VITALflow: Visual Interactive Traffic Analysis with NetFlow

Tina Tremel*, Jochen Kögel*, Florian Jauernig*, Sebastian Meier*,
Dennis Thom†, Franziska Becker†, Christoph Müller†, and Steffen Koch†
*IsarNet Software Solutions GmbH, †University of Stuttgart

***Abstract*—Traffic analysis in large enterprise networks has become a vital task for network experts, as understanding application and user traffic is the basis for proper network management with respect to planning, formulating intents, or analyzing causes for implausible behavior. In such networks, NetFlow provides input to network monitoring systems that typically show time series visualizations along different data dimensions. We studied tasks and requirements of network experts and derived a visual analytics approach that improves their analytic workflow as it enables for exploration of large time spans quickly in a multidimensional manner. Our approach guides users and improves the scalability of analyses through a novel combination of a clustered time series view and filtering in interactive parallel sets into a coherent visual analysis framework. Clustering reveals typical patterns and deviations from the daily norm and serves as entry point to exploring, filtering and comparing multiple dimensions in the parallel sets view. In addition, we briefly discuss the feedback we received on two case studies with network experts.**

I. Introduction

Enterprises have a strong interest in monitoring their network traffic for network management and for resolving issues. Providing a sound view on long-term network traffic developments and relations of their properties via NetFlow [1] is vital for proper planning, operation and troubleshooting. The amount of NetFlow data collected and processed is enormous, even for medium-sized networks and short periods of time. Therefore, it comes at no surprise that interactive visualization approaches are common and also widespread in commercial products for analyzing NetFlow data. These approaches typically employ different strategies to deal with the large volumes of data, such as limiting interactive analysis to short time periods. This allows visualizing the data in detail and avoids scalability problems during interactive exploration. Other approaches focus on a specific task like the identification of security threats [2]. When investigating a specific security incident, filtering for the relevant devices also reduces the amount of data that need to be processed for interactive analysis.

In contrast, providing practitioners with a holistic view of large amounts of NetFlow data for general planning and operational purposes is desirable, but difficult to achieve, specifically with no explicit task to optimize for. Even more so, if experts are to be equipped with the means to explore data interactively. In this paper, we present *VITALflow* (see Fig. 1), a new approach specifically targeting these kinds of open-ended analyses and troubleshooting tasks. Based on interviews with domain experts, who manage large enterprise networks on a daily basis, we identified what practitioners want to achieve during such analyses (section III). From these tasks, we generalized conceptual and technical requirements. With VITALflow, we contribute a visual analytics approach (section IV) that advances the scalability of visual interactive NetFlow analysis by introducing a novel combination of visualization techniques that support an iterative analysis workflow at the level of aggregated data. This workflow is characterized by combining clustering to reveal patterns and deviations from those in daily time series with parallel sets to investigate details. User-steering of the clustering allows for refining the clusters to isolate different temporal behavior based on prominent time spans. Interactions in the parallel sets then provide the means to explore these clusters with respect to multiple dimensions like endpoints, protocols or IP ranges. In particular, comparing different clusters and their median days enables analysts to quickly reason about possible problems and their causes. We implemented our approach as a prototype (section V) and discuss feedback from expert interviews (section VI).

978-1-6654-0601-7/22/$31.00 ©2022 European Union

II. Related work

Visualization approaches for network management [3] and traffic analysis [4] tasks typically combine a variety of visual methods to deal with the complexity of network data. We discuss such approaches in the following and delineate our approach from previous ones that make use of similar techniques or provide similar analysis workflows.

