




# DualNetView: Dual Views for Visualizing the Dynamics of Networks

V. Pham , V. T. N. Nguyen , and T. Dang 

Computer Science Department, Texas Tech University, Lubbock, Texas, USA

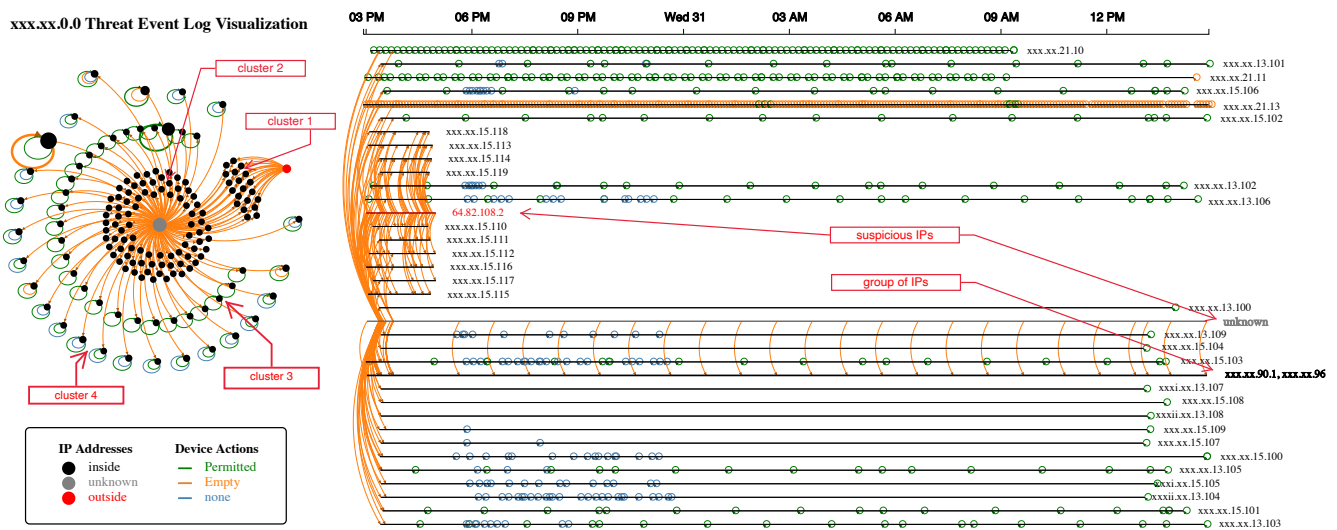


Figure 1: DualNetView: visualizing threat logs over one-day data from xxx.xx.0.0 subnetwork of an organization.

## Abstract

The force-directed layout is a popular visual method for revealing network structures, such as clusters and important vertices. However, it is not capable of representing temporal patterns, such as how clusters/communities evolve. Dynamic network visualizations trade the overall structures for temporal relationships. In this paper, we present a dual view framework for capturing both overall structures and temporal patterns within networks. The linked supplemental views utilize the strengths of both visualization techniques to provide useful insights into the given networks. To demonstrate the usefulness of our proposed dual views, we provide three use cases of dynamic networks: computer networks communications, activities of suspicious processes in computer systems, and social networks.

## 1. Introduction

Network representations are increasingly popular thanks to their expressive power to describe and support in analyzing relationships among interesting entities. Therefore they are being used in numerous application domains from cybersecurity [PD18] to soil profiling [PD19b, SBD\*20]. Most of the real-life networks are dynamic (i.e., their structures evolve) [CG18]. The dynamics may be due to the changes in participating components (e.g., joining and leaving) or the pattern of communications. The representative/analysis requirements for dynamic networks often mean (1) providing overview (i.e., to support analysis as a static network),

(2) having details on how the network changes over-time (i.e., to support dynamic network analysis). Both of these requirements, whenever possible, should help to capture trends or irregular activities within the network.

