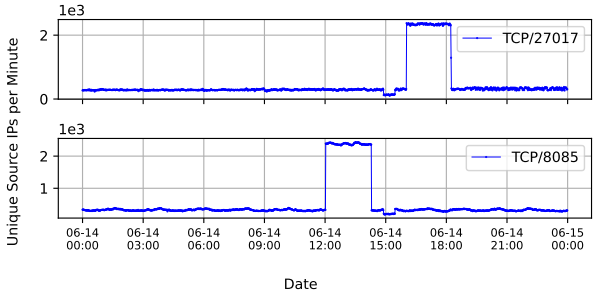
Equation 3 summarizes the conversion of a dissimilarity matrix m to a similarity matrix s. From processed segments, we first obtain $\hat{m}$, a dissimilarity matrix consisting of pairwise Euclidean Distances (ED) (*i.e.*, DTW scores computed using a zero warp-width $w = 0$). We additionally obtain m, the dissimilarity matrix computed using a specific warp-width value. Each row-wise maximum ED distance in $\hat{m}$ represents the empirical upper bound for a given segment [72] and thus we use these to invert and scale m's individual scores onto the continuous range between 0 and 1.

Table 10: Group assignments of *DarkSim* and *DarkGLASSO*'s detections (Groups A and B counted as true positives and Group C as false positives).

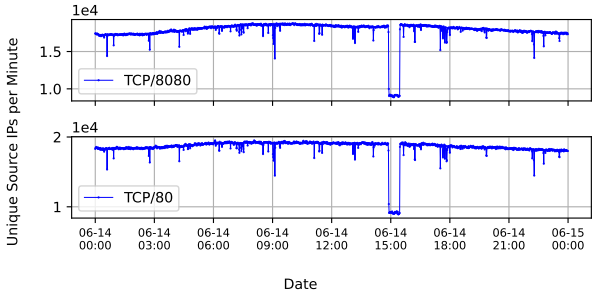
b	*DarkSim* Co-SM			*DarkGLASSO* Σ_{-}^{-1}			R_{-}^{-1}		
	TP		FP	TP		FP	TP		FP
	A	B	C	A	B	C	A	B	C
5	15	0	0	0	5	10	1	3	11
15	15	0	0	0	4	11	3	4	8
30	15	0	0	0	6	9	2	2	11
60	15	0	0	0	10	5	5	5	5
180	15	0	0	0	6	9	9	1	5
360	15	0	0	1	6	8	10	1	4
720	15	0	0	0	8	7	6	2	7
1440	15	0	0	5	1	9	8	1	6

Table 11: Overlap of *DarkSim* and *DarkGLASSO*'s detections. S' denotes the set of detections compared against.

b	*DarkSim* Co-SM (90th)		Co-SM (50th)		*DarkGLASSO* Σ_{-}^{-1}	R_{-}^{-1}
	$S' = \Sigma_{-}^{-1}$	$S' = R_{-}^{-1}$	$S' = \Sigma_{-}^{-1}$	$S' = R_{-}^{-1}$	S' = Co-SM	
5	0/5	1/4	3/5	2/4	1/15	1/15
15	2/4	6/7	3/4	7/7	0/15	2/15
30	1/6	3/4	4/6	4/4	2/15	2/15
60	7/10	8/10	10/10	10/10	0/15	4/15
180	6/6	10/10	6/6	10/10	0/15	9/15
360	5/7	8/11	6/7	11/11	0/15	4/15
720	7/8	8/8	8/8	8/8	2/15	10/15
1440	3/6	6/9	4/6	9/9	0/15	8/15



(a) Increases in unique source IP address counts to ports 27017 and 8085, originating from the IP range belonging to AlphaStrike, an acknowledged scanner.



(b) Drops in unique source IP address counts to ports 8080 and 80 (two among nearly 2000).

Figure 16: Notable events missed by *DarkGLASSO* that were detected by *DarkSim*.

C EVALUATION ADDENDUMS

Table 10 lists the complete assignment counts for *DarkSim* and *DarkGLASSO*'s detections used to calculate precision. Table 11 shows detection overlap in fraction form where the denominators are the true positive counts taken from Table 10. Figure 16 plots two notable *DarkSim* detections missed by *DarkGLASSO*.

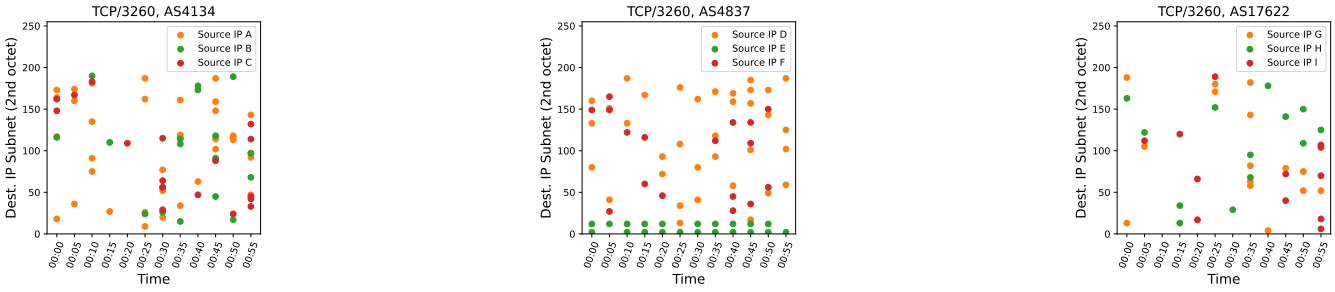
C.1 Comparing theoretical time complexity