Visual methods are often used to understand traffic relations or characteristics in computer networks. Frequently, these methods focus on security incidents, and a multitude of techniques has been proposed to highlight connections between attacked devices/networks and attack sources. Related works have used color to indicate potentially malicious or unsecured connections using list views [5] and node-link diagrams [6]. It is also common to use hierarchical views to represent IP address spaces and subnets. Such views include tree maps [7], quad-/octree representations [8], radial (hierarchical) representations [9], variants of parallel coordinate plots [10] with enhanced concepts for including hierarchies, and combinations of these. Line and bar charts as well as stacked graphs are commonly used in network traffic analysis to depict the temporal fluctuation in the amount of traffic, system health information, etc. Published approaches for analyzing NetFlow data [11], [12] rely on such

diagrams. This holds also for VITALflow and the IsarFlow software [13], which we use as the starting point of the requirement analysis for our approach.

Network data often has a high temporal resolution, which can cause scalability problems. Solutions have been proposed for a variety of domains and purposes, such as lens-based interaction [14] or layered techniques [15]. Using scatter plots instead of line charts [16] can mitigate visual clutter at the cost of missing daily cohesion. VITALflow shows aggregated NetFlow information (number of flows/bytes/packets) in a superimposed way on a daily basis. It is based on the idea by van Wijk et al. [17]. They proposed clustering daily times series to identify patterns, which can then be explored in a calendar view. VITALflow combines clustering of daily time series with a visual alignment of days to clusters.

Multiple coordinated views [18] are a common approach to understand and analyze multiple dimensions of computer networks [19]. These dimensions include, e.g., protocols, Quality of Service (QoS) levels as well as source and destination information. In addition to coordinated views (e.g. [20]), other techniques for multi-dimensional data visualization have been applied to achieve a higher integration of this information within views for analyzing log data. A considerable number of approaches, such as [10], is using parallel coordinates [21] to achieve this, while we employ the parallel sets approach [22]. Chen et al. [10] integrate the temporal aspect as an axis of their parallel coordinates view. Other solutions outside the field of computer network analysis have been proposed that let users select the axes of a parallel coordinates plot between which pseudo-perspective views of time series plots are integrated [23]. In contrast, we propose a hybrid solution that uses linked views and a parallel sets for analyzing multivariate data.

III. Typical tasks and visualization requirements

The design of VITALflow is the result of discussions with four network experts who have 15–30 years of experience. Two of them work in the network operations department of a large enterprise and use monitoring systems extensively. The other two design and customize network monitoring solutions at a network monitoring company. Hence, all are familiar with different state-of-the art network monitoring tools that predominantly use linear time series views with stacked charts.

The prototypical tasks Tx the experts perform with traffic analysis tools are the **search and analysis of anomalous behavior (e.g. peaks) (T1)**, which is done routinely to identify looming problems early, **troubleshooting (T2)** of specific user problems, the **analysis of change impact (T3)**, which compares traffic before and after a change (e.g. routing metrics, server locations) and the **identification of typical traffic behavior (baselining) (T4)** before adjusting network capacity. From these tasks, we generalised several conceptional (R_Cx, see Table I) and technical (R_Tx) requirements.

As a first requirement, we aim at providing **support for network experts (R_C1)** as all tasks require technical expertise and knowledge of the network. This also mandates at least basing visualizations on concepts experts are familiar with, most notably supporting **time-centric analyses (R_C2)**. For instance, experts use a visual representation of time series to identify anomalies in T1 in the first place. They filter the specific point in the network (e.g. a particular WAN port) and the time span for which the user reported problems in the troubleshooting task (T2). Then they compare the problematic time frame with previous weeks to identify recurring patterns correlating with issues on the user's side. The baselining task (T4) requires even longer time frames for identifying trends.

TABLE I
Summary of conceptual requirements R_Cx

Requirement R_Cx	T1 anom.	T2 troubls.	T3 change	T4 basel.
1: for network experts	tech	support	plan/tech	plan
2: time-centric analyses	+	+	+	+
3: characteristic shapes	+	+	-	+
4: temporal comparison	peaks	issue	bef./aft.	long-t.
5: large time spans	days	weeks	days/weeks	weeks
6: exploratory workflows	+	+	+	-
7: multiple dimensions	prot.	subnets	subnets	prot.
8: interactive filtering	+	+	change	basel.