Time series data is significant in many application domains from fraud detections [PD19a, PNL\*19] to animal movement analysis [RTJ\*11]. Thus, visualization solutions to tackle previously discussed analysis requirements are numerous with their own strengths and limitations. Due to limited expressing power of individual visualization techniques, most of the existing researches either does well at describing the static part (providing overview) or

showing the dynamic behaviors of the dynamic networks (i.e., indicating changes and trends). We argue that both requirements are significant in network analysis. Specifically, dynamic network data grows in size and complexity at an exponential rate. Thus, there is a need to summarize and give a brief overview to avoid overwhelming the analyst. On the other hand, for some critical events (e.g., cybersecurity attacks), analysts often need to drill down into low-level and temporal details while examining the circumstances. Therefore, in this paper, we provide a methodology that adopts a dual view approach to dynamic network representation and analysis. Thus, three novel contributions of this work are (1) aggregated node link diagram across all timesteps, (2) linking aggregated node link diagram with detailed linear network visualization, and (3) application of the technique in three use cases.

## 2. Related work

Visualizations gain additional research importance when applied to the computer network and network security [ZTZ\*19], and dynamic social networks [CSBG18]. Thus, this paper focuses on recent works related to these application domains. Guimaraes et al. [GFS\*15] and recently Zhao et al. [ZTZ\*19] provided several overviews of visualizations systems for network security. There are various visualizations techniques used in the reviewed works. However, the commonly used visualizations are node-link and tree view (some other methods are radial panel, parallel coordinates, and scatterplot, to name but a few).

Since the nature of the underlying network data is about computation nodes and their communications, node-link representation [NK13, WQY\*12] is the most used visualization techniques in this field [NK13, WQY\*12, MDS13, ZN12]. In this technique, systems are visualized as nodes, while links are communications among nodes. Other visual attributes, such as size, color, and line width, are used to encode further information. Also, common hierarchical network structures give rise to the use of tree representations in this field [KKL\*14, AAAAS13, MGC12], specifically, Internet Protocol address (IP, hereafter) prefixes or domain name structures are used to aggregate data at different levels. Furthermore, interactions are added to allow analysts to analyze data at various granularities. To visualize the changes in a dynamic network, dynamic streams are often used [IES\*12, TEK\*13].

In the field of dynamic malware analysis due to a large amount of the execution data, visualizations also tend to gain success. Furthermore, it involves analyzing interactions (e.g., load images, read data, and write data) between the monitoring malware and other related entities (e.g., processes, files, remote network domains). Thus, it is not surprising that many of them are also modeled and analyzed as networks. For instance, Gregio and Santos [GS11] developed an interactive timeline tool for visualizing dynamic malware behavior using various visualization techniques. They ran a given malware in a controlled environment and captured its behavior using a malware behavior monitoring tool. Also, Wüchner et al. [WPO14] proposed DEVAST to dynamically analyze the malware behavior using quantitative data flow graphs (QDFG).

In social network analysis, Greilich et al. [GBD09], van den Elzen [vdEHBvW13], Vehlow [VBAW15], and recently, Dang et

al. [DPF16], proposed a novel approach to visualizing the dynamic relationships between entities. The horizontal layout represents the time, and they used force-directed layouts to bring related entities together in the vertical orientation. Horizontal lines represent participating nodes, and directed arcs represent communications among participants. It then gained initial successes and being used in a broad range of application domains, which involve network analysis, from social community detection [DN18] to Biology [DMF17]. Though it is good at providing temporal information, it still has limitations in providing a summarized view about clusters and communications of the network in a period as a whole [VBSW13].

In this work, we adopt the node-link approach to create a static network graph view that gives an overview of the analyzing data. Also, similar to several other works [MMK08, OG07], we use force-directed layout [BOH11] with force formula defined to form clusters of nodes with similar attributes and frequent communications. Furthermore, we also provide a dynamic network [DPF16] to visualize temporal patterns while analyzing network data. These two views are complementing each other while analyzing data. Specifically, the first one gives quick summaries, so the analyst is not lost in detail, while the latter provides options for the user to drill down to temporal details, in case needed. Furthermore, interactions are used as a mechanism to link between them.

