Hybrid connection and host clustering for community detection in spatial-temporal network data

Roeling, M.P.; Nadeem, A.; Verwer, S.E.

DOI
10.1007/978-3-030-65965-3_12

Publication date
2020

Document Version
Final published version

Published in
ECML PKDD 2020 Workshops - Workshops of the European Conference on Machine Learning and Knowledge Discovery in Databases ECML PKDD 2020

Hybrid Connection and Host Clustering for Community Detection in Spatial-Temporal Network Data

Mark Patrick Roeling[1,2](✉), Azqa Nadeem[1], and Sicco Verwer[1]

[1] Department Intelligent Systems, Cyber-Security, Technical University of Delft, Delft, The Netherlands
m.p.roeling@tudelft.nl
[2] Department of Statistics, University of Oxford, Oxford, UK

Abstract. Network data clustering and sequential data mining are large fields of research, but how to combine them to analyze spatial-temporal network data remains a technical challenge. This study investigates a novel combination of two sequential similarity methods (Dynamic Time Warping and N-grams with Cosine distances), with two state-of-the-art unsupervised network clustering algorithms (Hierarchical Density-based Clustering and Stochastic Block Models). A popular way to combine such methods is to first cluster the sequential network data, resulting in connection types. The hosts in the network can then be clustered conditioned on these types. In contrast, our approach clusters nodes and edges in one go, i.e., without giving the output of a first clustering step as input for a second step. We achieve this by implementing sequential distances as covariates for host clustering. While being fully unsupervised, our method outperforms many existing approaches. To the best of our knowledge, the only approaches with comparable performance require manual filtering of connections and feature engineering steps. In contrast, our method is applied to raw network traffic. We apply our pipeline to the problem of detecting infected hosts (network nodes) from logs of unlabelled network traffic (sequential data). On data from the Stratosphere IPS project (CTU-Malware-Capture-Botnet-91), which includes malicious (Conficker botnet) as well as benign hosts, we show that our method perfectly detects peripheral, benign, and malicious hosts in different clusters. We replicate our results in the well-known ISOT dataset (Storm, Waledac, Zeus botnets) with comparable performance: conjointly, 99.97% of nodes were categorized correctly.

Keywords: Network data · Unsupervised learning · Clustering · Spatio-temporal

1 Introduction

Spatial-temporal network data have a spatial structure, where observations are linked via single or multiple features, and a temporal structure, meaning multiple time-points are (partly) available. The analyses of the spatial element is

I. Koprinska et al. (Eds.): ECML PKDD 2020 Workshops, CCIS 1323, pp. 178–204, 2020.
https://doi.org/10.1007/978-3-030-65965-3_12

usually performed via network clustering, which is a large field of research where a graph ($\mathcal{G}$), consisting of nodes ($\mathcal{V}$) and edges ($\mathcal{E}$), is represented by one or more pairwise distance matrices subject to an algorithm to group observations with, relatively speaking, small distances [21,25,40,44,48]. There are roughly two kinds of clustering methods: those that cluster edges (e.g. spectral-, density-, or centroid based clustering methods [10,30]) and those that cluster nodes (e.g. community detection algorithms like Louvain clustering [4] or mixture clustering like the Stochastic Block Model [1]).

The analyses of the temporal aspect is equally complex. Apart from collapsing time-points by analyzing the mean of multiple events [11], some methods allow to analyse time-series as discrete windows. Examples of these methods are 1) creating windows and train models for each window so that state-changes over time can be identified [31]; 2) treating time as a latent variable in latent variable growth models [22]; 3) creating temporal graphs so that every pairwise interaction over time becomes a link [24]; 4) the analyses of network evolution with Stochastic Actor Based Models [41]; 5) Temporal Exponential Random Graph Models [18]; and 6) Time-contrastive learning [19]. Even more complex is the analyses of streaming data, where time cannot be treated as a strictly discrete variable either due to an arbitrary sequence in time where cutting windows is difficult, or a negative balance between the volume of time windows and the specificity (larger time windows equals lower specificity).

This paper focuses on unsupervised clustering of streaming spatial-temporal network data by combining node and edge clustering. We aim to present a reliable procedure to communities of nodes with converging behavior, without the need for a labelled dataset and not requiring manual feature engineering or filtering steps. Our method computes pairwise edge distances based on the sequential behavior of network connections using Dynamic Time Warping (distance measure for continuous sequences) and N-grams with Cosine distances (for nominal sequences), as implemented in the MalPaCA tool [33]. In order to include these distances in node clustering, the pairwise distances are aggregated via Principal Component Analysis into a small set of features. These features are added as co-variates to a node clustering algorithm based on Stochastic Block Models (SBMs), which is a well-known generative model for random graphs that produces graphs containing communities. Here, those subgroups represent hosts characterized by being connected with one another with particular edge densities [32]. Our SBM-definition is based on a recent review [27].

SBMs are attractive because they seek highly connected blocks in network connections while allowing the inclusion of features, in a statistically tractable way. This removes the need to first cluster the sequential data before analyzing the network structure or attributes as both are considered in one single node clustering algorithm. Our approach is complementary to earlier work [36] where hosts and connections were classified sequentially by first filtering P2P hosts and then categorizing P2P traffic. Using sequential features is beneficial since it reduces the required number of features as all variation is (assumed to be) captured by the pairwise sequential distance [26,33]. Our approach (shown graphically in Fig. 1) does not require a priori (manual) host or sequence filtering and uses as input raw packet capture (.pcap) files.

We test our method in the setting of botnet-infected computers. Botnets are networks of computers that are infected with malware and are under the control of a botnet controller, able to use the computers for nefarious activities. Infection status is usually unknown to users or controllers and incomplete, meaning that in a large network not all computers are infected but only a relatively small number of machines can be part of a botnet. This motivates an unsupervised approach to cluster the hosts in a computer network, thereby uncovering yet unknown (latent) groups of similarly behaving hosts. The idea is that all infected hosts show different behavior from the normal hosts in a network and can thus be singled out, preferably in one or more dedicated clusters. We experiment with different packet thresholds to show which data-specific cutoffs are optimal (i.e. short but still informative). The reliability of our method is investigated by replicating the main result with another dataset containing different botnet captures.

This paper presents the following contributions:

- We present a clustering method of network data that does not require manual filtering of observations.
- Clustering of nodes as well as edges in spatial-temporal network data is conducted in one procedure.
- We present a competitive performance in the setting of detecting malware infected computers (bots) and replicate our main result in different types of botnets.

2 Related Work

To date, a common strategy is to collapse temporal data into aggregate values and neglect spatial structure [2,7,11,13,15,17,29,34,36,38,39,42,50,51]. This causes a loss of information as researchers remove streams of data that only occur once (e.g. because these connections are uninformative when calculating the variance of inter arrival time between packets in a sequence of connections). Apart from some studies using time-windows [16], removing temporal information by collapsing streaming data complicates botnet classification [37]. Neglecting spatial structure in botnet detection is equally problematic because this structure is informative for infection status [9]: the members of a botnet are more likely to have mutual contacts with each other than with benign hosts.

Another issue is that many studies apply some kind of manual filtering prior to analysis (e.g. removing approved DNS addresses via white-listing based on Alexa [17,39] or other rule based exclusion criteria (e.g. [5,36,46]). It is unclear whether the obtained results are due to the analysis or filtering steps. Manual feature engineering may also bias the results of these experiments [20], especially when combined with sparsely reported procedures and outcomes (e.g. [45,49]). Finally, only a few studies apply methods that do not require a labelled dataset (unsupervised learning: [15,50]). Especially in the botnet setting where computers are *zombies* per definition, the dependence on a labelled dataset is an important shortcoming for operational usefulness.