Experts have learned to visually identify **characteristic shapes (R_C3)** of problems. For instance, problems causing poor performance (T2) manifest as plateau shapes indicating bandwidth saturation, whereas thin spikes indicate short packet bursts that might cause forwarding issues. Furthermore, the typical daily pattern is a characteristic shape on a larger time scale and deviations might indicate a problematic anomaly (T1). Obviously, any new visualisation for network experts should retain their capability to rely on such learned shapes.

The ability to perform a **temporal comparison (R_C4)** is important for all tasks. For the analysis of change impact (T3), comparison of client subnet and protocol relations before and after the change is the actual task. This might reveal that after migrating a server between data centers, these are well balanced, but at the cost of higher utilization of expensive WAN links. Considered time spans differ by task. E.g. open search for anomalies (T1) compares a day with several previous ones. Identifying sporadic periodic performance problems (T2) and even more the baselining of network traffic (T4) requires an **analysis of large time spans (R_C5)** up to weeks or months.

Tasks 1 to 3 demand for an **explorative workflow (R_C6)**. E.g., users need to find the cause of a peak in bitrate by checking different dimensions of the traffic (sessions, QoS markings, protocols) to eventually find explanations like excessive DNS traffic or simply a high volume of legitimate HTTPS. All tasks require the **comparison of multiple dimensions (R_C7)**. Table I shows the most notable dimensions per task as a non-exclusive list. And finally, all tasks need **interactive filtering (R_C8)** in order to efficiently explore the large amounts of data collected.

Technically, we have to deal with **large amounts of data (R_T1)** and records represent **multi-dimensional data points (R_T2)** in time. Most dimensions exhibit a **high cardinality (R_T3)** with hundreds or thousands of categories. Despite high data volumes, the solution must provide a **low response time (R_T4)** for interactive use.

Fig. 1. VITALflow's three layer design: The timeline view (a) in Layer 1 shows time series of NetFlow data superimposed for each day, from which clusters or single days can be selected for investigation in the parallel sets view (b) of Layer 2. Here, traffic characteristics can be compared and filtered while the context views (c, d) in Layer 3 directly provide temporal context for the filtering in place. The time range, clustering, etc are configured in the global settings panel (e).

IV. Visual Analytics Approach

For the overall user interface, we chose an opinionated design approach, which organizes the workflow (R_C6), enabled by interactive filtering (R_C8), into three distinct layers (see Figure 1). Networks experts (R_C1) typically start their analysis given a rough time period, customer network/VPN and quantity of interest (bytes, flows, or packets). We refer to this starting point as the *global settings*, which are configurable in the respective panel (Figure 1 e).

Compared to state-of-the-art time series and drill-down based network monitoring tools, which often allow for freely adding and arranging views, VITALflow shows all views all at once. This design decision aims to support users in understanding interaction and filtering impact easily while also starting off with a highly aggregated view to cope with the huge amount of data (R_T1). Arrows between layers further emphasize the primary workflow. Although our approach encourages a top to bottom workflow, multiple iterations from top to bottom are possible, and the user is free to re-examine the data set, e.g. for different days or clusters of interest. Iterative workflows within the layers are supported as well, e.g. users can tune the clustering – among others by adjusting the number of clusters – or simplify the selection of outliers by reclustering in the timeline view for defined time spans (see video [24]).