## 3. DualNetView Visualization and Interactions

The components of our DualNetView interface, when applied in different application domains, may have slightly different aims/meanings. However, for the sake of simplicity, this section describes its components via an application in visualizing network threat logs. Figure 1 depicts DualNetView visualizations applied to network threat log data from a subnetwork of an organization ('xxx.xx.0.0' subnet mask, the actual network domain address is removed due to privacy requirement). On the left panel, a network graph is used to give an overview of the participants in this network and their communications during the monitoring period. The circles designate participating computers (with specific IP addresses). The size of a node indicates the total number of interactions involving it as either source or target. The color of a node encodes its type as inside this subnetwork, outside, or unknown. Similarly, links represent different types of threats/communications between source and target nodes with corresponding color codes for device actions. A loop-back arc means the source and the destination addresses are the same. The thickness of a link signifies the number of communications. Finally, a force-directed graph layout is defined in such a way that it helps to bring nodes with more and similar communications together.

The overview network graph is important as underlying network data tend to grow in terms of size and complexity. Thus, it would not be possible for the analyst to go through every detail with an overwhelming number of hosts and communications within the network. However, while examining critical events, finding out threat patterns, and predicting trends, the analyst needs to drill down to temporal details. Therefore, we provide a dynamic network [DPF16] on the right side to visualize the dynamic characteristics within the analyzing data. The horizontal axis is restricted

by time constraints. Thus, we also use a force-directed layout to bring related nodes together in the vertical orientation.

In this specific application of DualNetView to the network dataset, horizontal lines represent participant network IPs in this snapshot. Similar to the network graph on the left side, these lines are also color-coded by its origin. Also, arrow-headed arcs represent communications between nodes and the color and thickness encoding are the same as described previously. Links connect these IPs are vertically positioned at the timestamp when the event happens. Furthermore, though force-directed layout helps to reduce the cluttering issues, the dynamic network is still cluttered in cases where there is a large number of nodes and communications involved (e.g., in cases of Denial of Services attacks or network scanning process). Therefore, we provide a mechanism to group nodes and interactions with similar communication patterns.

Specifically, nodes with same node type (i.e., either source or target) and the same communication type (i.e., either ‘permitted,’ ‘empty,’ or ‘none’) happened at the same time are grouped into a single source/target. Furthermore, communications are aggregated for these grouped nodes accordingly. It’s also worth noting that ‘permitted’ and ‘none’ actions are allowed and ignored by the device correspondingly, and ‘empty’ means the related activity is suspicious. For instance, in Figure 1, the red call-out shows a horizontal line that accumulates 85 IPs, and communications to or from them are all similar in this period. Thus, they are aggregated accordingly. The force-directed layout and this grouping and aggregation strategies help in reducing the visual cluttering issue and enable the visualization to bring similar nodes together. These clusters, in turn, assist in identifying patterns and communication trends.

As the number of visual components increases, it is not possible for the analyst to grasp details of individual elements from different views. Therefore, it is crucial to provide a full range of interactions such as brushing, filtering, and especially linking between components. Specifically, when we mouse over a node in the left panel, this node and all its related actions are highlighted in the dynamic network on the right. Furthermore, a table at the bottom (not shown in Figure 1 due to space limitation) displays information about all related source/target and communication details. Similarly, when we mouse over a link, source and target IPs are highlighted, and details about the underlying interactions are shown at the bottom table.

## 4. Use cases

### 4.1. Computer Network Communications

A natural application of network analysis is for analyzing computer networks. Therefore, we applied our solution to investigate one day (starting on July 30, 2013, 3:00 PM) of network threat log data from a subnetwork of an organization. This one day snapshot contains 3,448 network communications (with three main device actions as ‘permitted’, ‘empty’, and ‘none’) among 157 unique devices (i.e., unique IPs) inside the subnetwork, one IP (‘64.82.108.2’) is from outside, and the rest of the source addresses are ‘unknown.’