3 Methods

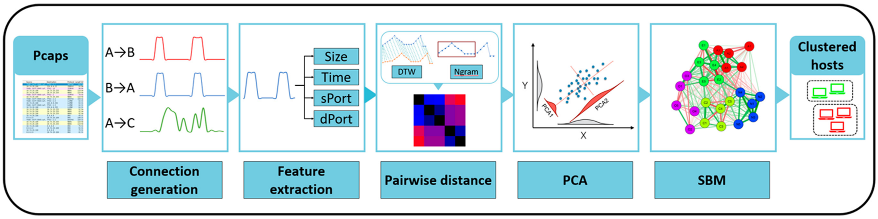


Fig. 1. Schematic illustration of the proposed pipeline

3.1 Connection Features

We build on a sequential feature paradigm presented recently in MalPaCA [33]: a behavior discovery framework for network traffic which uses Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBScan) [6], providing clusters of connection sequences.

From the original packet capture (.pcap file), we define dataframe C which is a matrix with $t \times p$ dimensions, with t rows (one row for every packet) and p features on the columns. C was made to include unidirectional connections, defined as an uninterrupted list of all packets sent from a source IP to destination IP. MalPaCA proposed to include four sequential features: packet size (bytes), time interval (gaps), source port (sport), and destination port (dport).

From every column of C we created the symmetric distance matrices D_{bytes}, D_{gaps}, D_{sport}, and D_{dport}. All distance matrices had $n_c \times n_c$ dimensions, with n_c unique unidirectional connections, and zero diagonals. For D_{bytes} and D_{gaps} the pairwise distance over time (t) was calculated via Dynamic Time Warping (DTW). For each pair of hosts we had time series $X \in \{1, ..., N\}$ and $Y \in \{1, ..., M\}$ and the average accumulated difference between X and Y is

$$d_\phi(X, Y) = \sum_{k=1}^{T} \frac{d(\phi_x(k), \phi_y(k)) m_\phi(k)}{M_\phi} \tag{1}$$

with warping functions: $\phi(k) = (\phi_x(k), \phi_y(k))$, $\phi_x(k) \in \{1...N\}$, $\phi_y(k) \in \{1...M\}$, which shape the warping curve $\phi(k); k \in \{1, ..., T\}$. $m_\phi(k)$ is a weighting coefficient and M_ϕ is the corresponding normalization constant, which ensures that the accumulated differences in time series are comparable along different paths [14]. DTW optimises by finding the minimum the difference: $dtw(X, Y) = \arg\min_{\phi} d_{\phi}(X, Y)$ and we normalized the DTW estimates to range [0–1] with

$$\hat{x}_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{2}$$

where x = $[dtw(X_1, Y_1)$, $dtw(X_1, Y_2)$, ..., $dtw(X_{n_c}, Y_{n_c})]$.

For source and destination port, the pairwise distances were calculated with the cosine similarity

$$cos(X, Y) = \frac{\sum_{k=1}^{T}(X_k * Y_k)}{\sqrt(\sum_{k=1}^{T}(X_k^2))\sqrt(\sum_{k=1}^{T}(Y_k^2))} \tag{3}$$

which were normalized as described to form D_{sport} and D_{dport}.

3.2 Host Features

The Stochastic Block Model (SBM) required to transform the connection distance matrices (D_{bytes}, D_{gaps}, D_{sport} and D_{dport}) to host distance matrices, which was achieved via Principal Component Analyses (PCA). The PCA works by calculating the singular value decomposition of the distance matrices so that by maximizing the variation captured per component a small number of components (ideally) captures a major proportion of the variation. We input the distance matrices so the aim was to acquire a number of dimensions less than the number of unique connections, accomplished by selecting the m components explaining at least 40% cumulative variation. For each of the 4 features, the PCA thus resulted in a matrix W with n_c rows and m columns, so that for each unique $a \rightarrow b$ connection m, component weights were available. We used W to create m host-host SBM covariates. Since every row of W referred to a unique $a \rightarrow b$ connection, the connection source (a) and destination (b) are used to indicate the rows and columns for each SBM covariate matrix Y_m with dimensions $n_h \times n_h$ where n_h is the unique number of hosts. Hence, the values in Y_{bytes,m_1}, the SBM covariate matrix for the first component of *bytes*, were inherited from m_1 of W_{bytes} (see Table 1).

Table 1. A fictional example of a distance matrix D_{bytes}, PCA component weights matrix W_{bytes}, and corresponding SBM covariate matrix Y_{bytes,m_1}.

D	ab	ac	bc	ca
ab	0	689	1262	512
ac	689	0	1169	680
bc	1262	1169	0	1062
ca	512	680	1062	0

W	m_1
ab	−3.18
ac	−2.96
bc	−4.60
ca	−2.92

Y_{m_1}	a	b	c
a	0	−3.18	−2.96
b	0	0	−4.60
c	-2.92	0	0

3.3 Stochastic Block Model

The SBM took as input a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ was the node set of size $n_h := |\mathcal{V}|$, and $\mathcal{E}$ was the edge list of size $M := |\mathcal{E}|$. The corresponding $n_h \times n_h$ adjacency matrix was denoted by Y, where $Y_{ab} = 1$ if there was a connection between hosts a and b and 0 otherwise. The main input graph was an undirected binary node matrix Y_{class} which held a 1 if there was any connection between nodes a and $b; Y_{class,ab} = 1$ or zero otherwise. The generated SBM covariate matrices are added to the model as covariates

$$SBM(Y_{class,ab}, List(Y_{packetSize,m}, Y_{gapsDist,m}, Y_{sourcePort,m}, Y_{destPort,m}))$$

Since group (g) membership is unknown, the membership labels for every host are captured by a latent variable Z_a, which elements are all 0, except exactly one that takes the value 1 and represents the group host a belongs to. This Z_a is assumed to be independent of Z_b for $a \neq b$. Finally, SBM outputs a $n \times g$ matrix $Z := (Z_1, ..., Z_n)^T$, such that $Z_{a,i}$ is the i^{th} element of Z_a. Graph generation and likelihood are explained elsewhere [27]. The lower and upper bound of fitted SBM models were 2 and 10. Model fit was evaluated with the Integrated Classification Likelihood (ICL), via a variational expectation maximization approach implemented in R [28].

3.4 Experimental Setup

This study used data from the Malware Capture Facility Project, which is a sister project of the Stratosphere IPS Project: an initiative to obtain malware and normal data. From all the published samples, a dataset was selected which included both normal $(N_b = 12)$ and infected $(N_i = 10)$ hosts and included the entire network. The malicious hosts were infected with the Conficker botnet. The data were downloaded from https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-91/ as a .pcap file consisting of 198818 lines (packets), capturing 1011 unique $(a \rightarrow b)$ connections. There were 3 isolated clusters which were removed, leaving 917 unique connections. The correlation between covariates was low (see Table 7) so instead of combining the distance matrices they were included in the SBM as individual predictors.

Not all observed connections are necessarily informative, so we experimented with a minimum number of packets-threshold (P_t) to ensure that the remaining connections represented sufficient information for effective behavioral modeling. The thresholds tested were $P_t \in \{5, 10, 15, 20\}$, respectively pruning to 631 (62.4%), 565 (55.9%), 523 (51.7%), and 483 (47.8%) connections (see Table 2). From analyses we determined that for this dataset a packet threshold of 10 is desirable, balancing the number of connections, nodes, MalPaCA and SBM clusters (see Supplementary Material). Higher thresholds resulted in too much pruning of the network structure, hindering accurate classification in this dataset.

Table 2. Descriptives of the Stratosphere CTU-91 data with different behavioral thresholds

Covariate	N_{seq}	N_{ip}	$Q_{MalPaCA}$	*outliers*	Q_{SBM}
5 packets	631	205	10	120	4
10 packets	565	182	9	154	4
15 packets	523	165	7	40	4
20 packets	483	148	6	38	5

This Table presents the number of unique $a \rightarrow b$ sequences (N_{seq}), unique hosts (N_{ip}), the optimal number of clusters ($Q_{MalPaCA}$) and *outliers* determined by MalPaCA, and optimal SBM-cluster solution (Q_{SBM}).

