CVExplorer: Multidimensional Visualization for Common Vulnerabilities and Exposures

Vung Pham
Dept. of Computer Science
Texas Tech University
Lubbock, TX, USA
vung.pham@ttu.edu

Tommy Dang
Dept. of Computer Science
Texas Tech University
Lubbock, TX, USA
tommy.dang@ttu.edu

***Abstract*—Cyber attacks cause great damage to our national security, ranging from individual internet user to biggest governmental/industrial organizations, such as Equifax (Data Breach 145.5 Million Accounts, reported in July 2017) or Uber (Data Breach 57 Million Records, reported in November 2017). The cyber assault has significantly increased in breadth and depth. This paper introduces *CVExplorer*, a novel interactive system for visualizing cybersecurity threats reported in the National Vulnerability Database. The proposed system aims to work as a reporting and alerting tool that can help enhance the security against cyber attacks can potentially reduce network vulnerabilities. The *CVExplorer* system containing multiple linked views allows users to visualize the relationships of various dimensions in the large number of vulnerability reports, such as types and levels of vulnerability, vendors, and products. The *CVExplorer* provides an intuitive interface and supports a range of interactive features, such as filtering and ordering by vulnerability severity ratings, allowing users to narrow down topics of interest quickly. To demonstrate the effectiveness of the proposed system, we demonstrate the *CVExplorer* on two case studies of Common Vulnerabilities and Exposures retrieved from the National Vulnerability Database.**

***Keywords*-Common Vulnerabilities and Exposures; Common Vulnerability Scoring System; High-Dimensional Visual Analytics; Parallel coordinates; Force directed layouts.**

I. Introduction

With virtually all computing networks and data storage under constant bombardment of cyber attacks and cyber espionage activities, the battlefield of national defense is no longer restricted to military facilities or security agencies. Computing network facilities and data storages in national, industry, academic research labs and offices are all possible targets of cyber attacks. Besides, the popularity of social networking sites and applications can quickly spread the vulnerability from sites seemingly irrelevant to national defense to locations within federal defense facilities. Thus, a network vulnerability analysis, remedy, and alerting tool that can help enhance the security against human-error-utilizing cyber attacks can potentially reduce network vulnerabilities. Even though human error is the most significant cybersecurity vulnerability (such as falling for phishing, unrestrained web browsing, and lousy password habits), the state-of-the-art vulnerability scanners are not designed to detect vulnerabilities introduced by humans interacting with the system [1]. The proposed system aims to fill in this gap.

In particular, we expand the features exposed by vulnerability scanners such as Nessus [2] and Shodan [3] (a kind of "dark" Google) and present vulnerability assessments to users via interactive visual interface instead of dealing with tediously technical outputs. Our analytics system means to provide better understandings of cybersecurity threats and will enable it to provide timely recommendations regarding potential risks via well-documented daily reports from the National Vulnerability Database (NVD), a widely used database containing millions of records about specific device vulnerabilities. The proposed method is implemented in JavaScript embedded in the standard web browsers and potentially extended as a browser plugin for alerting possible cybersecurity threats when users access a site/domain.

Our contributions in this paper thus are:

- We propose a new approach to analyze prominent features in Common Vulnerabilities and Exposures entries through coordinated multiple views. In contrast to existing techniques which mostly look into one dimension at a time, we inspect the relationships of these dimensions for interesting correlations.
- We develop an interactive prototype, named *CVExplorer*, which adopted the standard as well as customized visual representations to explore these relationships in big data. The *CVExplorer* supports a range of interactive features allowing users to narrow down events of interest quickly.
- We demonstrate the *CVExplorer* on two case studies of Common Vulnerabilities and Exposures in 2017 and of an Autonomous System Number.

The paper is structured as follows: We describe related work in the following section. Then we discuss the design motivations and considerations of *CVExplorer* in Section III. We introduce our *CVExplorer* interface and its components in Section IV. We illustrate the use of *CVExplorer* on two case studies in Section V and discuss its limitations and scalability for big data. Finally, we conclude the paper.

II. Related Work

In this section, rather than attempting to survey all cybersecurity visualization, we instead highlight the most related work. Current approaches to vulnerability assessments can be roughly classified into passive and active vulnerability assessments. Passive vulnerability assessment techniques aim to cross-reference system specific characteristics with databases of known vulnerabilities [4], such as the National Vulnerability Database. Techniques belong to this category include p0f [5], PRADS [6], and ShoVAT [7]. In contrast to passive vulnerability assessment, active vulnerability assessment techniques actively probe devices to identify vulnerabilities, including port scanning, checking for SQL injections and HTML injections, monitoring network traffic, and dropping malicious or exploitative payloads [8]. While passive vulnerability assessment supports historical vulnerability assessments on vulnerabilities throughout a services lifetime [7], active vulnerability assessment only provide a snapshot in time of known vulnerabilities [2]. An exemplary scanner of active vulnerability assessment tools is the Nessus Network Security Scanner that lists the various vulnerabilities present in the remote host.

