©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

T-MAW: Online Network Traffic Monitoring and Analysis using Weighted Stochastic Block Models

Maximilian Stephan, Johannes Zerwas, Wolfgang Kellerer
Chair of Communication Networks
Technical University of Munich
Munich, Germany
{maximilian.stephan, johannes.zerwas, wolfgang.kellerer}@tum.de

***Abstract*—A significant portion of modern network traffic analysis still relies on human expertise only. To overcome human limitations in light of increases in volume, dynamicity, and overall traffic complexity, modern networks need to autonomously gain an understanding of traffic patterns and present them in an interpretable way. This work presents T-MAW, an approach for Traffic Monitoring and Analysis using Weighted Stochastic Block Models (WSBMs). T-MAW applies WSBMs to network data to create traffic characterizations in human-interpretable form. In addition to the insights gained from the fitted models, T-MAW evaluates unseen traffic against these models to perform anomaly detection. Both, network node behavior characterization and anomaly detection complement human expertise in modern network traffic analysis. As an example, we show how T-MAW can be used to create a behavior-based structured view of network nodes in a real campus network. In the anomaly detection context, we present results for an IP scan attack against the network, as well as from a layer-2 device fault that caused network disruption.**

***Index Terms*—ntma, machine learning, wsbm**

I. Introduction

Artificial Intelligence (AI) is becoming more relevant in all parts of our lives. However, network management still heavily relies on human expertise [1]. Within network management, network analysis is a key building block for most network operations. Even for non-automated decisions, operators usually need rich information about network dynamics. Considering increased traffic volume and complexity, human-only analysis becomes a problem in network analysis tasks. The key to allowing network growth beyond the limitation of human-only network operation is data-driven analysis methods. For specific use cases and applications, this has already been studied [2]. While useful in certain scenarios, we think that a general model that represents a certain understanding of the network's behavior and inherent dynamics is needed for daily network management work. Even considering the recent advances in the field of general-purpose AI, especially based on Large Language Models (LLMs), we think there is still a strong need for human experts in network management [1]. Instead of replacing human expertise, we propose to enhance it by supporting specialists as best as possible in their tasks by providing meaningful insights into network behavior through data-driven methods. For this to be effective, we propose the following criteria for suitable methods:

1) Approaches in this context should work on IP-level data.

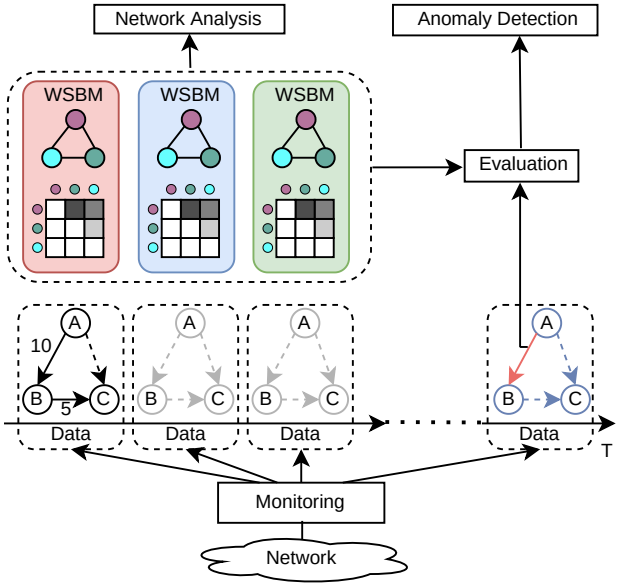


Fig. 1. T-MAW overview: Based on IP level flow data from network monitoring T-MAW represents chunks of said data as traffic graphs. For these graphs T-MAW fits WSBMs, resulting in host clustering and statistics about cluster-to-cluster relations. Each WSBM represents network behavior for the respective data and time. The whole ensemble of models allows for a node-behavior-based network analysis. Further, evaluating current traffic observations against the model collection T-MAW performs anomaly detection.

2) In modern networks with high levels of encrypted traffic, approaches should not rely on deep packet information.
3) Network phenomena can manifest on different levels (link, host, host-group, network). A holistic analysis approach should consider all of these levels.
4) Any considered approach should work in an online fashion to be usable in daily network analysis tasks.
5) To not restrict an approach to well-documented phenomena, considered methods should be unsupervised.
6) Models found as part of an approach should be interpretable to combine human and machine-learned intel.
7) Any approach should be tested with real data.

Addressing all of the above criteria, we propose T-MAW, a network analysis approach depicted in Fig. 1. T-MAW uses IP traffic observations to formulate probabilistic network behavior models (Weighted Stochastic Block Models (WSBMs)). The key component of such a WSBM is a host clustering, including cluster relations. Due to the unsupervised nature, the informa-

tion about what hosts are grouped together and how the groups relate to each other, provide important insights into network dynamics in a meaningful and interpretable way. In addition to these findings on a structural level, T-MAW is able to provide a comparison of newly seen traffic against the learned models. Such comparison results in an anomaly score that can be analyzed on the link, host, and network levels. Our work illustrates the effectiveness of the described functionalities by applying T-MAW to data from a production network. We showcase both, the characterization of the network through a behavior-based structured view of network nodes during normal operation and anomaly detection by identifying a scanning attack.

The main contributions of this work are:

- An approach for applying WSBMs to network traffic monitoring in an online fashion,
- characterization of network behavior dynamics using multiple WSBMs,
- anomaly detection using multiple WSBMs,
- application of these capabilities on real network data.

II. Requirements for data-driven network analysis

To provide background for the analysis and discussion of related work in Sec. III we define our key features for traffic monitoring and analysis as follows. Approaches should work with IP-level data. Most of today's computer networks are connecting hosts via the Internet Protocol. Additionally, the related data can naturally be collected at central points in the network. We think that it is of great importance to not rely on lower-level data, like actual topology, since this will immediately rule out an approach for a lot of use cases where such information might not be available.

As more and more traffic becomes encrypted it is important to not rely on packet-internal data that might be encrypted. Most of the times it is simply not feasible to work against encryption efforts. Moreover, we also believe that network monitoring shall be privacy preserving at all times.