3.5 Replication Sample

For replication of our main finding we used the ISOT dataset from the University of Victoria (https://www.uvic.ca/engineering/ece/isot/datasets) as presented in [38], which included of a collection of neutral/background data and 4 samples (Waledac, Storm, Zeus) of botnet data. Storm, Waledac, and Zeus are Windows targeting botnets predominantly used in spamming campaigns which peaked in 2007–2008. They can all be managed via a Command and Control as well as Peer to Peer communication. From the neutral data we selected the data from the Traffic Lab at Ericsson Research in Hungary [43]. The latter contained a large number of general traffic from a variety of applications, including HTTP web browsing behavior, World of Warcraft gaming packets, and packets from popular bittorrent clients. ISOT documentation states IP addresses of infected machines were mapped to the background traffic and all trace file were replaced to homogenize network behavior. The infected data contained 747264 packets with 25308 unique connections and the Ericsson lab data included 2300385 packets from 12778 unique connections. These two sets were combined so that MalPaCA features could be extracted.

4 Results

4.1 Stratosphere Data

MalPaCA Directly. Applying MalPaCA directly to the data assigned the connections to 9 dense clusters (see Table 3). Visual inspection of the nodes belonging to the connections classified as outliers revealed that these were mostly peripheral, supporting the notion that nodes on the edges of the network, with negligible activity, are more likely to fall outside a MalPaCA cluster.

Different subsets of connections were identified. Cluster 1 captured all traffic from 192.168.0.118 to peripheral hosts. Cluster 3 included bidirectional traffic

Table 3. MalPaCA clusters and infection status in the CTU-91 data. Connections in −1 are unclustered. $srcip_p, srcip_n, scrip_i$ are connections where the source host was peripheral, normal, or infected (respectively). The same for destination ports *dstip*.

Cluster	$srcip_p$	$srcip_n$	$srcip_i$	$dstip_p$	$dstip_n$	$dstip_i$
−1	8	6	23	17	10	10
1	0	0	14	14	0	0
2	10	0	0	0	0	10
3	119	0	0	0	0	119
4	62	0	0	0	0	62
5	0	0	125	125	0	0
6	0	12	78	73	10	7
7	0	4	4	0	0	8
8	0	0	8	0	8	0
9	0	10	0	0	0	10

between normal and infected hosts as well as connections from normal to normal, infected, and peripheral hosts. Clusters 4 and 5 included connections from normal and infected to peripheral hosts (opposite to cluster 2: peripheral to infected and normal), but apparently specific clusters were required to capture specific connections from peripheral to infected (clusters 6 and 7) and infected to peripheral hosts (clusters 8 and 9), illustrating the heterogeneity in connections from and to infected nodes. Relating the connections to their respective nodes, we identified 11 true negatives (cluster 1), 11 false positives (clusters 2:5), and 389 true positives, yielding an accuracy of 97.32%, sensitivity of 100% and specificity of 50%.

SBM Directly. Fitting the SBM directly on the network matrix, ignoring the MalPaCA features, resulted in a 6-class solution. This solution was incapable of distinguishing normal and peripheral nodes (as described earlier in [37]). Class 1 and 3 captured 11 peripheral and 2 normal hosts, class 2 and 5 respectively captured 2 and 3 infected hosts, class 4 included 3 normal and 5 infected hosts, and class 6 only included 148 peripheral hosts. Hence, there are 10 true positives, 3 false positives (class 4), and 312 true negatives, resulting in a performance of: accuracy = 99.08%, sensitivity = 100% and specificity = 99.05%.

Our Approach. Applying MalPaCA to obtain the distance matrices, representing the distances between connections for the four features, resulted in 565 surviving connections. The average connection length was 348.48, with a minimum of 10 packets ($P_t = 10$) and a maximum of 5333. The PCA solution on the MalPaCA distance matrices commended a 1 (bytesDist), 3 (destPort), 1 (gapsDist), and 3 (sourcePort) component solution that cumulatively explained >40% of the variation. This result was P_t invariant; including more packets per connection does not change the amount of variation explained by the components.

Fitting the SBM on the PCA derived covariates favoured a 4-class solution. The network with original- and cluster labels is visualized in Fig. 2 and the performance matrix for the 10 threshold solution is provided in Table 4. After obtaining the cluster solution we used straightforward descriptive analyses and visualization to interpret the clusters (see Supplementary Material and [33]). We found that all malicious hosts were assigned to one cluster with a posterior probability of >.998. Most of the peripheral hosts were captured by one cluster, indicating behavioral similarity, with a class assignment posterior probability of .9982. The non-infected/normal hosts were divided over two clusters, that also included peripheral hosts. Only one normal host had a posterior probability <.95, which was host 192.168.1.6 with .82, with the remaining probability belonging to the other *normal/mixed* class. If we consider all peripheral hosts $(136 + 9 + 1)$ and normal hosts $(4+3)$ to be true negatives, and the correctly clustered infected hosts as true positives, the classification is perfect. These findings are consistent for all four tested packet thresholds (P_t).

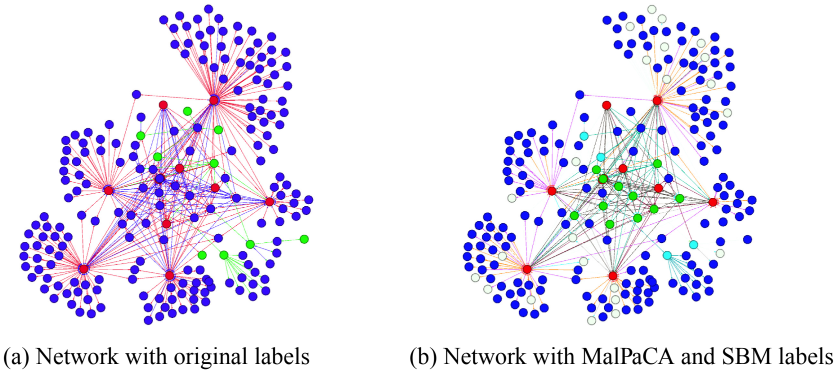
Table 4. Performance matrix from the SBM node-based clustering in the CTU-91 data

Cluster	*Peripheral*	*Normal*	*Infected*
1	–	–	10
2	136	4	–
3	9	–	–
4	1	3	–

Table 5. Performance comparison with other studies using ISOT data

Method	*Accuracy*	*Sensitivity*	*Specificity*	*Study*
BClus	.5	.4	.5	[12]
CAMNEP	.5	0	.9	[12]
BotHunter	.4	.01	.9	[12]
BotGM	.91	.83	n.p	[26]
Decision tree	.99	.98	n.p	[51]
Decision tree	.75	.99	n.p	[3]

n.p. = not provided


(a) Network with original labels

(b) Network with MalPaCA and SBM labels

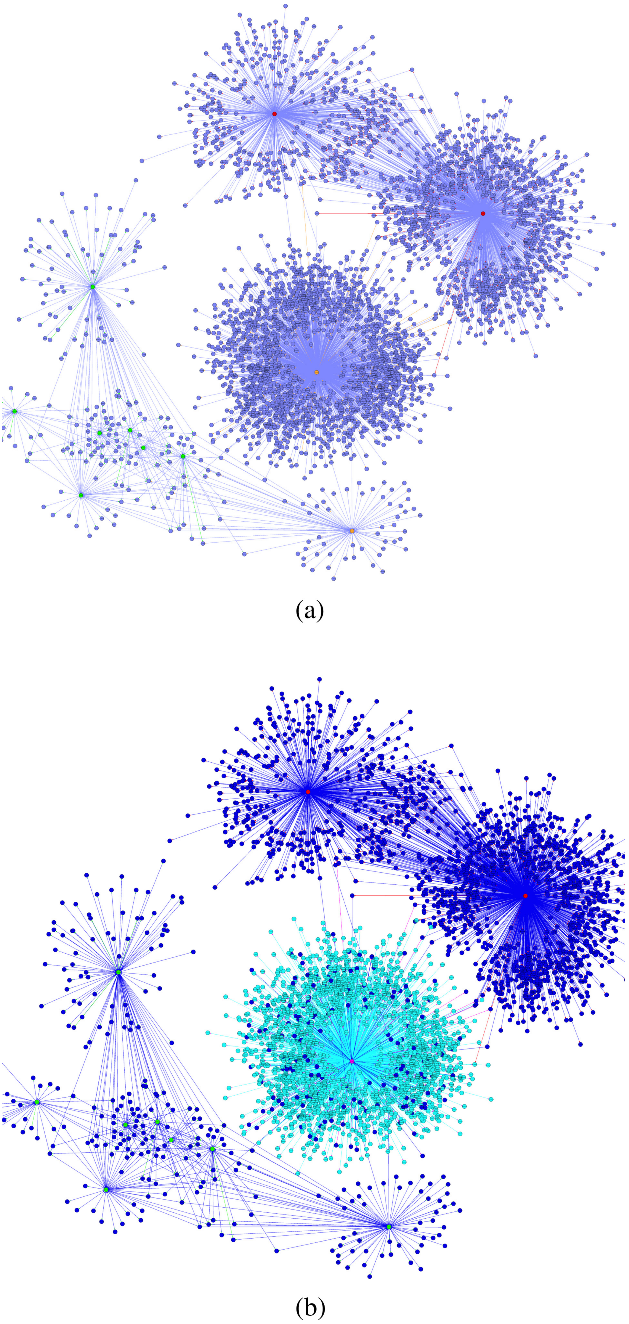
Fig. 2. Network plots of a subset of the CTU-91 network (including hosts with a packet threshold $P_t = 10$). Left: network with original host labels, used in this analyses as ground truth (blue = peripheral, red = infected, green = normal). Right: network with the MalPaCA connection label colours and SBM host labels (blue = peripheral, red = infected, green & turquoise = normal & peripheral). (Color figure online)