In a recent research, Watson et al. [9] proposed a visualization for vulnerability scan data by network zone using free and open-source tools. The proposed visualization uses a mean of all Nexpose severity scores for a given node to determine its overall severity score. However, this visualization technique is limited since it is attempting to capture large data in a simplified, visual representation. Scalability is another issue with a one to one mapping between devices and nodes for large networks with thousands of devices.

Taking a data driven approach, Shiravi et al. classify the recent works of network security visualization into five use-case classes: host/server monitoring [10], [11], internal/external monitoring [12], [13], port activity [14], [15], attack patterns [16], [17], and routing behavior [18], [19]. Various standard visualization techniques have been extensionally adopted to amplify cognitive activities, such as node-link diagrams [20], [21], Scatterplots [15], [22], [23], and parallel coordinates [13], [24]. CVE details [25] also provides a list of basic statistical charts looking into important features of published CVE, such as popular vendors, products, vulnerability types, and severity over time. Similar visualizations can be found on the National Vulnerability Database webstie[1]. In contrast to these techniques which inspect one feature (dimension) at a time, our proposed system reveals the dynamic correlations among these features.

Shiravi et al. [26] have also identified that most three dimensional systems [27]–[29] are harder for an security analyst to perceive and interact with (compared to conventional 2D systems) due to occlusions which requires a substantial amount of interactions (such as rotating and zooming) from an already overworked security analyst. Consequently, our proposed system focuses on 2D standard and modified visualization techniques to tackle the design requirements for analyzing a large number of Common Vulnerabilities and Exposures entries described in the next section.

[1]https://nvd.nist.gov/general/visualizations/

III. Design motivations and Decisions

In this section, we present the design considerations of multidimensional *CVExplorer* Visualization for Common Vulnerabilities and Exposures (CVE). We first start with some background knowledge on the Common Vulnerability Scoring System (CVSS) defined by the NVD[2].

A. Vulnerability Metrics

CVSS is a free and open industry standard for assessing the severity of computer system security vulnerabilities. CVSS attempts to assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to threat. NVD provides CVSS scores for almost all known vulnerabilities. The NVD supports both CVSS v2.0 and v3.0 standards which contains *base scores* (which represent the innate characteristics of each vulnerability), *temporal scores* (metrics that change over time due to events external to the vulnerability), and *environmental scores* (scores customized to reflect the impact of the vulnerability on an organization). We focus on the latest CVSS (v3.0) and use color-encodings in Table I consistently in this paper.

d11

Severity	None	Low	Medium	High	Critical
Base Score	0.0	0.1-3.9	4.0-6.9	7.0-8.9	9.0-10.0

Table I
NVD Vulnerability Severity Ratings[3].

Common Vulnerabilities and Exposures (CVE) is a catalog of known security threats. The catalog is sponsored by the United States Department of Homeland Security (DHS), and threats are divided into two categories: vulnerabilities and exposures[4]. The CVE entries available in NVD include a variety of fields [30] such as a the vulnerability scores (described above), the vulnerability types (such as *CWE-400: Uncontrolled Resource Consumption* or *CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer*), a list of vulnerable vendors (*Microsoft*, *Oracle*) and products (*Windows*, *apple_tv*, or *android*), external references to advisories, CVE published date, CVE last modified date, and entry descriptions. These are the important variables for our *CVExplorer* visualization.

[2]https://nvd.nist.gov/vuln-metrics/cvss
[4]https://cve.mitre.org/

B. Design motivations

Shodan has been acknowledged as one of the first search engines designed to crawl the Internet and to index discovered services and to provide advanced vulnerability assessment capabilities [7]. Figure 1 shows an example of Shodan output for a given host. This output can be further reconstructed into Common Platform Enumeration (CPE) names and extracted an extensive list of CVEs from NVD. The bottom of Figure 1 shows only a summary portion of the obtained vulnerability information on Port 80 (while there is more information available on other ports as well). To make the matter worse, users can click on "View Raw Data" (at the red arrow) to view a long list of relevant CVEs and their various fields.