A. Layer 1: Superimposed time series clusters

The time-based overview of Layer 1 is the starting point of any analysis and gives an overview for the quantity and time frame specified in the global settings. Its main component is a line chart indicating the measurement of the respective unit (y-axis) during a day (x-axis). The daily time series are clustered and visualized as superimposed lines, enabling comparison (R_C4) of patterns at high temporal resolution similar to van Wijk et al. [17]. We use the agglomerative clustering algorithm from [17] (k-means as opt-in) with Euclidean distance as measure between time series. Since daily traffic patterns are rather regular in large company networks, deviations from such regular behavior are good starting points for exploration and detailed inspection. We found that clustering using Euclidean distance clearly separates typical daily patterns and deviations from them for our data sets. Because of the networks' regular behaviour, using five clusters as a default turned out to be a reasonable choice for capturing regular daily patterns (e.g., working days, weekends, bank holidays) and the most salient outliers for time spans of weeks up to a few months. The visual representation of clusters differs from single days in that the cluster coherence is indicated as a shaded area or band of the same color around the average line of the cluster.

Especially for peak comparison (R_C4), providing context in form of typical daily patterns is essential. For example, Figure 2 illustrates how VITALflow provides a scalable visualization for large time frames as the peak at 2:00 is present on all days, whereas the blue peak at 21:00 represents unusual behavior. In contrast to van Wijk et al., we use circular glyphs (symbols) for days below the line graph using color to indicate cluster membership instead of a calendar-based representation. Glyphs with an additional circle indicate the median day of a cluster. For browsing through daily patterns, hovering over the glyphs immediately highlights the respective time series of that day. Interactive time-based filtering in Layer 1 can be accomplished in different ways: First, complete clusters or median days can be selected via the legend. Second, interaction with the glyphs

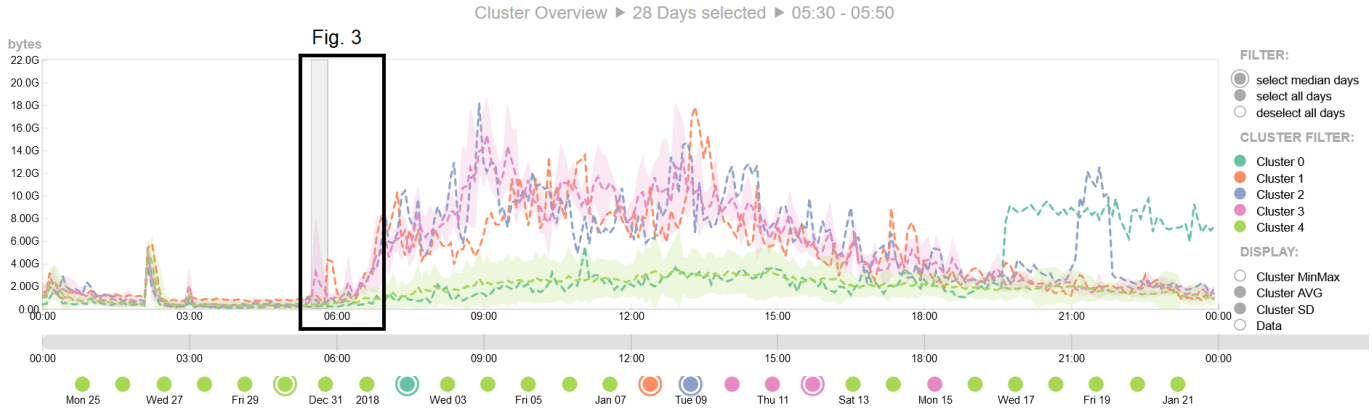

Fig. 2. Clustered time series of total bytes per 5 min interval for 28 days. The x-axis shows the time for the superimposed time series of each day. Similar time series are assigned to clusters indicated by color. Average values per cluster are shown by a dashed line and standard deviation by colored area, if the cluster contains more than one day. The grey area indicates the time filter for the next layer.