Network behavior patterns manifest themselves on different levels. A holistic approach considers at least link-level aspects as well as host characteristics. A clustering-based approach like T-MAW can additionally unveil behavior patterns on the host-group level and provide valuable insight into collective characteristics. To conclude about the general state of a network, all of the above levels of analysis should be combined into a network/graph-level analysis.

To be of practical use in real-world networks we argue that network monitoring and analysis should ideally be achieved in an "online" fashion. While offline analysis definitely has applications, we think that a system that is actually applied in practice needs to work in a "live" fashion.

To not restrict the analysis to the recognition of patterns that are well-known beforehand, we think that approaches that use unsupervised learning are generally superior to the ones using supervised learning for the specific scenario at hand. Further, labeled data in large quantities is oftentimes expensive to obtain or simply not obtainable at all.

Neural Networks and other so-called black-box models have been successfully used in a variety of network analysis tasks. However, we believe that a solution for network monitoring and analysis has to be interpretable to a high degree. That means especially, that the properties of detected patterns should be reflected in the model properties in a way that the model itself can be helpful for analysis by gaining valuable insights into the network's dynamics.

Finally, we think that any approach that is a strong candidate for usage in network analysis should be tested with real data. Especially in the domain of anomaly detection, there is a strong evolution of solutions around a small set of widely used, partly synthetic, data sets with known properties for performance evaluation. While this makes approaches comparable to some extent, relying on such datasets only creates a lack of sense of how a particular approach will perform in a real-world scenario with all its complexity.

III. Related Work

This section provides an overview of related work. It is structured into two subsections. Section III-A covers existing work about network behavior characterization. Section III-B presents the usage of Stochastic Block Models in networking.

A. Network Traffic Analysis

The topic of network traffic monitoring and analysis (NTMA) is well-reviewed by the community. In the following, we present the three most recent surveys [3], [4], [5], that cover the scenario we are addressing with T-MAW most accurately.

Firstly, D'Alconzo et al. (2019) [4] give an overview of the state of the art for the usage of "Big Data" in Network Traffic Monitoring and Analysis (NTMA). From this extensive survey of different aspects of NTMA and the respective existing approaches, we consider SeLINA [6] closest to this work. Like this work, SeLINA relies on clustering to provide network insights and detect anomalies. However, SeLINA clusters flows based on statistical features and detects anomalies in changing RTTs compared to T-MAW's IP host clustering and behavior-based anomaly detection.

Pacheco et al. (2019) [5] survey the deployment of ML solutions in network traffic classification. Here, similar to this work, [7], [8], [9] use features from traffic graphs. Additionally, in the most recent suitable survey that we found, Lyu et al. (2023) [3] mention [7], [8], [9] in the context of host clustering and modeling of group interactions.

In [7], Jakalan et al. cluster IP hosts in a campus network based on their similarity in interacting with IP hosts from outside the network. In contrast, T-MAW focuses on IP host interactions within the local network. Jusko et al. [8] successfully identify member nodes of P2P networks through a clustering approach, thereby enhancing existing anomaly detection approaches.

By far closest to T-MAW is the approach proposed by Xu et al. [9]. Here, bipartite graphs and one-mode projections are

used to cluster end hosts, revealing behavioral structures within the network. Similar to T-MAW, the structural knowledge is then used to detect anomalies. While conceptually similar, the cluster meaning differs between [9] and T-MAW. In [9], similarity within a cluster is characterized by shared destination or source IPs among cluster members. In contrast, the WSBMs in T-MAW formulate similarity from group-to-group level relations (see Sec. IV). Therefore, both approaches capture a different form of structure in the network traffic.

B. Stochastic Block Models in Communication Networks

SBMs and their variants have been applied mainly in two different areas within the networking community, traffic synthesis and analysis. Kalmbach et al. [10] use SBMs to generate realistic synthetic traffic. In [11], WSBMs are used for offline data center traffic replication. Both NOracle [12] and AwareNet [13] perform analysis in the sense of anomaly detection. While NOracle [12] applies traditional SBMs to traffic in a testbed network and subsequently performs anomaly detection. AwareNet [13] uses a fitted WSBM to detect targeted host scans. The latter is our own prior work, which this work is partly built on. In contrast to T-MAW, both approaches rely on a single (W)SBM fit and, therefore, can not account for natural change within the network.

IV. Background

Stochastic Block Models (SBMs) are a class of generative models designed to understand and analyze network community structure. Originating from the field of network science and statistical modeling, SBMs provide a probabilistic framework for representing the heterogeneity and modular organization of complex networks. The concept of SBMs roots in the need to statistically model social networks. The initial formulation aimed to capture the intuitive notion that entities (nodes) within a network are often organized into groups (blocks or communities) with different patterns of connections both within and between these groups. The pioneering work by Holland et. al. [14] in 1983 formalized these ideas, laying the foundation for modern SBMs.

At its core, an SBM consists of:

1) Nodes and Communities: The network's nodes are partitioned into distinct communities or blocks.
2) Connection Probabilities: The probability of an edge existing between any two nodes depends solely on the communities to which these nodes belong.

This probabilistic approach allows SBMs to capture varying densities of connections within and between communities, thus modeling the modular structure of real-world networks. Formally, an SBM is defined by the following parameters:

- n: The number of nodes in the network.
- k: The number of communities.
- z: A vector of length n where z_i denotes the community assignment of node i
- B: A KxK matrix where B_{xy} represents the probability of an edge between a node in community x and a node in community y.

An extension to the traditional SBM are Weighted Stochastic Block Models (WSBMs) [15] [16]. While the traditional SBM framework assumes binary edges (i.e., the presence or absence of a connection), many real-world networks exhibit weighted edges, where connections have varying strengths. To address this, the WSBM extends the SBM by incorporating edge weights into the model.