DarkSim's time complexity consists of the number of operations to compute DTW scores for N co-DMs, each containing $\frac{p(p-1)}{2}$ unique scores. Since computation of a single score is bounded by $O(b^2)$ operations, the combined bound for an entire analysis workload is $O(N \cdot p^2 \cdot b^2)$ (unchanged by co-SM conversion).

DarkGLASSO's time complexity to estimate either type of inverse matrix combines the number of operations from two steps: 1) computation of either Σ or R; and 2) the optimization procedure of GLASSO. For an entire analysis workload, the first step is bounded by $O(N \cdot p^2 \cdot b)$ operations, similar to *DarkSim*'s time complexity except that pairwise covariance/correlation computes in linear time. The second step applies GLASSO's optimization procedure per matrix. Each matrix requires a variable number of iterations, K, to reach a stopping condition (larger λ penalities generally reduce this number). Each iteration entails $O(p^3)$ operations. The overall bound for a workload is $O(N \cdot K \cdot p^3)$.

Table 12: Overview of traffic metrics before and during concurrent anomalies of Mar. 31st (Event A) and Nov. 22 (Event B).

	Event	Time Interval (UTC)	Source		Packet Counts			Uniq. Src. IP Counts			Uniq. Dest. Port Counts (by AS)	
			Country	AS	Total	Country	AS	Total	Country	AS	TCP	UDP
A	Baseline	2022-03-31 13:20-13:50	NL	202425	1.97e9	2.38e8	1.76e8	1.91e6	2534	59	2102	71
	Anomaly I	2022-03-31 13:50-14:25	NL	202425	3.70e9	1.94e9	1.72e9	1.93e6	2726	55	2245	69
	Anomaly II	2022-03-31 14:45-15:15	NL	202425	3.44e9	1.72e9	1.66e9	1.53e6	2664	55	2278	69
B	Baseline	2022-11-22 20:45-21:05	US	14987	1.22e9	2.52e7	32	5.4e5	2.2e4	1	1	0
	Anomaly	2022-11-22 21:05-21:25	US	14987	3.03e9	2.29e9	2.08e9	5.01e5	1.8e4	156	306	101



(a) AS4134 **(b) AS4837** **(c) 17622**

Figure 17: Scanning strategy for 3 ASes each partially responsible for anomalous activity to TCP/3260 (§5.1.4) revealed by sequences of /16 UCSD-NT subnets that received at least 1 packet from source IPs for a given AS.

D CASE STUDY ADDENDUMS

For §5.1, Figure 17 depicts the scanning strategy observed for ASes responsible for the anomalous patterns of Figure 10 for TCP/3260. We produce these results by analyzing flow-resolution (FlowTuple [12]) traffic data, inspecting the sequence of /16 subnets of UCSD-NT targeted by individual senders belonging to AS networks responsible for the anomalies. For §5.1.3, we supplement match analysis with an assessment of score threshold choice on unique ports matched (out of 65536 possible TCP ports) and unique days matched (out of 546). The CDFs in Figure 19 provide a rough approximation of the quantity of total positives though we did not fully assess the accuracy of matches.

For §5.2.1, we assess the proportion of total WDs identified as outliers (out of 542) based on the MAD multiplier used in the Hampel Filter (Figure 18).

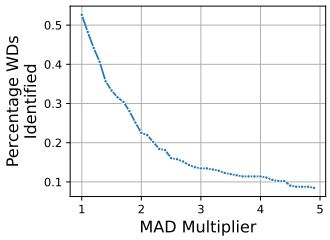
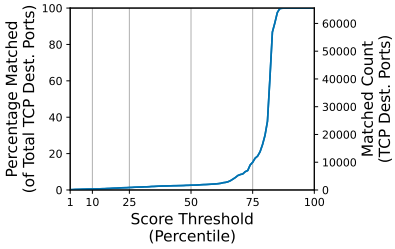
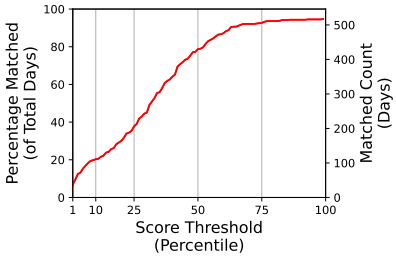


Figure 18: Proportion of WDs identified as outliers based on MAD threshold (§5.2.1).



(a)



(b)

Figure 19: Search matrix match counts by score threshold (§5.1.3).