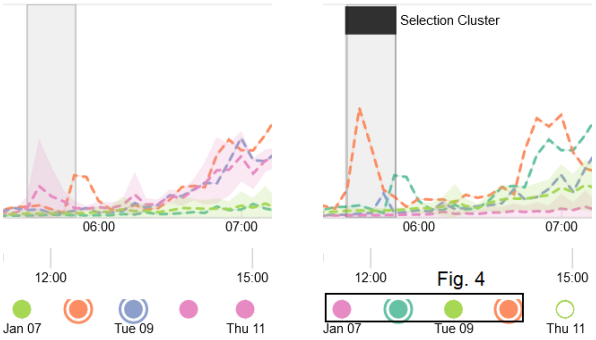

Fig. 3. By default, clustering is applied to the entire time frame (left). For smaller time intervals, global clustering can be suboptimal, which is indicated by a large violet SD band around the selection. On the right, reclustering has been triggered for the selected time interval revealing the orange peak.

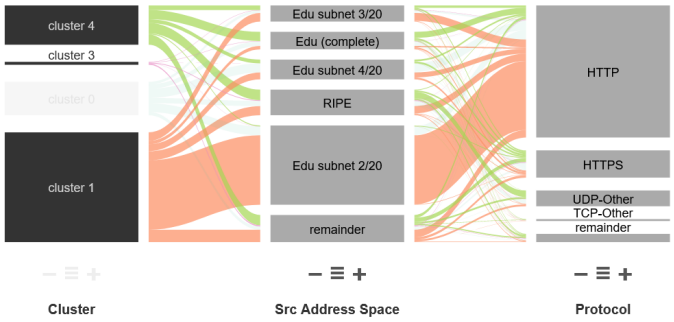

Fig. 4. Layer 2 provides a multidimensional overview of the time selection using parallel sets. Here, the time and days of clusters from Figure 3 right have been selected. The clusters from Layer 1 are the left-most dimension using the same cluster colors, providing a direct visual link between the layers. The view shows the traffic pattern of the orange peak (Cluster 1) compared to the other days. Dimensions can be interactively reordered, added or deleted allowing for comparison regarding originating address space and protocol.

below the chart allows for single day selection. Third, a time frame can be selected by brushing [25] a region in the line chart, as indicated in Figure 2 by the grey area at 5:30.

Typically, network experts want to compare traffic volumes of the same time interval length, i.e. single days with single days and peak hours with peak hours. According to early experiments comparing a cluster of seven days with one containing a single day is difficult. By selecting median days for comparison instead of the whole cluster, the problem is simplified to comparing two days or the same time span of two days.

Our clustering is adjustable in several ways via the global settings panel. First, we allow switching the clustering algorithm between hierarchical clustering and k-means. Second, the number of clusters can be customized. In addition, we allow for user-steered selective reclustering as shown in Fig. 3. Reclustering is based on the current selection and minimizes the error in time frame of interest, which is also useful for analyzing shorter peaks (R_C4). Once the desired clustering has been found and the desired time – both in terms of days and time of day – has been selected, the filtered data are loaded into Layer 2, which provides a multidimensional overview.

B. Layer 2: Parallel sets comparison and filtering

The parallel sets view shown in Figure 4 serves three purposes: multi-dimensional data browsing (R_C7), interactive filtering (R_C8), and providing a direct visual link between time series and clusters. We chose parallel sets as it is one of the few visualization techniques that work well for categorical data, such as protocols in our case. This also suits the aggregated data, which is binned into categories and still allows for depicting the measured quantities via the width of connections between categories in different dimensions. Due to the high cardinality of our data (R_T3), we adapted the original parallel sets technique [22]: First, we use a horizontal layout with spacing and minimum category height for improved label readability and decreased height usage. Second, we initially only display the top-5 categories with respect to volume of the selected quantity (initially bytes). This does not only improve

visual scalability, but reflects the fact that the causes for high traffic volumes are often the primary concern.

By default, the parallel sets view displays the *cluster* dimension leftmost (Figure 4). This provides a direct visual link to Layer 1, as the first dimension defines link colors, which then directly match the colors of the clusters in Layer 1. From left to right, the distribution of different traffic properties of the clusters is directly visible for the selected traffic. The axes can be reordered and dimensions can be added or removed to support exploratory workflows (R_C6) for multiple dimensions (R_C7). Furthermore, each dimension can be explored in more detail as the initial top-5 selection can be changed by adding or removing categories from the remainder.