In a WSBM, each edge (i,j) is assigned a weight w_{ij}, which can be modeled in various ways depending on the application. Common approaches include using a continuous distribution such as Gaussian or exponential to represent the edge weights. The model then defines a probability distribution for the weights between nodes, conditioned on their block memberships. Formally, the WSBM includes a weight distribution matrix W in addition to the block partition and connection probability matrix B. Each entry W_{xy} specifies the parameters of the weight distribution for edges between nodes in block x and block y. This allows the WSBM to capture not only the presence of connections but also their intensity, providing a richer representation of the network structure. WSBMs are particularly useful in fields where the strength of interactions between nodes plays a crucial role, which is the case in communication networks, where traffic flow intensity is relevant. By incorporating edge weights, WSBMs enhance the ability to detect and analyze communities in weighted networks, leading to more nuanced insights and robust community detection results.

In summary, SBMs and their weighted counterparts, WSBMs, offer powerful methodologies for modeling and analyzing complex networks. These models help uncover underlying community structures and provide a deeper understanding of the connection patterns within various types of networks.

V. Traffic Analysis with T-MAW

Given the capabilities of WSBMs described in Section IV, T-MAW's overall process is as follows: Firstly, T-MAW represents chunks of IP level flow data obtained from a suitable monitoring system and aggregated over a fixed-size time window as directed weighted graphs. Then, the relevant traffic characteristics of each chunk of data are captured by fitting a WSBM to said graph. The resulting host clusterings and the respective cluster-to-cluster relations for different time windows can already be used for traffic analysis by unveiling network inherent dynamics. Further, by evaluating current traffic observations against the collection of fitted WSBMs containing characteristics for network behavior in previous time windows T-MAW creates an online anomaly score for present data. Fig. 1 shows an overview of this architecture. The next subsections cover the individual parts in more detail.

A. Fitting a WSBM to traffic observations

At its core T-MAW relies on fitting a Weighted Stochastic Block Model to traffic that was observed during a specific time period. The idea is that the resulting model then contains characteristics about the network behavior from the respective time period. The raw data stems from a monitoring system

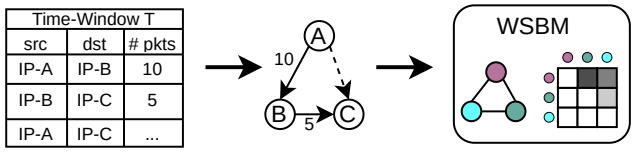


Fig. 2. WSBM fit: Data about IP to IP communication for a given time window is represented as a directed graph. Then a WSBM is fit to the graph to capture the communication pattern.

that processes a stream of mirrored traffic from the network's core switches. Any other technology that allows the collection of flow-level data, like, for example, NetFlow, would work equally well in this scenario. Here, a state-of-the-art traffic analyzer, including special capture hardware, receives the mirror streams and processes the raw packets. Aggregated data about IP-to-IP communication in terms of sent packets and bytes is then created for 1 min buckets (finest granularity at time of writing). T-MAW pulls and further aggregates the 1 min data chunks relevant for a specific time window of interest via a REST API. Fig. 2 shows a scheme of this data for a single time window represented as a table. It contains information about the number of packets that were observed from individual source to destination IP address pairs within the time window. Note that T-MAW currently only considers IP address (Layer 3) information but including Layer 4, i.e., considering five tuples is a straightforward extension. T-MAW then creates a directed graph representation from the data. In the graph, nodes represent IP hosts, and edges represent the traffic seen from one host to another, described by the number of packets. For example, 10 packets from IP-A to IP-B would be represented by a directed edge from A to B annotated with a weight of 10. If no packet was observed from one host to another, no edge will be added. Consequently, all edges have at least a weight of 1. The resulting graph includes two important features about the network traffic:

1) Which nodes are communicating with each other?
2) How "heavily" do two communicating nodes interact?

To structure nodes according to these characteristics, T-MAW fits the parameters of a WSBM, as described in Sec. IV, to the graph. In more detail, we chose the degree-corrected hierarchical variant of the WSBM in this work [17], [18]. T-MAW uses a normal distribution for the general weight distribution form since this has already shown promising results in previous work [13]. Finding the 'optimal' fit for any variant of SBMs is an NP-hard problem. A Markov-chain monte-carlo approach is used to find 'good' fits [19]. T-MAW uses the Graphtool [20] library to implement the described WSBM fit procedure. The resulting model consists of:

- a partition z of the nodes into k groups. This is indicated by the node's colors in Fig. 2.
- statistics about group-to-group relationships based on edge existence and edge weight distribution between member nodes of the groups (indicated by the greyscale colors of the matrix in Fig. 2).

Recall that the unsupervised nature of this method, and thus,

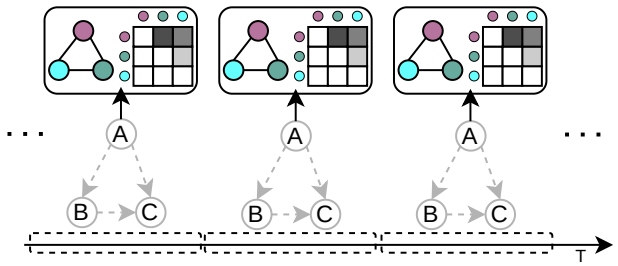


Fig. 3. Fitting multiple WSBMs over time.

node grouping and the found group relations, can yield new insights into the network state for the respective time window. However, a single WSBM is not enough to capture the changing nature of modern communication networks. As depicted in Fig. 3, T-MAW fits an individual WSBM for consecutive time windows. The series of models then represents the network behavior over time.

B. Multi-WSBM traffic evaluation

The fitted WSBM represents the network traffic characteristics seen during the respective time window, as described in the previous subsection. A natural next step is to evaluate how well a traffic observation from outside the model-related time window fits these characteristics. Given the WSBM's representation of whether an edge between two nodes exists and, if so, how heavy the connection is in terms of the number of packets sent over the edge, we formulate likelihoods for arbitrary observed edge data.