4.2 ISOT Data

Previous studies have used the ISOT data for botnet identification purposes and Table 5 presents a selection of the performance reported in related works. As mentioned before, most of these methods require manual feature engineering and connection filtering to be applied, while others operate in a supervised setting. We compare our unsupervised clustering method to these results.

Creating the distance matrices with MalPaCA pruned the network (see Fig. 3a) to 7683 surviving connections with $P_t = 20$. Average connection length was 365.95, with a minimum of 20 and a maximum of 525256. This amounted to 3847 nodes. There was one isolated sub-network of hosts connected to 172.16.2.3, of which only the connection between 172.16.2.3 and 193.88.8.59 survived the packet threshold of 20. Isolation supported their removal from subsequent clustering analyses, leaving 3845 nodes (running the analyses with these two nodes included yielded similar results in the optimal SBM solution; both were allocated to the cluster with infected nodes).

Identical to the Stratosphere data, a PCA fitting resulted 1, 1, 3, 3 components for respectively bytes, gaps, dport and sport to explain >40% of the variation. The SBM model fitted on the binary adjacency matrix, with the PCA features resulted in an optimal 5 class solution (see Figs. 3b and Table 6). Of these 5 clusters, clusters 1 and 2 captured the peripheral nodes, where the peripheral nodes in cluster 1 were all linked to host 172.16.2.11 (Storm + non-malicious) which was the only host allocated to cluster 3. Cluster 4 consisted of the Waledac and Storm hosts, confirming the comparability of Waledac and Storm activity. Cluster 5 captures eight hosts, of which seven are non-malicious: 172.16.2.2, 172.16.2.13-14, 172.16.2.111-114, and one host in cluster 5 (172.16.2.12) had combined (non-malicious & malicious) traffic. If we consider 1734 and 2100 peripheral



(a)

(b)

Fig. 3. (a) Network plots of a subset of the ISOT network for $P_t = 20$. Network with original host labels, used here as ground truth (blue = peripheral, red = malicious, orange = malicious + non-malicious, green = non-malicious. (b) Network with labels assigned by our method: Turquoise (cluster 1) & blue (cluster 4) = peripheral, red (cluster 2) = malicious + misclasification, orange (cluster 3) and purple (cluster 5) = Waledac, and green (cluster 6) = non-malicious. (Color figure online) nodes (cluster 1 and 2) and 7 non-malicious nodes (cluster 5) as true negatives, the Waledac and Storm nodes in cluster 3 and 4 as true positives, and the combined traffic node in cluster 5 as a false negative, the accuracy and sensitivity = 99.97 % and the specificity = 100%. This performance is similar to other work on supervised learning using decision trees [26,51] and nearest neighbours [13] on manually curated collapsed data. We outperform the methods listed in [12].

Table 6. Performance matrix from the SBM node-based clustering in the ISOT replication data

Cluster	$Peripheral$	$Normal$	$Normal + infected$	$Infected$
1	1734	–	–	–
2	2100	–	–	–
3	–	–	1	–
4	–	–	–	2
5	–	7	1	–

5 Discussion

Here, we combined two unsupervised methods to solve the problem of analysing spatio-temporal data so that botnet infected computers can be identified via connection- and host clustering. In our discovery sample (CTU-91) we identified all infected machines and classification was perfect. The infected machines were all allocated to one cluster, indicating marked similarities between infected machines infected with the Conficker botnet. In the replication sample (ISOT), one host with malicious and non-malicious traffic was allocated to a cluster of non-malicious nodes, yielding one false negative with an overall accuracy of 99.97%. This procedure outperforms other botnet detection studies using the ISOT dataset [3,8,26,38,47] and has comparable performance to [13,36]. Compared to the studies that report similar classification performance, our method does not require any type of filtering [36] or manual feature selection [13], and is therefore less sensitive to external factors. In the discovery sample, the normal and peripheral hosts were allocated together in a cluster, whereas in the replication data, the peripheral hosts formed a separate cluster. This may be due to the mapping procedure used in the ISOT dataset, where botnet data were collected in a VM and mapped a posteriori, so that the differences in the ISOT data may be captured by our model, underlining the sensitivity of our approach. Furthermore, although not explicitly illustrated, the output of MalPaCA has been found to be informative to identify malware families or other specifically tuned categories of traffic [33], and other similar connection profile based approaches exist [36].

A potential limitation of this study is the relatively short time window in which the data were collected. Ideally one would capture the temporal structure of the network traffic in more specific analyses. A prominent example of such analyses is creating snapshots [23], which facilitates network clustering within snapshots, so that state changes (nodes hopping to another cluster) between snapshots can be analysed [31]. However, given the length of the CTU-91 capture (roughly 20 min, compared to for example one year of data from mobile devices in [31]) we argue there is little sense in making 5-min snapshots, since this would result in many, difficult to compare, local network clusters. Again, these packet thresholds are data specific, and shorter or other snapshots may be applicable in other types of network data (e.g. social network data where snapshots represent school-years). Although our approach does not require manual curation, understanding the effects of sample specific factors is a focus of future research. Another limitation of this approach is the speed of Variational Inference when fitting a SBM with covariates to large datasets (>2500 nodes). The runtime of our discovery (CTU-91) sample was about 2.5 h on a Windows 10 (i7-7700K CPU, 4.2 GHZ, 8-core, 16 GB ram) machine, but new developments in fast optimization [35] will reduce run-time from hours to minutes.

6 Conclusion

The overarching aim of this study is to present a combination of clustering methods to simultaneously cluster nodes and host in spatial-temporal network data, where the features capture a sequential or time series structure. In the setting of botnet detection, our method is able to allocate labels to distinguish different types of nodes, with near-perfect classification, while ingesting raw unfiltered network traffic data. This makes it an easy-to-use and effective tool for network traffic analysis.

Our method and results add to existing studies that botnets are relatively easy to detect. Indeed, our performance is higher compared to most earlier studies and we depend less on manual curating of the data, but methods solely based on community detection, or collapse temporal variation in composite features yield excellent results as well.

In future work, we aim to make our approach less computationally intensive using sketching and related methods from data-stream mining. Moreover, we afterwards intend to apply it to large network captures, and simplify the connection-to-host transformation to a PCA independent yet robust implementation.

7 Supplementary Material

Location of R-scripts and raw input file: https://drive.google.com/drive/folders/121pnmgob-f-T0lE60yQmFlnnq6MW2VQy?usp=sharing (Fig. 4).