It is observable, from the left panel of Figure 1, that there are four clusters of devices inside the network and two suspicious addresses (‘64.82.108.2’ and ‘unknown’). Two clusters of computers

(clusters 1 and 2) were attacked by suspicious IPs. Clusters 3 and 4 are the computers being attacked by unknown sources and also had ‘permitted’ or ‘permitted’ and ‘none’ loop-back actions correspondingly. These clusters are easily perceivable from the left panel, which shows the usefulness of the force-directed layout in forming clusters as well as giving indications for suspicious events. However, the static graph layout lacks temporal details about the interactions. On the other hand, the dynamic network on the right reveals the temporal attack pattern. It is visible that from 3:00 PM, the ‘64.82.108.2’ address started to attack the network massively. Shortly after that, unknown sources also began to attack. These attacks are assumed to be the scanning process that suspicious attackers were using to look for vulnerable devices inside the network.

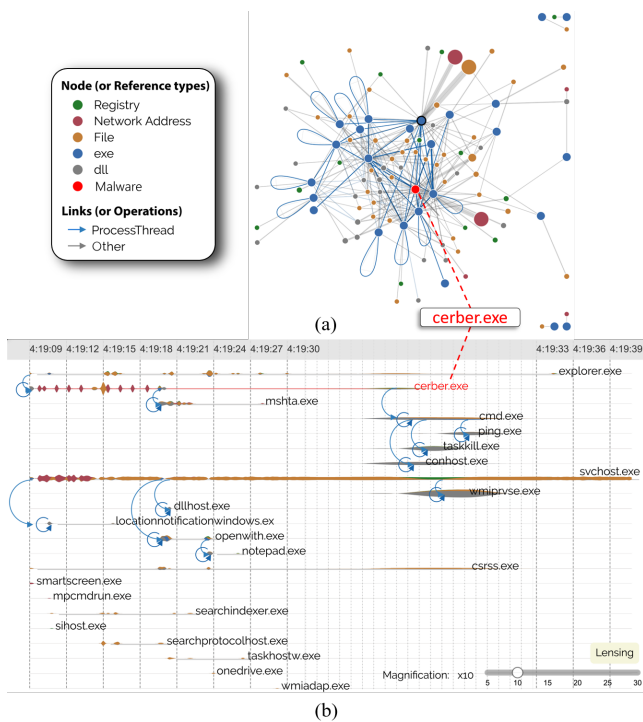
After this massive scanning process, from about 6:00 PM onward, unknown sources periodically communicated with a group of 85 computers. These 85 computers were discovered by the attackers as vulnerable and were being exploited by unknown sources. Specifically, mousing over the attacking arcs from unknown sources to the group of vulnerable IPs and inspecting the attack details in the table at the bottom reveal that, from 5:39 PM, unknown sources attacked this group once every hour until the next day. All in all, this use-case demonstrates the utility of the dual view approach. One helps to give an overview and indicate suspicious events and clusters while the other is for examining the temporal patterns. The timeline view is useful but also has a limitation: it is harder to track/follow paths. For example, it is not trivial to visually discern the degree of separation between ‘64.82.108.2’ and ‘unknown’ in the timeline, which is evident in the force-layout on the left of Figure 1.

### 4.2. Malware Analysis

We also applied DualNetView to analyzing the dynamic behaviors of computer malware. The malware execution trace was generated by running it (i.e., ‘cerber.exe’ in this case) in a sandbox environment. We used Windows Process Monitor [Mic] (*ProcMon*) to capture the run time behavior of this malware. Furthermore, we categorized the participating entities into six node types (i.e., ‘registry’, ‘network address’, ‘file’, ‘exe’, ‘dll’, and ‘malware’). Due to a large number of the action types within the execution traces, we categorized the monitored actions into *Process* and *Thread*-related activities (as they are important for malware analysis) into one group (called ‘ProcessThread’) and all others into another category (called ‘Other’).

Figure 2 shows DualNetView visualization applied for malware analysis. It is noticeable that *ProcMon* captured numerous background processes running in the system. Hence, there is a large number of entities and communications that happened during the execution time of this malware. In this case, though the static network graph, panel (a), seems to be overwhelmed with nodes and links, the busier nodes (e.g., *svchost.exe* and *cmd.exe*) and nodes related to the malware (at the red circle) are brought to the middle of the graph. This behavior helps to narrow down the entities, which are most related to the malware activities, to be examined in the dynamic network graph. In Figure 2(b), we can see the series of process creations starting with *cerber.exe* to *cmd.exe* on the exact

timestamps which have been expanded using the lensing effect to zoom into details on demand.