For any edge e_{ij} between node i and j, the log-likelihood for the existence of this edge in context of the WSBM model θ is calculated according to:

$$\log \mathcal{L}_{exist}(e_{ij}|\theta) = \log \mathrm{Pois}_{\lambda_{z_i,z_j}}(1) \tag{1}$$

Where z_i and z_j denote the respective groups of the two involved nodes i and j in θ. At the core, Eq. 1 assumes a Poisson distribution for edge existence between two nodes from groups z_i and z_j. λ is calculated as the ratio of total observed edges from nodes in group z_i to nodes in group z_j and the total number of possible edges between nodes from the two groups. Here, both observed and possible edges refer to the data that θ was originally fitted to. The related λ value can be computed at the time of model fitting according to:

$$\lambda_{z_i,z_j} = \frac{|e_{observed}|}{|z_i| \cdot |z_j|} \tag{2}$$

Assuming independence from edge existence, we calculate the log-likelihood for any edge e_{ij} with a certain number of packets annotated as weight w_{ij} as follows:

$$\log \mathcal{L}_{weight}(e_{ij}|\theta) = \log \mathcal{N}_{\mu_{z_i,z_j},\sigma_{z_i,z_j}}(w_{ij}) \tag{3}$$

In accordance with the WSBM fitting process, Eq. 3 assumes a normal distribution for the edge weights. Both parameters μ_{z_i,z_j} and σ_{z_i,z_j} are the respective maximum likelihood estimates based on the set of weights annotated to edges from group z_i to group z_j in the data that was used to fit the model.

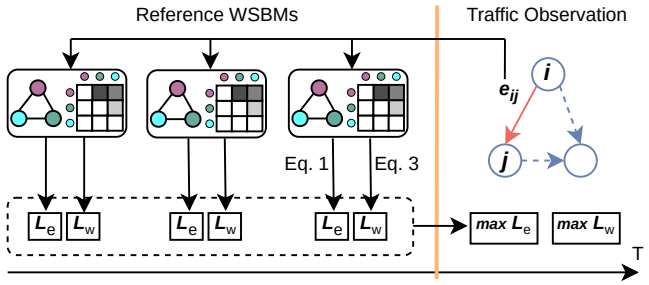


Fig. 4. Traffic Evaluation.

T-MAW uses the evaluation scheme depicted in Fig. 4 to enable multi-level anomaly detection. On the left are the reference WSBMs representing known network behavior patterns from past time windows, created as shown in Fig. 3. On the right, Fig. 4 shows the traffic observation from the current time window represented in graph form. From there, any edge e_{ij} is evaluated against all reference WSBMs. For each WSBM, log-likelihood values for existence and weight (L_e and L_w in Fig. 4) are calculated for e_{ij} according to Eq. 1 and Eq. 3. The result is a vector of values for each likelihood representing how e_{ij} from the current observation fits communication patterns in the reference WSBMs. To capture the notion of whether the behavior described by e_{ij} in the current observation was seen before, T-MAW extracts the maximum values from the likelihood vectors, representing the best fit. Any current traffic observation (graph in Fig. 4) might result from overlapping independent patterns seen at different times in the past. Consequently, these patterns are possibly encoded in different reference WSBMs. To account for that, the procedure is done per edge, allowing edges to have different best-fit WSBMs.

The whole evaluation can be performed directly on any newly seen traffic graph, allowing T-MAW to be deployed in an online fashion.

VI. Data and Network

To provide an understanding of the data that is used in the evaluation part, this section briefly introduces the monitored network and presents notable statistics about the collected data.

The underlying network is the production network of a large research group at a German university. It holds infrastructure for network experiments, machine learning tasks and data storage. Further, it provides services like cloud and chat applications. In summary, the network activities result in a healthy mix of static, periodic and dynamic traffic patterns.

This is reflected in the number of active nodes for 30 minute intervals depicted in Fig. 5. In order to limit the number of different nodes that T-MAW needs to keep track of, all IP addresses that are not native to any of the locally known subnets are mapped onto one single node. With that, we still retain the knowledge of how nodes connect with the outside. Over 6 months of data collection, the number of total unique nodes observed is at around 500. While nearly all of the nodes act as destinations (*dst*) in most of the time windows, the proportion of nodes that act as sources (*src*) is smaller at

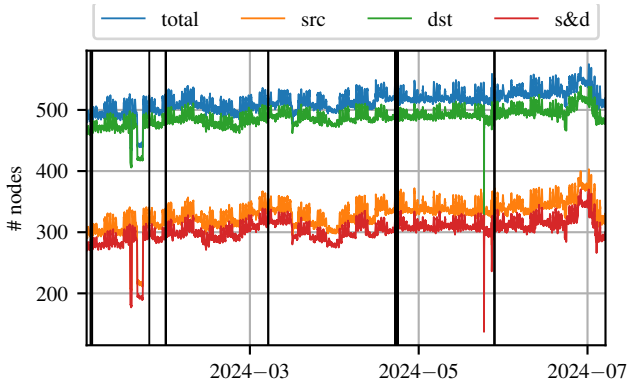


Fig. 5. Number of unique active nodes aggregated for 30 minute intervals in the last 6 months. The orange and green lines distinguish between nodes that were seen as source (src) and destination (dst) of a packet respectively, the blue line represents all nodes, regardless whether seen as src or dst and the red line represents nodes that were seen as src and dst. (Missing data is marked with black lines)

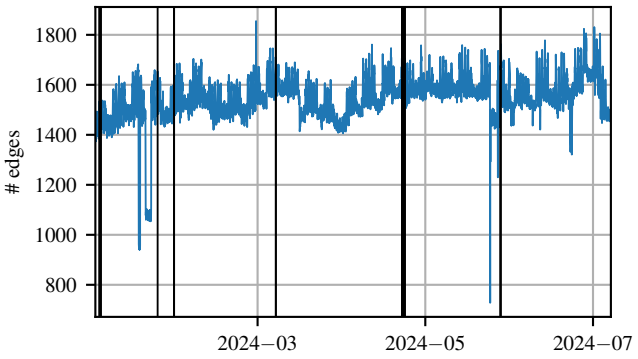


Fig. 6. Number of unique src-dst pairs (edges) aggregated for 30 minute intervals in the last 6 months. (Missing data is marked with black lines)

around 300. The number of nodes that act as both sources and destinations (*s&d*) is even slightly smaller.