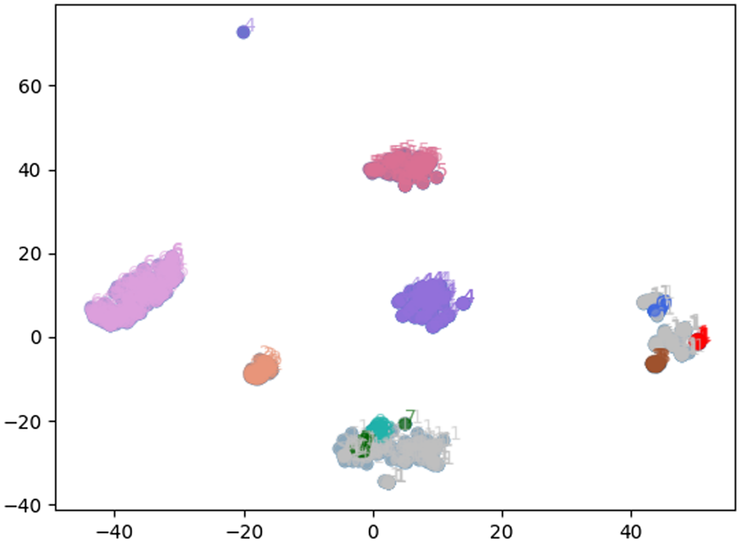
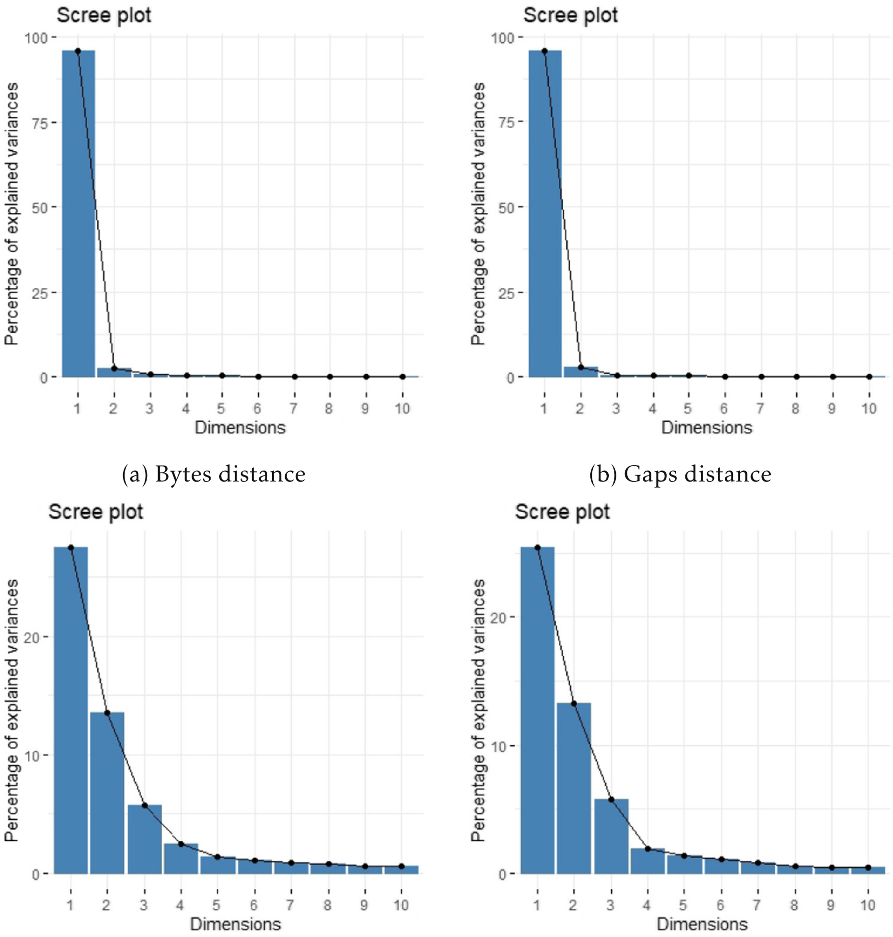


Fig. 4. This Figure shows the connections clustered with MalPaCA on the CTU-91 data. The grey dots indicate connections labeled as outliers by HDBScan. For this plot, the multidimensional sample space was reduced to two axes by TSNE, resulting in the ability to visually identify 7 clusters, of which the top cluster belongs to the middle cluster (letter 4), the right cluster decomposes into 3 sub-clusters (blue, red, brown) and outliers, and the bottom cluster consist of 2 sub-clusters (magenta, darkgreen) and outliers. Hence, 9 clusters are displayed. (Color figure online)

7.1 Host Clustering CTU-91 Dataset

Node assignment to a cluster does not immediately inform which cluster(s) contain the infected nodes. Descriptive analyses are typically used to interpret the cluster output. For example, when comparing cluster 1 (10 hosts) with cluster 2 (140 hosts), we observed an almost 3-fold increase of packets send (93100 versus 33917), a higher occurrence of bigger packets send ($Mean_{c1} = 138.22 (SD = 180.51), Mean_{c2} = 118.97 (SD = 135.63), t = 1.9547, p = .051$) and received ($Mean_{c1} = 167.26 (SD = 226.31), Mean_{c2} = 142.92 (SD = 194.23), t = 1.6614, p = .09703$), and higher frequencies of HTTPS, UDP, and SMTP/IMF protocol traffic, whereas SMTP, TCP, NBNS, and BROWSER protocol traffic was significantly higher in cluster 2. This behavior of nodes (more connections via specific protocols) is coherent for botnets. Further visualisation (not provided) resulted in the identification of cluster 1 as likely malicious (and verified with the original labels). All of the malicious hosts (192.168.1.238, 192.168.1.239, 192.168.1.236, 192.168.1.91, 192.168.1.71, 192.168.1.9, 192.168.1.243, 192.168.1.242, 192.168.1.247, 192.168.1.245) were assigned to one cluster with a posterior probability of $>.998$ (Figs. 5 and 6).



(a) Bytes distance

(b) Gaps distance

(c) Destination port distance

(d) Source port distance

Fig. 5. CTU-91 data: Explained variance of components from the Principal Component Analysis on the four distance matrices, where the packet threshold was 10 packets. The connection distances in the bytes and gaps matrices were captured by one component approximately explaining 90% of the variance, whereas 3 components were required to capture >40% of the variance in the destination and source port distances.

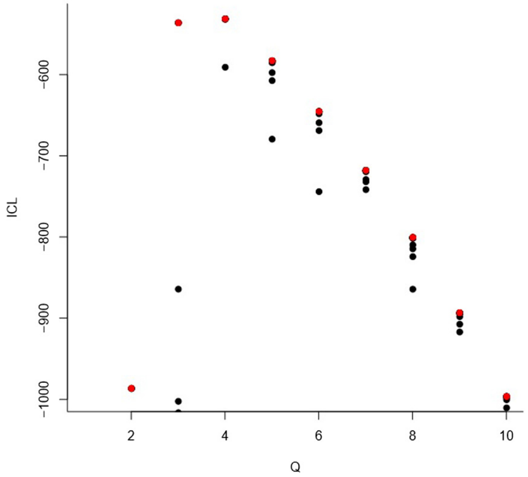


Fig. 6. Plots of the ICL fit evaluation statistic based on the CTU-91 data. The peak at $Q = 4$ illustrates that the optimal SBM clustering solution is reached at 4-classes, and model fit decays when Q increases.

Table 7. Correlation between distance matrices in the CTU-91 data

	bytes	gaps	dport	sport
bytes	–			
gaps	.04	–		
dport	.13	.09	–	
sport	.05	−.03	−.04	–

Our observation that mean differences between clusters (as exampled above) show a trend but are not significant, illustrates that just comparing mean differences to detect groups, with a straightforward anomaly detection approach, would be less successful in this particular setting (Fig. 8).