The interactive dimension filter supports brushing, similar to axis brushing in parallel coordinates. We support the removal of single elements or entire dimensions and additionally allow for retaining context by displaying excluded categories in a lighter color instead of removing them from the view. The fraction of filtered data is directly shown in the percentage circle left of the parallel sets view . Switching to *filtered data* completely hides the deselected categories and allows for detailed assessment of the relevant categories only, while switching back to *all data* or even the *complement* provide novel means for both: iterative data understanding and filter tuning.

C. Layer 3: Time-based context views

Layer 3 returns to a familiar (R_C1) time-centric view (R_C2) while the parallel sets view of Layer 2 focussed on multi-dimensional data browsing. It comprises two views: a stacked graph view showing traffic volumes of specific filtered data on a timeline and a filter summary showing the filtered time series. Both the time-based filters of Layer 1 and the dimensional filters of Layer 2 apply to Layer 3. In contrast to the timeline view, the time axes used here use a normal timeline for the data, providing the user with additional context.

Whenever the user hovers over a column or node of the parallel sets, the corresponding traffic volume is highlighted in the time-based stacked graph. The remaining traffic is drawn as a silhouette, allowing for assessing the impact of the active filters. To work with large time frames (R_C5) and focus on a particular range, the stacked graph view provides a horizontal scroll bar and temporal zoom via brushing. In addition, users can zoom in vertically using the mouse wheel (R_C3), to view data that is not easily discernible due to its drawing area. The filter summary provides the same visualization approach as used in Layer 1, but differs by only showing the data based on the filtering criteria applied in Layer 1 and Layer 2. This facilitates a visual summary of the displayed time period and dimensions with temporal comparison (R_C4).

V. Prototype Implementation

Figure 5 illustrates the overall architecture of VITALflow. The flow monitoring system (left) collects NetFlow data and adds dimensions by enrichment, such as protocol and subnet information. The data is pre-aggregated keeping only relevant dimensions to be stored in a distributed columnar database (DB)

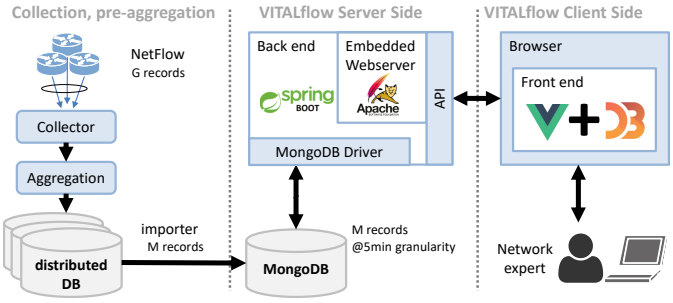

Fig. 5. Overview of the VITALflow system architecture and magnitude of data processed for our data sets denoting billions (G) and millions (M) of records processed.

from where it is imported to MongoDB. Periodic jobs further pre-aggregate data per dimension in MongoDB to improve query speed (R_T4) for the huge amount of data (R_T1).

The back end (middle of Figure 5) interacts with MongoDB, including query formulation, pre-processing, and conversion of results to JavaScript objects. The application is deployed on an integrated Apache Tomcat server that also serves the client-side JavaScript front end. Furthermore, the back end handles all aspects of the cluster analysis, including pre-computation of standard deviations, mean-time series and median-time series, and performs some caching. For some aspects like the k-means and hierarchical clustering algorithms, we use the Java-ML [26] library. The front end communicates with the back end via a Web Application Programming Interface (API). This API is not stateless, but the back end keeps a session for every connected client, which avoids costly recomputations. The web-based UI is built using Vue.js with D3 visualizations.