For the same time period, the number of observed unique edges is between 1500 and 1750 (see Fig. 6). Both, the number of nodes and edges show a static component with dynamic patterns on top. Further, Fig. 7 depicts the distribution of the number of packets recorded for the seen edges. It shows the presence of various connection intensities between nodes. While a lot of edges have few packets, some nodes exchange

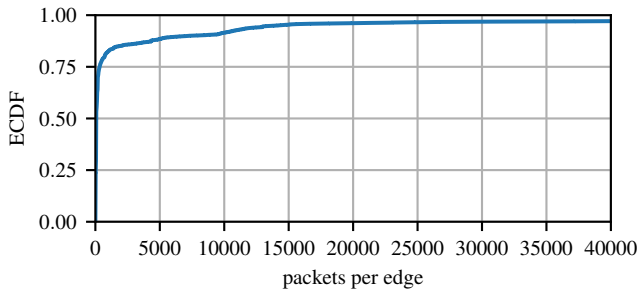


Fig. 7. Empricial CDF of number of packets seen per edge in 30 minute intervals in the last 6 months.

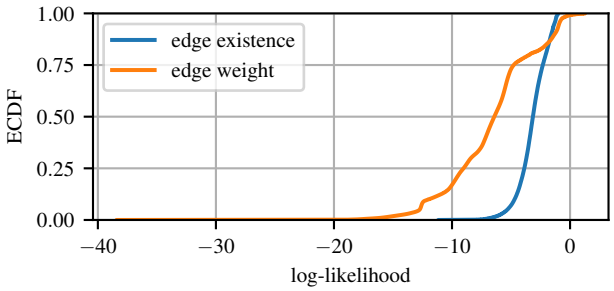


Fig. 8. ECDFs of log-likelihood values for edge existence and weights.

vast amounts. We think that all those characteristics render the presented data an excellent basis to evaluate our approach.

Because T-MAW relies on the ability to track hosts over time via their IP, frequent DHCP-related IP changes have the potential to throw off T-MAW. However, the network investigated partly uses DHCP, and we have not experienced issues so far. We suspect that this is due to relatively long lease times. In a more fast-paced DHCP setting, another layer of device mapping that assigns fixed identities to endpoints could solve the problem (e.g., host names).

Ethical considerations: We are aware of the ethical challenge that comes with working on data from a live network. We want to stress again that the data itself stems from the datacenter/core parts of the overall network. Therefore, it does not include day-to-day user traffic (e.g. browsing). However, if users interact with components in the core part of the network (e.g., an experimental setup), the related data is recorded on the described level. Therefore, IPs that are in a part of the address space that allows linkage to a specific person are pseudonymized before usage. Furthermore, the users are aware of the traffic monitoring system in place and are encouraged to approach us with any concerns.

VII. Evaluation

In this section, we evaluate T-MAW as described in Section V on the data described in Section VI. This involves fitting a WSBM for each 30 minute interval in the data. Because both, Eq. 1 and 3 return unnormalized log likelihoods we need to establish a notion which value ranges can be considered normal. For that, we evaluated every edge against the WSBM from the edge's own time window. Fig. 8 shows the results of that as ECDFs of log-likelihood values for edge existence and weights. From the results, we derive a normal value range for edge existence of -1 to -8 and 0 to -20 for edge weight log-likelihoods.

Comment: We removed a single node that had consistently bad edge weight log-likelihoods from the results. Upon further investigation, we found that it is an SNMP-based network management node.

A. Network Analysis

Next to the online evaluation of newly seen traffic described in Section VII-B, its interpretability is the main perk of the T-MAW approach. One aspect of this is that the fitted WSBMs provide a natural behavior-based grouping of the network nodes. In contrast, many network management and analysis tools group nodes by subnet membership. Especially in heterogeneous networks, this approach groups together nodes with very different roles and behaviors.

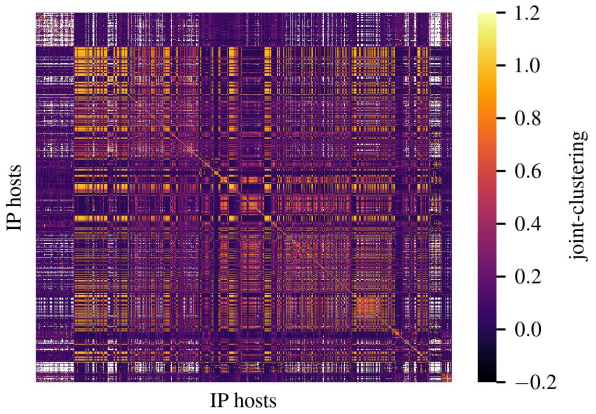


Fig. 9. Node to node relative joint-clustering values. Nodes on the x and y axis are sorted by /24 subnets. White fields indicate missing values.

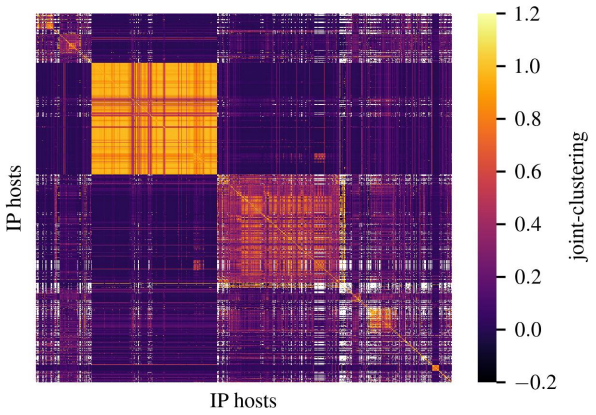


Fig. 10. Node to node relative joint-clustering values. Nodes on the x and y axis are sorted by scikit-learn's AgglomerativeClustering. White fields indicate missing values.