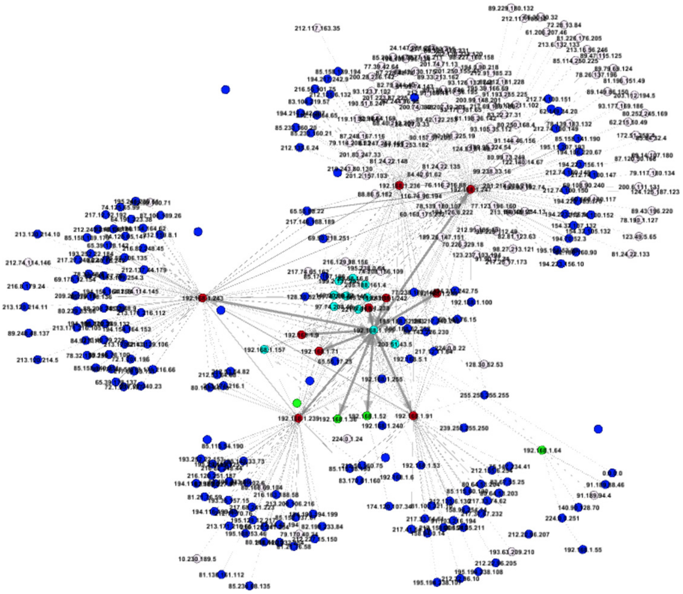


Fig. 7. This Figure shows the full network with the nodes coloured according to the labels from the optimal 4-class SBM solution. This plot is based on the analyses of 483 connections and 148 hosts (nodes) with packet threshold = 20. Nodes are coloured blue (normal), green (normal), turquoise (normal), red (infected), or white (outliers). (Color figure online)

Most of the peripheral hosts were captured by one cluster, indicating behavioral similarity, with a class assignment posterior probability of .9982. The non-infected/normal hosts (192.168.1.155, 192.168.1.52, 192.168.1.157, 192.168.1.36, 192.168.1.6, 192.168.1.53, 192.168.1.64) were divided over two clusters, that also included peripheral hosts. Only one normal host had a posterior probability <.95, which was host 192.168.1.6 with .82, with the remaining probability belonging to the other *normal/mixed* class (Figs. 9, 10, 7, 11, 12 and 13 (Tables 8, 9, 10 and 11).

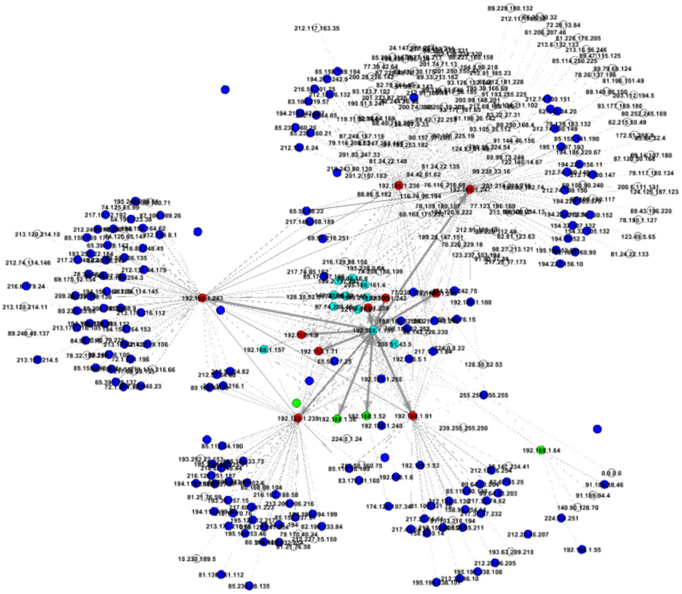


Fig. 8. This Figure shows the full network with the nodes coloured according to the labels from the optimal 4-class SBM solution. This plot is based on the analyses of 631 connections and 205 hosts (nodes) with packet threshold = 5. Nodes are coloured blue (normal), green (normal), turquoise (normal), red (infected), or white (outliers). (Color figure online)

Table 8. Performance matrix from the SBM node-based clustering when packet threshold = 5

Cluster	*Peripheral*	*Normal*	*Infected*
1	9	0	0
2	0	0	10
3	1	4	0
4	158	3	0

Table 9. Performance matrix from the SBM node-based clustering when packet threshold = 15

Cluster	*Peripheral*	*Normal*	*Infected*
1	133	4	0
2	3	1	10
3	0	1	0

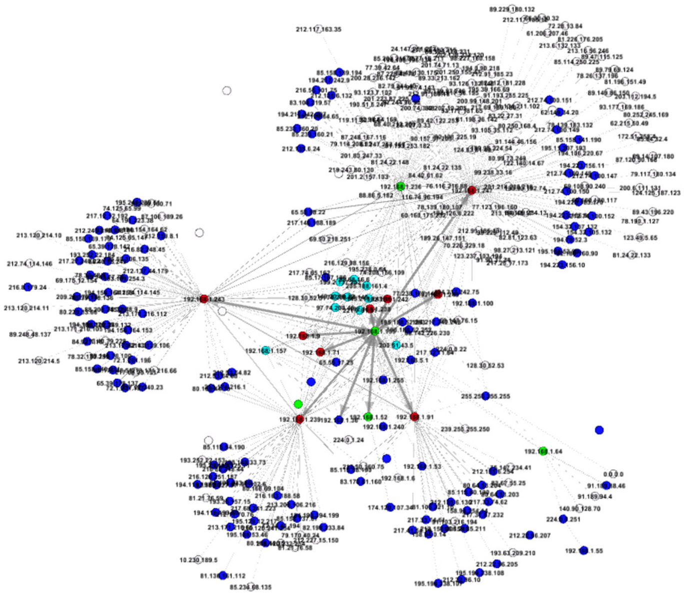


Fig. 9. This Figure shows the full network with the nodes coloured according to the labels from the optimal 4-class SBM solution. This plot is based on the analyses of 565 connections and 182 hosts (nodes) with packet threshold = 10. Nodes are coloured blue (normal), green (normal), turquoise (normal), red (infected), or white (outliers). (Color figure online)

Table 10. Performance matrix from the SBM node-based clustering when packet threshold = 20

Cluster	*Peripheral*	*Normal*	*Infected*
1	123	5	0
2	0	0	6
3	0	0	4
4	2	1	0

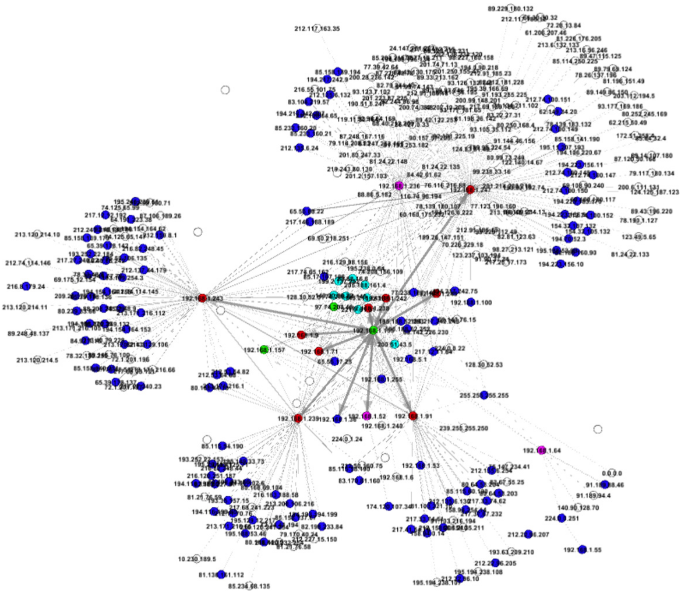
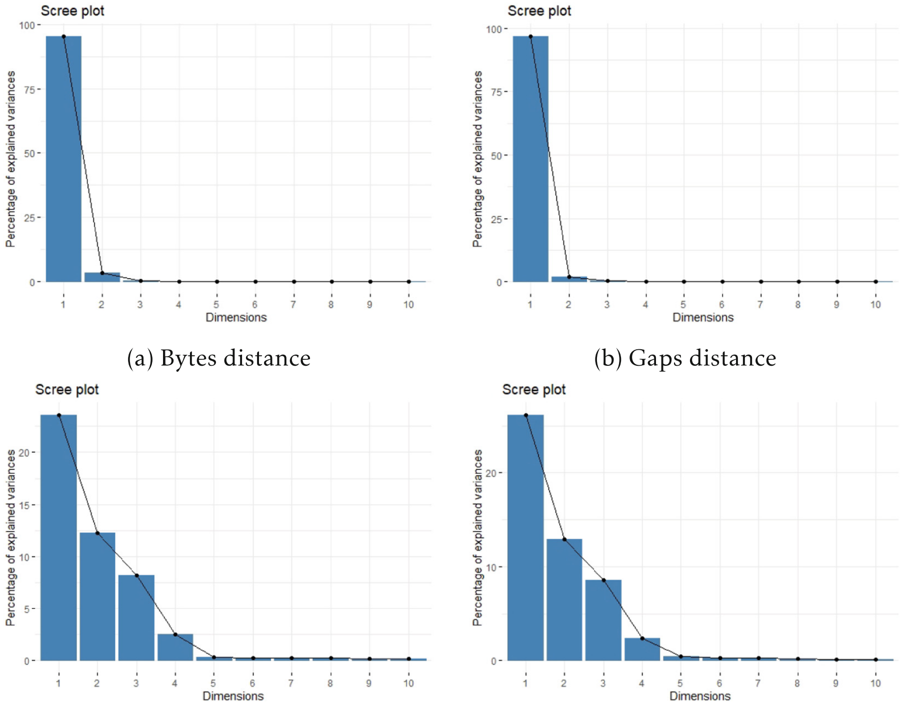