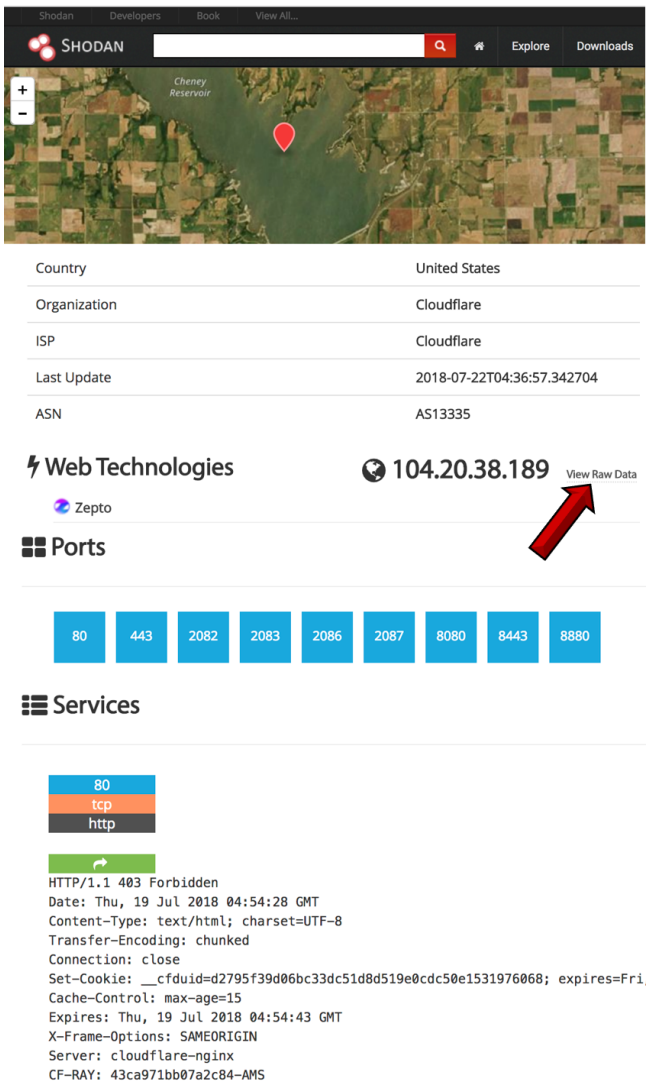

Figure 1. Shodan output for 104.20.38.189: Only summary information on Port 80 (*http*) is displayed at the bottom.

To free a system administrator or cyber analyst from the tediously long vulnerability outputs, we propose *CVExplorer* visualization for analyzing a set of obtained CVEs. While other visualization approaches [9], [25] for analyzing individual CVE feature at a time are publicly available, our *CVExplorer* visualization inspect the dynamic correlations between these dimensions to answer the following research questions:

- **R1:** Are there any relations within and between vendors, products, and vulnerability types at a given time point/interval?
- **R2:** For a given vendor, what are the targeted products/software and what are their levels of vulnerability change over time?
- **R3:** What are the popular vulnerability types and how did they evolve?
- **R4:** What are the popular topics associated to different level of vulnerability severity over time?

o answer these research questions, our prototype supports a full range of analysis tasks [31], [32]. Notice that our proposed visualization is not limited by the four above research questions but rather to find the correlations/associations of important features within the CVE data:

- **T1**: Provide a summary view of prominent fields in the CVE data [33]. Our prototype provides a quick overview of multiple dimensions in interactive parallel coordinates and evolution of important topics in stream graphs (Section IV-A and Section IV-C).
- **T2**: Retrieve and display details on demand. Users can select a topic of interest (such as a vendor name or product) to start the exploratory data analysis. Multiple views are updated accordingly (Section V).
- **T3**: Filter vulnerability types or severity level on user request. For example, users may want to see the evolution of critical vendor or product overtime (Section V).
- **T4**: Sort topics by vulnerability severity so that users can focus on more critical ones (Section IV-C).

Based on these research challenges and visualization requirements, we come up with a multiple view interactive system which contains popular visual components (such as parallel coordinates and force-directed layouts) and customize graphs (the *Timed Wordle*). In the next section, we discuss our design considerations and decisions for the *CVExplorer* visualization.

C. Design Decisions

Parallel coordinates are a standard way of visualizing high-dimensional geometry and analyzing multivariate data [34], [35]. Therefore parallel coordinates are adapted to display the correlations of prominent dimensions within the data (visualization task **T1**). As force-directed layout uses repulsive forces between nodes and attractive forces between adjacent nodes and therefore it is handy to highlight network structures (such as clusters or outliers). We use force-directed layouts as