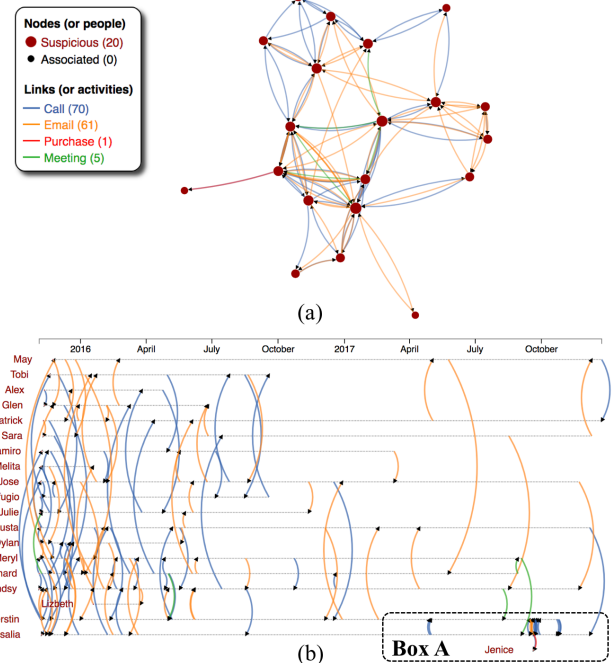
**Figure 2:** Network analysis for the Cerber.exe malware: (a) Standard network view (b) Timeline view focusing on the process-related activities. Blue arcs are the process creations while loops are self-calls.

### 4.3. Social Network Analysis

For this social network analysis case study, we use the data from VAST 2018 - Mini Challenge 3 dataset [DP18]. We have filtered and focused on the 30 Kasios International employees (vertices in the network) as identified by the insider. Links are color-coded by activity category, such as calling, emailing, buying chemical products, and meeting) occupy lower positions in the network. Figure 3 shows the dual views: Traditional force-directed node-link diagram is on the top while the timeline view is at the bottom. Even with the small network of 20 vertices, no evident patterns can be visually discerned in Figure 3(a) due to the complex nature of the communications [BVB\*11, BBW12]. In the timeline view of the same data, each horizontal line represents an employee's participating time. Color encoding allows us to view the type of relationships between nodes.

In Figure 3(b), there are many activities at the end of 2015 and the beginning of 2016. The interactions in this group are significantly reduced and replaced by smaller cliques. For example, there are many calls and emails exchanged between the three people in Box A. Within this group, *Kerstin* made many phone calls to *Rosalia*, who later performed the suspicious purchase transaction with

*Jenice* (the seller). The overlapped arcs (on their associated timestamps) make these activities easily discerned in the overall timeline. These can be considered as the temporal hot spots in our visual layout.



**Figure 3:** Activities of 20 Kasios employees (red vertices/names): (a) Network view - Node sizes are computed based on their activities (b) Timeline view - Each horizontal line presents a timeline of the employee whose name is printed on the left.

### 5. Conclusions

This paper presents a dual view framework for dynamic network visualization. The first view utilizes the standard network view to give the overall network structures, such as communities, influential entries in the network, network bridges, and degree of separations between two vertices. The second view focuses on the temporal details where connections are projected horizontally by their timestamps. Our DualNetView integrates multiple coordinated views into a single snapshot. The supplement views support linked operations on user interactions, such as filtering, grouping/expanding, and brushing. The tool is applied to three case-studies in the application domains of computer networks, malware analysis, and social networks. In future work, we will investigate the scalability of the system and recommend suitable operations (such as adaptive aggregations) for navigating and exploring large and dynamic networks at different level granularities. DualNetView is implemented as a JavaScript-based web application using D3.js [BOH11]. The source codes, demo video, and the web application of our visualization are available on our Github project at <https://idatavisualizationlab.github.io/V/Threats/>.