Fig. 10. This Figure shows the full network with the nodes coloured according to the labels from the optimal 4-class SBM solution. This plot is based on the analyses of 523 connections and 165 hosts (nodes) with packet threshold = 15. Nodes are coloured blue (normal), green (normal), turquoise (normal), red (infected), or white (outliers). (Color figure online)



(a) Bytes distance

(b) Gaps distance

(c) Destination port distance

(d) Source port distance

Fig. 11. ISOT data: Explained variance of components from the Principal Component Analysis on the four distance matrices, where the packet threshold was 5 packets. The connection distances in the bytes and gaps matrices were captured by one component approximately explaining 90% of the variance, whereas 3 components were required to capture >40% of the variance in the destination and source port distances.

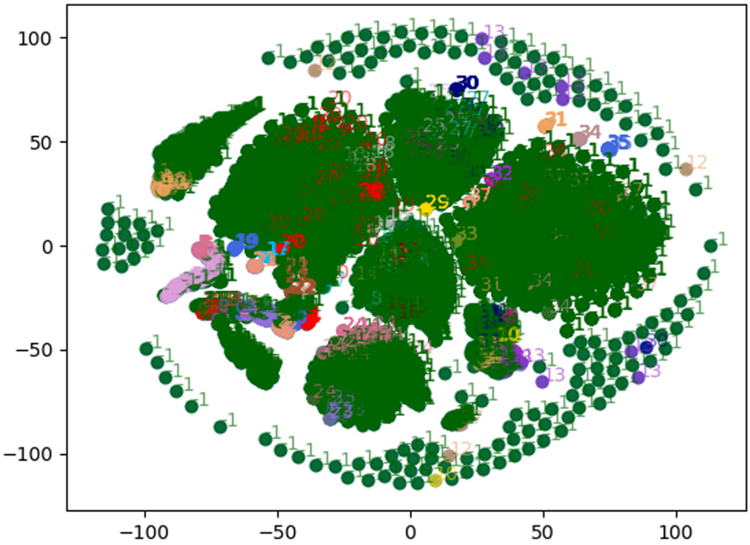


Fig. 12. This Figure shows the connections clustered with MalPaCA on the ISOT data. The green dots indicate connections labeled as outliers by HDBScan. For this plot, the multidimensional sample space was reduced to two axes by TSNE. By colour we different clusters (e.g. orange and purple). Compared to the CTU-91 dataset we see the connections occupy a larger sample space, indicating more variance in the ISOT replication data. (Color figure online)

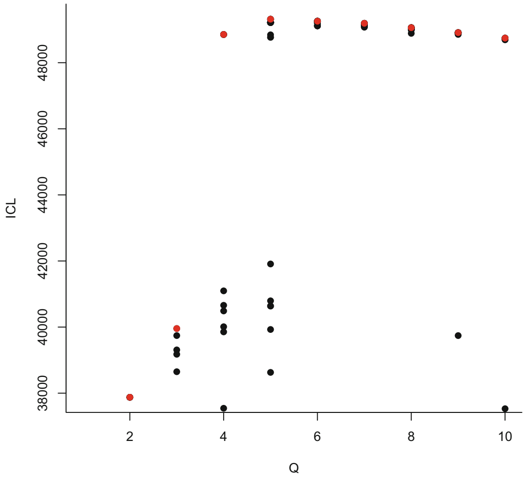


Fig. 13. Plots of the ICL fit evaluation statistic in the ISOT data. The subtle peak at $Q = 5$ indicates that the optimal SBM clustering solution is reached at 5-clusters, and model fit decays when Q increases.

Table 11. MalPaCA clusters and infection status in the ISOT data

Cluster	$srcip_n$	$srcip_i$	$dstip_n$	$dstip_i$
-1	1948	3415	1703	3216
1	0	0	9	10
2	12	12	0	0
3	21	0	0	0
4	0	0	24	0
5	22	0	0	0
6	0	0	20	6
7	0	0	90	17
8	92	10	0	0
9	0	0	0	10
10	0	16	0	0
11	0	9	0	0
12	0	43	0	0
13	0	48	0	0
14	0	38	0	0
15	0	0	0	10
16	0	0	0	11
17	0	0	0	22
18	0	0	0	8
19	0	0	0	8
20	0	0	0	10
21	0	0	0	49
22	0	0	0	10
23	0	0	0	27
24	0	0	0	7
25	0	0	0	40
26	0	7	0	0
27	0	7	0	0
28	0	11	0	0
29	0	7	0	0
30	0	4	0	4
31	0	27	0	0

(*continued*)

Table 11. (*continued*)

Cluster	$srcip_n$	$srcip_i$	$dstip_n$	$dstip_i$
32	11	11	0	0
33	8	8	0	0
34	8	8	0	0
35	11	11	0	0
36	11	11	0	0
37	8	8	0	0
38	15	15	0	0

Interpretation of rows and columns equal to Table 3. Clusters 1, 6, and 7 contain connections from peripheral hosts to normal and infected hosts. Clusters 2, 8, 32–38 contain connections from both infected and normal host to peripheral nodes. Clusters 3 and 5 both include connections from a normal source ip to a peripheral nodes. Cluster 9 includes connections from peripheral nodes to infected destination hosts. Clusters 10–14, 26–29, and 31 comprise of connections from infected source hosts to peripheral hosts. Cluster 30 includes connections from infected source IPs to infected destination IPs.