VI. Evaluation and Discussion

We carried out two case studies with practitioners using data from a university (also used in figures) and from an enterprise network (main study). The main study is available online in detail [24] and briefly summarized in the following. We evaluated VITALflow with three experts from the network operations department of an IT solution provider with 15–30 years of experience. We presented the timeline view of eight weeks of data from their network to the experts in individual sessions and asked them to find deviations from expectations and their causes in a pair analytics study. Furthermore, we asked them to answer a series of questions. All experts found the same deviations and causes from the visualizations while the interviewer provided hints on navigation and controls.

The case studies indicate that VITALflow's novel combination of clustered time series and interactive parallel sets aligns well with the tasks and requirements of network experts described in section III. The implemented clustering method delivered a comprehensible starting point in time series evaluation by drastically reducing the number of lines and highlighting the most important anomalies at the same time. It worked well for both data sets and in combination with the possibility to quickly compare days using hovering,

it could provide hints to the users for further filtering or re-clustering actions. While the experts expressed generally positive feedback and the wish to have the approach integrated into their production software, there is naturally room for improvement, which we want to discuss here. First, we currently do not support filtering based on network regions in or before Layer 1, which the experts explicitly wished for. While this is easy to implement, it adds additional complexity to the workflow: different filtering could happen before Layer 1 and in Layer 2, which might be confusing. Nevertheless, such a feature would be useful when problems can be isolated to a specific region of the network right from the start, e.g. as part of task T2. Second, our solution is derived from the tasks described in section III and as such tailored towards those. Therefore, it lacks features typically used in security analyses, which are a different application area for NetFlow. Adding additional dimensions such as IP addresses ("top talkers"), port numbers or TCP flags and access to raw data requires only minimal effort, but additional application scenarios will most likely also mandate changes to the workflow itself.

Currently, the back end uses only a single database instance to process requests, which can provide the requested data within a few seconds (R_T4). This turned out to be satisfactory for the case study, and scaling out MongoDB or building the backend on top of the existing distributed columnar database (Figure 5) would allow us to deal with larger data sets (R_T1) or more dimensions (R_T2) and higher cardinality (R_T3).

VII. Summary

In this paper, we presented VITALflow, an interactive network traffic analysis tool. VITALflow combines a clustered time series view and filtering in interactive parallel sets into a coherent, powerful iterative analysis workflow to support network experts in performing large-scale analyses of aggregated flows. This workflow and the overall design of the approach have been derived from typical tasks experts have to deal with on a daily basis. We evaluated our solution with network experts using data from an enterprise network to understand how well the experts' needs are met. Positive feedback confirmed that our approach is suitable for assisting network operators in typical troubleshooting scenarios and is able handle the data volumes from a large enterprise network.

Acknowledgments

The authors want to thank Manfred Rosswinkel, Markus Führer and Joscha Ott of DB Systel GmbH as well as André Kosak and Dirk Kurfürst of IsarNet for taking part in interviews and user studies. This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) as part of the Wintermute project under contract No. 16KIS1131 and by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833418.

References

[1] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with netflow and ipfix," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, 2014.

[2] J. R. Goodall, E. D. Ragan, C. A. Steed, J. W. Reed, G. D. Richardson, K. M. T. Huffer, R. A. Bridges, and J. A. Laska, "Situ: Identifying and explaining suspicious behavior in networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, 2019.

[3] V. T. Guimarães, C. M. D. S. Freitas, R. Sadre, L. M. R. Tarouco, and L. Z. Granville, "A survey on information visualization for network and service management," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, 2016.

[4] H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A survey of visualization systems for network security," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 8, 2012.

[5] J. L. Guerra, E. Veas, and C. A. Catania, "A study on labeling network hostile behavior with intelligent interactive tools," in *Proc. IEEE Symp. on Vis. for Cyber Security*, 2019.