For the data from one month, we analyzed how often two nodes are clustered together across all WSBMs fitted for that month. The resulting value is then normalized by the number of their joint appearances. This then represents a kind of similarity. For example, if node A and node B were both seen in the same 100 30-minute time windows and were in the same cluster in 50 of those intervals, the resulting value would be 0.5.

Fig. 9 is a heatmap of these values for all node pairs sorted by /24 subnets. The white cells indicate that the respective nodes are never seen in the same time window, resulting in missing values for our calculation. The matrix shows structure to some degree, especially on the outer part.

In order to show the potential benefit of a behavior-based structure, we applied scikit-learn's *AgglomerativeClustering* [21] method to the data from Fig. 9. Fig. 11 depicts the related dendrogram. Here, the y-axis values are inverse to the heat values in Fig. 10 and represent distance instead of similarity. Fig. 10 shows the very same data as Fig. 9 but resorted

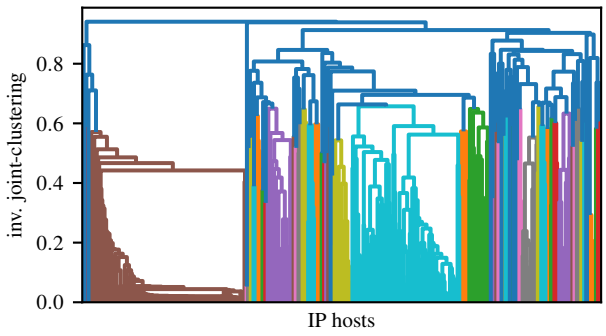


Fig. 11. Dendrogram for AgglomerativeClustering of nodes based on joint cluster occurences. Values on the y-Axis are inverse to Fig. 9.

based on the outcome of the AgglomerativeClustering. The difference between subnet-based and WSBM-clustering-based visualization becomes apparent immediately. Node pairs with high similarity values are grouped. Especially the large bright orange-colored square on the top left provides a concise view of a group of nodes from different subnets that are consistently clustered together over time. Upon further investigation, the respective nodes can be identified as client machines with moderate amounts of communication. In Fig. 11 the respective group (brown color) additionally shows how well the particular nodes fit together.

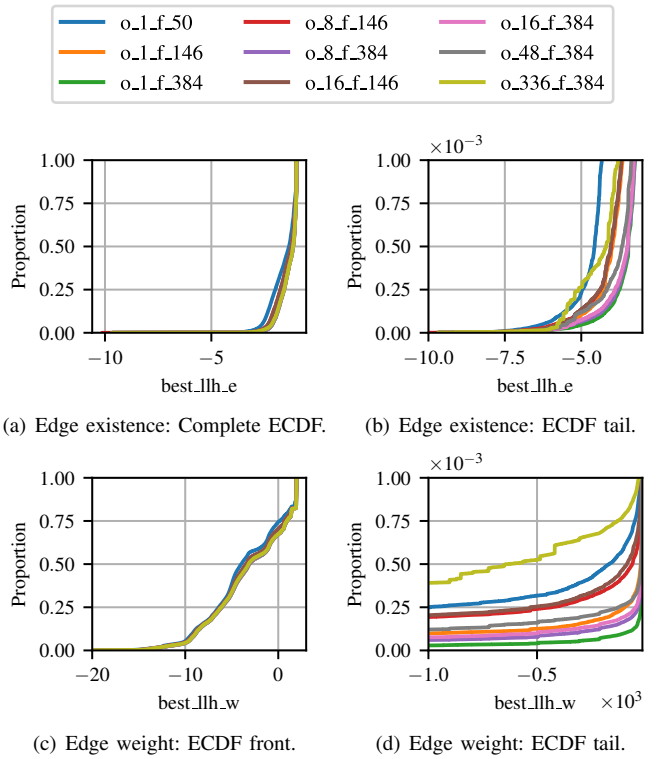
In summary, we propose to use the node clustering provided by the T-MAW approach to create a behavior-based view of network nodes, in addition to existing ones, for traffic analysis. We acknowledge that, as of now, this analysis is quite static in the sense that it is performed on historic data and does not account for a stream of new data and the respective WSBMs. Future work is directed towards solving this by creating a meaningful connection between consecutive WSBMs that, for now, are treated independently.

B. Anomaly detection

As described in Section V, one objective of the proposed T-MAW approach is to evaluate traffic observations for anomalies based on how well current observations fit previous fitted models. Because T-MAW is an unsupervised approach by design and, therefore, does not deal with any labels, the question arises: ”What exactly constitutes an anomaly?” The short answer: ”Everything that differs enough from the previously seen patterns!” To establish a more firm notion about what ”enough” and ”previously” means in this context, we further formalize the T-MAW approach as follows.

It is certainly not feasible for a running system to keep a record of all WSBMs fitted for every 30-minute interval forever. Also, depending on the resources, it might not be possible to have the latest fitted WSBM ready for every interval. To formalize the specific WSBMs in the evaluation context for this study, we introduce two parameters:

- o: An offset describing how far in the past the latest considered model is.
- f: The number of models considered, going back in time, starting at the offset o.



(a) Edge existence: Complete ECDF. (b) Edge existence: ECDF tail.

(c) Edge weight: ECDF front. (d) Edge weight: ECDF tail.

Fig. 12. ECDF of log-likelihood values of edge existence (a & b) and weight (c & d) for evaluations with different parameters o and f. The first 15 days of the evaluation were cut for fairness between parameter sets.

Both o and f are in multiples of the time window size (here 30 min). For example, $o = 2, f = 48$ describe the case where the first model to consider is from one hour ago. From there, the considered models would cover 24 hours backwards in time.

Following, we applied T-MAW to the whole 6 months of data with different o, f combinations. That means every edge in every 30-min time window was evaluated against all WSBMs specified by o and f with respect to the evaluated time window itself. From all the values calculated in accordance with Eq. 1 and 3, the best ones were picked for each edge.

Fig. 12(a) shows the ECDF of edge existence log-likelihood values for various parameter settings. As expected, there is a steep increase towards 1.0 at the -1 mark. This corresponds to $\lambda = 1$ from the underlying Poisson distribution. The tail end of the values is shown in more detail in Fig. 12(b). Here, it becomes evident that both large o (light blue line) and small f (orange line) struggle to find good matches in some cases. With 384 models corresponding to 8 days of data to match against, the configurations depicted by red, brown, and grey show stable behavior.