## References

- [AAAAS13] ALSALEH M., ALQAHTANI A., ALARIFI A., ALSALMAN A.: Visualizing phpids log files for better understanding of web server attacks. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security* (2013), pp. 1–8. 2
- [BBW12] BURCH M., BECK F., WEISKOPF D.: Radial edge splatting for visualizing dynamic directed graphs. In *Proc. Int. Conf. on Information Visualization and Applications* (2012), pp. 603–612. 4
- [BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D<sup>3</sup> data-driven documents. *IEEE Transactions on Visualization & Computer Graphics*, 12 (2011), 2301–2309. 2, 4
- [BVB\*11] BURCH M., VEHLLOW C., BECK F., DIEHL S., WEISKOPF D.: Parallel edge splatting for scalable dynamic graph visualization. *IEEE Trans. Vis. Comput. Graph.* 17, 12 (2011), 2344–2353. 4
- [CG18] COSTA J. P., GRBAC T. G.: A methodology to evaluate the evolution of networks using topological data analysis. In *Data Mining and Data Warehouses-SiKDD* (2018). 1
- [CSBG18] CORDEIRO M., SARMENTO R. P., BRAZDIL P., GAMA J.: Evolving networks and social network analysis methods and techniques. *Social Media and Journalism: Trends, Connections, Implications* (2018), 101. 2
- [DMF17] DANG T., MURRAY P., FORBES A.: Biolinker: Bottom-up exploration of protein interaction networks. In *2017 IEEE Pacific Visualization Symposium (PacificVis)* (April 2017), pp. 265–269. 2
- [DN18] DANG T., NGUYEN V. T.: Comodeler: Topic modeling using community detection. In *Proceedings of the EuroVis Workshop on Visual Analytics* (Goslar, DEU, 2018), EuroVA '18, Eurographics Association, p. 1–5. 2
- [DP18] DANG T., PHAM V. V.: Timematrix: Visual representation for temporal pattern detection in dynamic networks, vast 2018 mini-challenge 3. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2018), pp. 108–109. 4
- [DPF16] DANG T. N., PENDAR N., FORBES A. G.: Timearcs: Visualizing fluctuations in dynamic networks. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 61–69. 2
- [GBD09] GREILICH M., BURCH M., DIEHL S.: Visualizing the evolution of compound digraphs with timearctrees. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 975–982. 2
- [GFS\*15] GUIMARAES V. T., FREITAS C. M. D. S., SADRE R., TAROUCO L. M. R., GRANVILLE L. Z.: A survey on information visualization for network and service management. *IEEE Communications Surveys & Tutorials* 18, 1 (2015), 285–323. 2
- [GS11] GRÉGIO A. R. A., SANTOS R. D. C.: Visualization techniques for malware behavior analysis. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense X* (2011), Carapezza E. M., (Ed.), vol. 8019, International Society for Optics and Photonics, SPIE, pp. 9–17. 2
- [IES\*12] INOUE D., ETO M., SUZUKI K., SUZUKI M., NAKAO K.: Daedalus-viz: novel real-time 3d visualization for darknet monitoring-based alert system. In *Proceedings of the ninth international symposium on visualization for cyber security* (2012), pp. 72–79. 2
- [KKL\*14] KIM U.-H., KANG J.-M., LEE J.-S., KIM H.-S., JUNG S.-Y.: Practical firewall policy inspection using anomaly detection and its visualization. *Multimedia tools and applications* 71, 2 (2014), 627–641. 2
- [MDS13] MATUSZAK W. J., DIPIPPO L., SUN Y. L.: Cybersave: situational awareness visualization for cyber security of smart grid systems. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security* (2013), pp. 25–32. 2
- [MGC12] MANSMANN F., GÖBEL T., CHESWICK W.: Visual analysis of complex firewall configurations. In *Proceedings of the ninth international symposium on visualization for cyber security* (2012), pp. 1–8. 2
- [Mic] MICROSOFT DOCS: Process Monitor - Windows Sysinternals - Microsoft Docs. <https://docs.microsoft.com/en-us/sysinternals/downloads/procmon>. Accessed: 2020-03-02. 3