References

1. Abbe, E.: Community detection and stochastic block models: recent developments. J. Mach. Learn. Res. **18**(1), 6446–6531 (2017)
2. Barthakur, P., Dahal, M., Ghose, M.K.: A framework for P2P botnet detection using SVM. In: 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 195–200 (2012)
3. Beigi, E.B., Jazi, H.H., Stakhanova, N., Ghorbani, A.A.: Towards effective feature selection in machine learning-based botnet detection approaches. In: 2014 IEEE Conference on Communications and Network Security (CNS), pp. 247–255 (2014)
4. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech.: Theory Exp. **2008**(10), P10008 (2008)
5. Cai, T., Zou, F.: Detecting HTTP botnet with clustering network traffic. In: 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–7 (2012)
6. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7819, pp. 160–172. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37456-2_14

7. Carl, L., et al.: Using machine learning techniques to identify botnet traffic. In: Proceedings of the 31st IEEE Conference on Local Computer Networks. IEEE (2006)
8. Chowdhury, S., et al.: Botnet detection using graph-based feature clustering. J. Big Data **4**(1), 14 (2017). https://doi.org/10.1186/s40537-017-0074-7
9. Coskun, B., Dietrich, S., Memon, N.: Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts. In: Proceedings of the 26th Annual Computer Security Applications Conference, pp. 131–140 (2010)
10. Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
11. Feizollah, A., Anuar, N.B., Salleh, R., Amalina, F., Shamshirband, S., et al.: A study of machine learning classifiers for anomaly-based mobile botnet detection. Malays. J. Comput. Sci. **26**(4), 251–265 (2013)
12. Garcia, S., Grill, M., Stiborek, J., Zunino, A.: An empirical comparison of botnet detection methods. Comput. Secur. **45**, 100–123 (2014)
13. Garg, S., Singh, A.K., Sarje, A.K., Peddoju, S.K.: Behaviour analysis of machine learning algorithms for detecting P2P botnets. In: 2013 15th International Conference on Advanced Computing Technologies (ICACT), pp. 1–4 (2013)
14. Giorgino, T., et al.: Computing and visualizing dynamic time warping alignments in R: the DTW package. J. Stat. Softw. **31**(7), 1–24 (2009)
15. Gu, G., Perdisci, R., Zhang, J., Lee, W.: BotMiner: clustering analysis of network traffic for protocol-and structure-independent botnet detection (2008)
16. Gu, G., Zhang, J., Lee, W.: BotSniffer: detecting botnet command and control channels in network traffic (2008)
17. Haddadi, F., Morgan, J., Gomes Filho, E., Zincir-Heywood, A.N.: Botnet behaviour analysis using IP flows: with HTTP filters using classifiers. In: 2014 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 7–12 (2014)
18. Handcock, M.S., et al.: Temporal exponential random graph models (TERGMs) for dynamic network modeling in statnet. In: Sunbelt 2015 (2015)
19. Hyvarinen, A., Morioka, H.: Unsupervised feature extraction by time contrastive learning and nonlinear ICA. In: Advances in Neural Information Processing Systems, pp. 3765–3773 (2016)
20. Ioannidis, J.P.A.: Why most published research findings are false. PLos Med. **2**(8), e124 (2005)
21. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. (CSUR) **31**(3), 264–323 (1999)
22. Jung, T., Wickrama, K.A.S.: An introduction to latent class growth analysis and growth mixture modeling. Soc. Pers. Psychol. Compass **2**(1), 302–317 (2008)
23. Kostakis, O., Tatti, N., Gionis, A.: Discovering recurring activity in temporal networks. Data Min. Knowl. Discov. **31**(6), 1840–1871 (2017). https://doi.org/10.1007/s10618-017-0515-0
24. Kostakos, V.: Temporal graphs. Phys. A: Stat. Mech. Appl. **388**(6), 1007–1023 (2009)
25. Kumar, V., Dhok, S.B., Tripathi, R., Tiwari, S.: A review study of hierarchical clustering algorithms for wireless sensor networks. Int. J. Comput. Sci. Issues (IJCSI) **11**(3), 92 (2014)
26. Lagraa, S., François, J., Lahmadi, A., Miner, M., Hammerschmidt, C., State, R.: BotGM: unsupervised graph mining to detect botnets in traffic flows. In: 2017 1st Cyber Security in Networking Conference (CSNet), pp. 1–8 (2017)

27. Lee, C., Wilkinson, D.J.: A review of stochastic block models and extensions for graph clustering. arXiv preprint arXiv:1903.00114 (2019)
28. Leger, J.-B.: Blockmodels: a R-package for estimating in latent block model and stochastic block model, with various probability functions, with or without covariates. arXiv preprint arXiv:1602.07587 (2016)
29. Liu, F., Li, Z., Nie, Q.: A new method of P2P traffic identification based on support vector machine at the host level. In: 2009 International Conference on Information Technology and Computer Science, pp. 579–582 (2009)
30. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
31. Masuda, N., Holme, P.: Detecting sequences of system states in temporal networks. Sci. Rep. **9**(1), 1–11 (2019)
32. Mossel, E., Neeman, J., Sly, A.: Stochastic block models and reconstruction. arXiv preprint arXiv:1202.1499 (2012)
33. Nadeem, A., Hammerschmidt, C., Gañán, C.H., Verwer, S.: MalPaCA: malware packet sequence clustering and analysis. arXiv preprint arXiv:1904.01371 (2019)
34. Nagaraja, S., Mittal, P., Hong, C.-Y., Caesar, M., Borisov, N.: BotGrep: finding P2P bots with structured graph analysis. In: USENIX Security Symposium, pp. 95–110 (2010)
35. Park, Y., Bader, J.S.: Fast and reliable inference algorithm for hierarchical stochastic block models. arXiv preprint arXiv:1711.05150 (2017)
36. Rahbarinia, B., Perdisci, R., Lanzi, A., Li, K.: PeerRush: mining for unwanted P2P traffic. In: Rieck, K., Stewin, P., Seifert, J.-P. (eds.) DIMVA 2013. LNCS, vol. 7967, pp. 62–82. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39235-1_4
37. Roeling, M.P., Nicholls, G.: Stochastic block models as an unsupervised approach to detect botnet-infected clusters in networked data. Data Sci. Cybersecur. **3**, 161 (2018)
38. Saad, S., et al.: Detecting P2P botnets through network behavior analysis and machine learning. In: 2011 Ninth Annual International Conference on Privacy, Security and Trust (PST), pp. 174–180 (2011)
39. Sakib, M.N., Huang, C.-T.: Using anomaly detection based techniques to detect HTTP-based botnet C&C traffic. In: 2016 IEEE International Conference on Communications (ICC), pp. 1–6 (2016)
40. Saxena, A., et al.: A review of clustering techniques and developments. Neurocomputing **267**, 664–681 (2017)
41. Snijders, T.A.B.: Stochastic actor-oriented models for network dynamics. Ann. Rev. Stat. Appl. **4**, 343–363 (2017)
42. Strayer, W.T., Lapsely, D., Walsh, R., Livadas, C.: Botnet detection based on network behavior. In: Lee, W., Wang, C., Dagon, D. (eds.) Botnet Detection. ADIS, vol. 36, pp. 1–24. Springer, Boston (2008). https://doi.org/10.1007/978-0-387-68768-1_1
43. Szabó, G., Orincsay, D., Malomsoky, S., Szabó, I.: On the validation of traffic classification algorithms, In: Claypool, M., Uhlig, S. (eds.) PAM 2008. LNCS, vol. 4979, pp. 72–81. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79232-1_8
44. Tavse, P., Khandelwal, A.: A critical review on data clustering in wireless network. Int. J. Adv. Comput. Res. **4**(3), 795 (2014)
45. Torres, P., Catania, C., Garcia, S., Garino, C.G.: An analysis of recurrent neural networks for botnet detection behavior. In: 2016 IEEE Biennial Congress of Argentina (ARGENCON), pp. 1–6 (2016)

46. Wang, C.-Y., et al.: BotCluster: a session-based P2P botnet clustering system on NetFlow. Comput. Netw. **145**, 175–189 (2018)
47. Wang, J., Paschalidis, I.C.: Botnet detection based on anomaly and community detection. IEEE Trans. Control Netw. Syst. **4**(2), 392–404 (2016)
48. Xu, R., Wunsch, D.C.: Clustering algorithms in biomedical research: a review. IEEE Rev. Biomed. Eng. **3**, 120–154 (2010)
49. Yamauchi, K., Hori, Y., Sakurai, K.: Detecting HTTP-based botnet based on characteristic of the C & C session using by SVM. In: 2013 Eighth Asia Joint Conference on Information Security, pp. 63–68 (2013)
50. Zhang, J., Perdisci, R., Lee, W., Sarfraz, U., Luo, X.: Detecting stealthy P2P botnets using statistical traffic fingerprints. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), pp. 121–132 (2011)
51. Zhao, D., Traore, I., Ghorbani, A., Sayed, B., Saad, S., Lu, W.: Peer to peer botnet detection based on flow intervals. In: Gritzalis, D., Furnell, S., Theoharidou, M. (eds.) SEC 2012. IFIPAICT, vol. 376, pp. 87–102. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30436-1_8