A similar conclusion can be drawn from Fig. 12(d) depicting a part of the tail of the ECDF for the edge weight log-likelihood values. While the front part of this ECDF shown in Fig. 12(c) indicates a wider range of common log-likelihood values for edge weights than for existence.

Recall that to evaluate an edge against a specific WSBM, both edge nodes need to be present in the model or, in other

TABLE I
INVALID EVALUATIONS FROM 12.7M EDGES

variant	*no-match*	*no-valid-llh-w*	*no-valid-llh-e*
o_1_f_50	12541	13688	12742
o_1_f_146	6720	6892	6834
o_1_f_384	3225	3334	3306
o_8_f_146	30515	30766	30683
o_8_f_384	13442	13622	13577
o_16_f_146	47317	47505	47454
o_16_f_384	20688	20863	20805
o_48_f_384	41196	41527	41461
o_336_f_384	184920	185207	185122

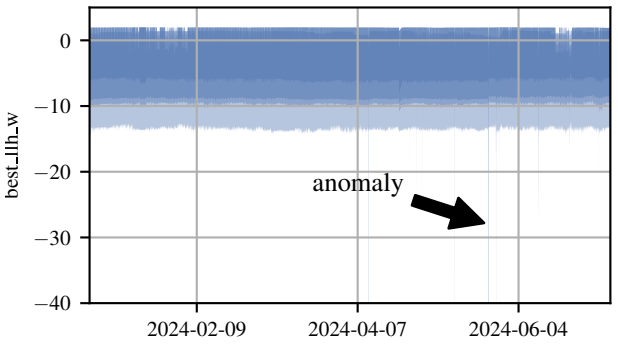


Fig. 13. Distribution of per edge best weight log-likelihood values for each time window. Color shades from light to dark describe percentile ranges (1, 99), (5, 95), (10, 90), (25, 75).

words, need to be seen during the time window corresponding to the model. Naturally, not all node pairs are active in all time windows, which is why the total number of calculated likelihood values to choose from per edge can vary. Worst case, no match can be found for a particular edge. Further, there is the possibility that in some cases, the distribution parameters for evaluation of a certain edge against a specific WSBM do not allow a valid calculation. For example if λ_{z_i,z_j} from the Poisson distribution in Eq. 1 becomes zero due to no observed edges between the groups of the two nodes. Table I shows an overview of how often no valid evaluation can be performed out of all 12.7M edges for different settings. (Note that both "no-valid-llh-w" and "no-valid-llh-e" also include the cases from "no-match".) Like before, the setting with $o = 1$ and $f = 384$ shows the best results, but especially $o = 8$ and $f = 384$ seems to offer a good trade off between performance and having a higher offset. That is why the remainder of this section utilizes this particular variant.

1) Natural Anomaly: One of the big advantages of T-MAW is that its unsupervised nature allows for detecting all kinds of anomalies. To showcase this capability we applied T-MAW to 6 months of data. Fig. 13 shows the distribution of per edge best weight log-likelihood values for each time-window. The different color shades represent different percentile ranges. With the lightest one covering values between the 1th to 99th percentile. Similarly, ranges between 5th to 95th, 10th to 90th, and 25th to 75th are represented by increasingly darker shades. (Note, the graph is cut off at -40 to increase visibility in the main focus area. The visible spikes that touch the bottom line extend to more than -1000 without exception.) While

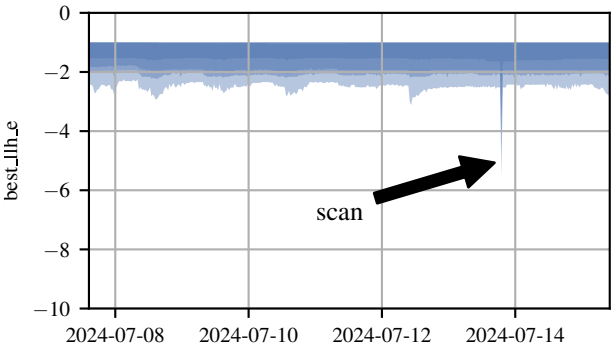


Fig. 14. Distribution of per edge best existence log-likelihood values for each time window. Color shades similar to Fig. 13.

there are some instances that show significant drops the most prominent one is the one marked with an arrow. An extended analysis for the specific time window revealed very low weight-likelihoods for pairwise edges between a select small group of nodes. The respective IPs belong to management interfaces of servers in the research group's compute cluster which are used to synchronize the cluster state. At the time, a layer-2 fault on one of the data center switches caused a loss of connection between the nodes. The result was that some of the nodes panicked, causing an unusually high amount of cluster sync packets. This example indicates that applying T-MAW can facilitate the detection of naturally occurring faults in the network.

2) Malicious Intent Anomaly: In contrast to such natural anomalies are actions driven by malicious intent. Oftentimes such malicious activities contain some sort of network scan as part of the initial reconnaissance phase. In this subsection, we show how applying T-MAW can facilitate the identification of a network ping scan. Fig. 14 shows the distribution of per edge best existence log-likelihood values for each time window (similar to Fig. 13). The underlying data represents an 8 day period. In the second half of this period, we initiated a Nmap ping scan of three local subnets on a virtual machine usually used for ML tasks. The ping scan causes a clear spike in the edge existence log-likelihood values. Many of the observed edges created by the scan heavily violate the usual connection properties and thus allow detection of the scan.

One case where a valid anomaly judgment might be difficult with T-MAW is if a previously unseen IP starts to behave maliciously immediately. Since T-MAW's notion about what is "normal" and what constitutes an "anomaly" relies on previously seen patterns, the malicious behavior of that IP would not show a significant deviation in likelihood. One solution could be to identify newly seen hosts and flag their behavior as unknown until their initial legitimacy is confirmed.