- [MMK08] MANSMAN F., MEIER L., KEIM D. A.: Visualization of host behavior for network security. In *VizSEC 2007*. Springer, 2008, pp. 187–202. 2
- [NK13] NOVIKOVA E., KOTENKO I.: Analytical visualization techniques for security information and event management. In *2013 21st Euro-micro International Conference on Parallel, Distributed, and Network-Based Processing* (2013), IEEE, pp. 519–525. 2
- [OG07] ONUT I.-V., GHORBANI A. A.: Svision: A novel visual network-anomaly identification technique. *computers & security* 26, 3 (2007), 201–212. 2
- [PD18] PHAM V., DANG T.: Cvexplorer: Multidimensional visualization for common vulnerabilities and exposures. In *2018 IEEE International Conference on Big Data (Big Data)* (Seattle, WA, USA, USA, Dec 2018), IEEE, pp. 1296–1301. 1
- [PD19a] PHAM V., DANG T.: Outliagnostics: Visualizing temporal discrepancy in outlying signatures of data entries, 2019. 1
- [PD19b] PHAM V., DANG T.: SOAViz: Visualization for Portable X-ray Fluorescence Soil Profiles. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)* (Porto, Portugal, 2019), Bujack R., Feige K., Rink K., Zeckzer D., (Eds.), The Eurographics Association. 1
- [PNL\*19] PHAM V., NGUYEN N., LI J., HASS J., CHEN Y., DANG T.: Mtsad: Multivariate time series abnormality detection and visualization. In *2019 IEEE International Conference on Big Data (Big Data)* (2019), IEEE, pp. 3267–3276. 1
- [RTJ\*11] REDA K., TANTIPATHANANANDH C., JOHNSON A., LEIGH J., BERGER-WOLF T.: Visualizing the evolution of community structures in dynamic social networks. In *Proc. Eurographics Conf. on Visualization* (2011), pp. 1061–1070. 1
- [SBD\*20] SUN F., BAKR N., DANG T., PHAM V., WEINDORF D. C., JIANG Z., LI H., WANG Q.-B.: Enhanced soil profile visualization using portable X-ray fluorescence (PXRF) spectrometry. *Geoderma* (2020). 1
- [TEK\*13] TAKAHASHI T., EMURA K., KANAOKA A., MATSUO S., MINOWA T.: Risk visualization and alerting system: Architecture and proof-of-concept implementation. In *Proceedings of the first international workshop on Security in embedded systems and smartphones* (2013), pp. 3–10. 2
- [VBAW15] VEHLLOW C., BECK F., AUWÄRTER P., WEISKOPF D.: Visualizing the evolution of communities in dynamic graphs. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 277–288. 2
- [VBSW13] VEHLLOW C., BURCH M., SCHMAUDER H., WEISKOPF D.: Radial layered matrix visualization of dynamic graphs. In *Proc. Int. Conf. on Information Visualisation* (2013), pp. 51–58. 2
- [vdEHBvW13] VAN DEN ELZEN S., HOLTEN D., BLAAS J., VAN WIJK J. J.: Dynamic network visualization with extended massive sequence views. *IEEE transactions on visualization and computer graphics* 20, 8 (2013), 1087–1099. 2
- [WPO14] WUCHNER T., PRETSCHNER A., OCHOA M.: Davast: Data-centric system level activity visualization. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security* (2014), pp. 25–32. 2
- [WQY\*12] WANG K., QI Y., YANG B., XUE Y., LI J.: Livesec: Towards effective security management in large-scale production networks. In *2012 32nd International Conference on Distributed Computing Systems Workshops* (2012), IEEE, pp. 451–460. 2
- [ZN12] ZHUO W., NADJIN Y.: Malwarevis: entity-based visualization of malware network traces. In *Proceedings of the ninth international symposium on visualization for cyber security* (2012), pp. 41–47. 2
- [ZTZ\*19] ZHAO H., TANG W., ZOU X., WANG Y., ZU Y.: Analysis of visualization systems for cyber security. In *Recent Developments in Intelligent Computing, Communication and Devices*. Springer, 2019, pp. 1051–1061. 2