[6] M. Schufrin, D. Sessler, S. L. Reynolds, S. Ahmad, T. Mertz, and J. Kohlhammer, "Information visualization interface on home router traffic data for laypersons," in *Proc. Intl. Conf. on Advanced Visual Interfaces*, 2020.

[7] F. Mansmann, F. Fischer, D. A. Keim, and S. C. North, "Visual support for analyzing network traffic and intrusion detection events using treemap and graph representations," in *Proc. Symp. on Computer Human Interaction for the Management of Information Technology*, 2009.

[8] S. T. Teoh, K. L. Ma, S. F. Wu, and X. Zhao, "Case study: Interactive visualization for internet security," in *Proc. VIS*, 2002.

[9] Y. Livnat, J. Agutter, , and S. Foresti, "Visual correlation for situational awareness," in *IEEE Symp. on Information Vis.*, 2005.

[10] S. Chen, C. Guo, X. Yuan, F. Merkle, H. Schaefer, and T. Ertl, "Oceans: Online collaborative explorative analysis on network security," in *Proc. Workshop on Vis. for Cyber Security*, 2014.

[11] L. Braun, M. Volke, J. Schlamp, A. von Bodisco, and G. Carle, "Flow-Inspector: A framework for visualizing network flow data using current web technologies," *Computing*, vol. 96, no. 1, 2014.

[12] B. Li, J. Springer, G. Bebis, and M. H. Gunes, "A survey of network flow applications," *J. Netw. Comput. Appl.*, vol. 36, no. 2, 2013.

[13] IsarNet Software Solutions GmbH. IsarFlow Network Monitoring System. https://isarflow.com/. [Online]. Available: https://isarflow.com/

[14] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan, "Exploratory analysis of time-series with chronolenses," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, 2011.

[15] M. Wörner and T. Ertl, "Smoothscroll: A multi-scale, multi-layer slider," in *Proc. Intl. Conf. on Computer Vision, Imaging and Computer Graphics*, 2011.

[16] P. Velan, J. Medková, T. Jirsík, and P. Čeleda, "Network traffic characterisation using flow-based statistics," in *Proc. NOMS*, 2016.

[17] J. J. van Wijk and E. R. van Selow, "Cluster and calendar based visualization of time series data," in *Proc. IEEE Symp. on Information Vis.*, 1999.

[18] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *Proc. Intl. Conf. on Coordinated and Multiple Views in Exploratory Visualization*, 2007.

[19] J. R. Goodall and D. R. Tesone, "Visual analytics for network flow analysis," in *Proc. Cybersecurity Applications & Technology Conf. for Homeland Security*, 2009.

[20] L. Harrison, X. Hu, X. Ying, A. Lu, W. Wang, and X. Wu, "Interactive detection of network anomalies via coordinated multiple views," in *Proc. Intl. Symp. on Vis. for Cyber Security*, 2010.

[21] A. Inselberg and B. Dimsdale, "Parallel coordinates for visualizing multi-dimensional geometry," in *Computer Graphics 1987*. Springer, 1987.

[22] R. Kosara, F. Bendix, and H. Hauser, "Parallel sets: Interactive exploration and visual analysis of categorical data," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 4, 2006.

[23] H. Gruendl, P. Riehmann, Y. Pausch, and B. Froehlich, "Time-series plots integrated in parallel-coordinates displays," in *Computer Graphics Forum*, vol. 35, 2016.

[24] T. Tremel, J. Kögel, F. Jauernig, S. Meier, D. Thom, F. Becker, C. Müller, and S. Koch, "Supplemental Material for "VITALflow: Visual Interactive Traffic Analysis with NetFlow"," 2022. [Online]. Available: https://doi.org/10.18419/darus-2779

[25] R. A. Becker and W. S. Cleveland, "Brushing scatterplots," *Technometrics*, vol. 29, no. 2, 1987.

[26] Java Machine Learning Library. (2018) Java-ml. [Online]. Available: http://java-ml.sourceforge.net/