VIII. DISCUSSION

To complement the findings of Section VII, this section discusses important considerations in the context of T-MAW. **Scalability:** A single run of the Markov-chain Monte Carlo

method for finding good WSBM model fits is linearly dependent on the number of edges [20]. For the 1500-1750 edges per time window of our data, we ran the method for 1000 iterations. On a single standard server core, this took about 5min for a single model to be fitted onto a 30min interval of data. This enables T-MAW to run easily in an online fashion. The calculation of log-likelihoods for traffic observations is linearly dependent on the number of edges to evaluate and the number of models to evaluate against (dependent on o and f). Both, fitting and log-likelihood calculations lend themselves well to parallel computing. The memory complexity for a single model is given as $O(k^2 + n)$. With k groups and n nodes, this represents group-to-group relations as well as node-to-group assignments that need to be stored. The total memory needed then is, of course, linearly dependent on the number of models used in the specific variant of T-MAW.

Data availability: This work relies on mirrored traffic from core switches of the network, but other solutions involving tools like NetFlow can also support a T-MAW approach. One important consideration is the place the data is collected. Ideally the monitored network allows for a collection of traffic at only a few central devices.

Online vs. real-time: T-MAW could be applied to network monitoring data in an online fashion, e.g., in the context of anomaly detection. However, it is important to distinguish between "online" and "real-time". "Online" refers to the fact that each new time window data is processed immediately with minimal delay. But because T-MAW operates on graph representations of traffic, it involves aggregating monitored traffic until a meaningful new graph has formed creating an inherent delay (in our case, 30 minutes). This is the reason why, in its current form T-MAW is not "real-time"; efforts in that direction are part of our ongoing research.

IX. Conclusion

This work presents T-MAW, a multi-WSBM-based approach for network analysis and online anomaly detection. We describe a methodology for applying WSBMs to IP network traffic observations over time in a meaningful way. A collection of fitted WSBMs then allows for a behavior-based structured view of network nodes that facilitates understanding of network inherent dynamics. Further, by evaluating new traffic observations against a collection of WSBMs, T-MAW is able to perform anomaly detection. We show this by applying the approach to 6 months of data from a production network and identifying an occurring fault in a layer-2 device as well as an initiated scan attack against parts of the network. Possible avenues for future work include the integration of more fine-grained traffic data such as Layer 4 information, the identification of more sophisticated attack profiles as well as evaluating possible steps towards a real time application.

Acknowledgment

This work is partially funded by Federal Ministry of Education and Research in Germany (BMBF) as part of the projects AI-NET-PROTECT (16KIS1294) and 6G-life (16KISK002).

References

[1] T. Zhang, H. Qiu, M. Mellia, Y. Li, H. Li, and K. Xu, "Interpreting AI for networking: Where we are and where we are going," *IEEE Commun. Mag.*, vol. 60, no. 2, pp. 25–31, 2022.

[2] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine learning in network anomaly detection: A survey," *IEEE Access*, vol. 9, pp. 152 379–152 396, 2021.

[3] M. Lyu, H. Habibi Gharakheili, and V. Sivaraman, "A survey on enterprise network security: Asset behavioral monitoring and distributed attack detection," *IEEE Access*, vol. 12, pp. 89 363–89 383, 2024.

[4] A. D'Alconzo, I. Drago, A. Morichetta, M. Mellia, and P. Casas, "A survey on big data for network traffic monitoring and analysis," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 800–813, 2019.

[5] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2019.

[6] D. Apiletti, E. Baralis, T. Cerquitelli, P. Garza, D. Giordano, M. Mellia, and L. Venturini, "Selina: A self-learning insightful network analyzer," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 696–710, 2016.

[7] A. Jakalan, J. Gong, Q. Su, X. Hu, and A. M. Abdelgder, "Social relationship discovery of ip addresses in the managed ip networks by observing traffic at network boundary," *Computer Networks*, vol. 100, pp. 12–27, 2016.

[8] J. Jusko and M. Rehak, "Identifying peer-to-peer communities in the network by connection graph analysis," *International Journal of Network Management*, vol. 24, no. 4, pp. 235–252, 2014.

[9] K. Xu, F. Wang, and L. Gu, "Behavior analysis of internet traffic via bipartite graphs and one-mode projections," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 931–942, 2014.

[10] P. Kalmbach, A. Blenk, M. Kluegel, and W. Kellerer, "Generating synthetic internet-and ip-topologies using the stochastic-block-model," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 911–916.

[11] P. Kalmbach, L. Gleiter, J. Zerwas, A. Blenk, W. Kellerer, and S. Schmid, "Modeling ip-to-ip communication using the weighted stochastic block model," in *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*, 2018, pp. 48–50.

[12] P. Kalmbach, D. Hock, F. Lipp, W. Kellerer, and A. Blenk, "Noracle: Who is communicating with whom in my network?" in *Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos*, ser. SIGCOMM Posters and Demos '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 48–50.

[13] M. Stephan, P. Krämer, and W. Kellerer, "Awarenet: using wsbms for network traffic analyis," in *Proceedings of the 3rd International CoNEXT Student Workshop*, ser. CoNEXT-SW '22. Association for Computing Machinery, 2022, p. 35–36.

[14] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[15] C. Aicher, A. Z. Jacobs, and A. Clauset, "Learning latent block structure in weighted networks," *Journal of Complex Networks*, vol. 3, no. 2, pp. 221–248, 2015.

[16] T. P. Peixoto, "Nonparametric weighted stochastic block models," *Physical Review E*, vol. 97, no. 1, p. 012306, 2018.

[17] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 83, no. 1, p. 016107, 2011.

[18] T. P. Peixoto, "Hierarchical block structures and high-resolution model selection in large networks," *Physical Review X*, vol. 4, no. 1, p. 011047, 2014.

[19] ——, "Efficient monte carlo and greedy heuristic for the inference of stochastic block models," *Physical Review E*, vol. 89, no. 1, p. 012804, 2014.

[20] ——, "The graph-tool python library," *figshare*, 2014. [Online]. Available: http://figshare.com/articles/graph_tool/1164194

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.