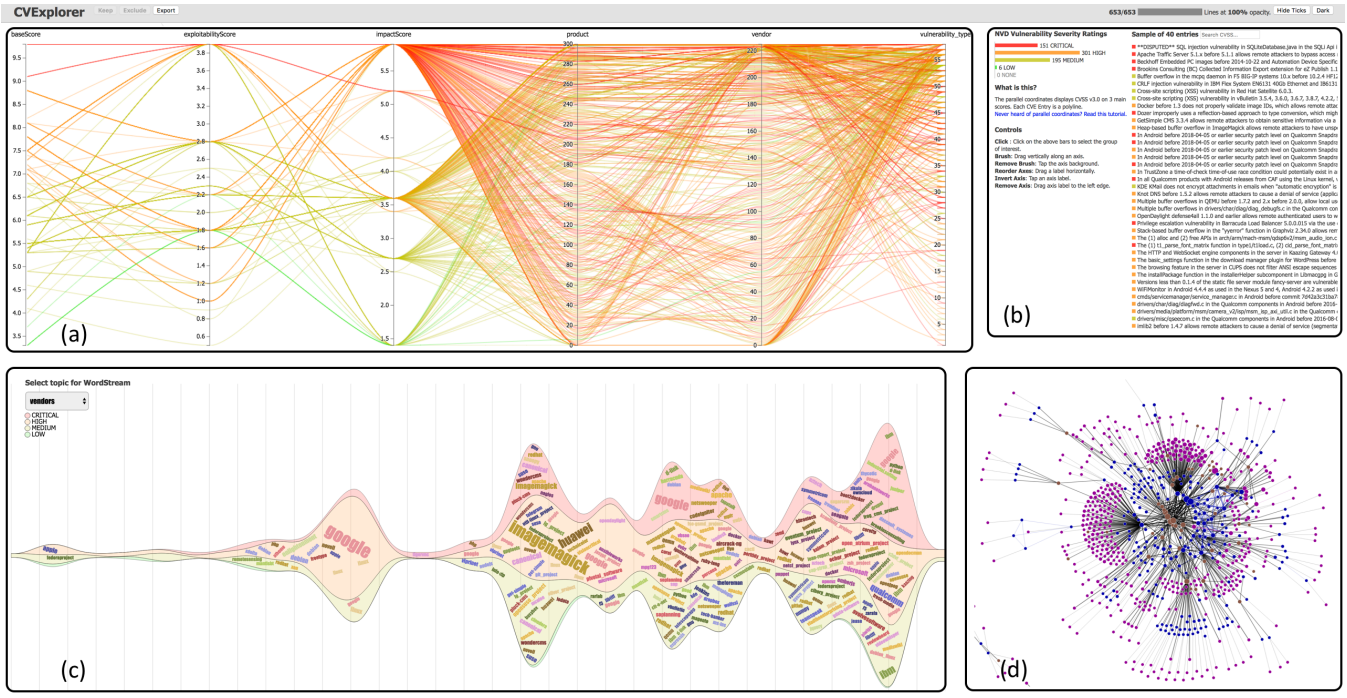

Figure 2. *CVExplorer* visualization for 653 CVEs: (a) Parallel coordinates of important features in CVEs (b) Summary of NVD Vulnerability Severity Ratings and sample CVEs (c) *Timed Wordle* of popular vulnerable vendor over time (d) Network of vendors (in blue), products (in purple), and vulnerability types (in brown).

the primary way to group related entities (vendors, products, and vulnerability types) and minimize link crossings which are the main limitation of parallel coordinates.

While very effective for visualizing network structures, the lack of temporal information is the main drawback of the force-directed graph. Therefore, we propose a hybrid visualization of *Streamgraph* [36] and *Wordle* to maximize the space usage for displaying the evolution of important topics (vendors, products, and vulnerability types) and hence communicate global criticality trends [37], [38]. Within this graph, the time axis is aligned horizontally from left to right. This design is widely adopted when visualizing time series data [39], [40]. The main drawback of this hybrid visualization: It is tricky to follow the progression of individual entities. In other words, visually identifying different instances of the same label is challenging due to changes in text orientation produced by the *Wordle* algorithm. We alleviate this problem via user interaction: a local stream of the interested entity will be highlighted on demand (visualization task **T2**).

IV. *CVExplorer* Visualization

Figure 2 provides a schematic overview of *CVExplorer*. This section discusses these main components in detail.

A. Parallel Coordinates

As discussed above, we adopted parallel coordinates [41] to present the relationships of the following prominent dimensions in the CVE data: vulnerability scores (basic, temporal, and environmental scores), vendors, products, and vulnerability types. The last three dimensions/coordinates are ordered by how popular they are in the input data. Filters on each dimension can be applied by simply dragging on the axes. Figure 3 shows an example of filtering 473 *critical* CVEs out of 3,291 CVEs in 2018 (downloaded from NVD in July 2018). We can notice that *critical* CVEs are mostly reported on certain types of vulnerability types (at the top portion of the right axis).

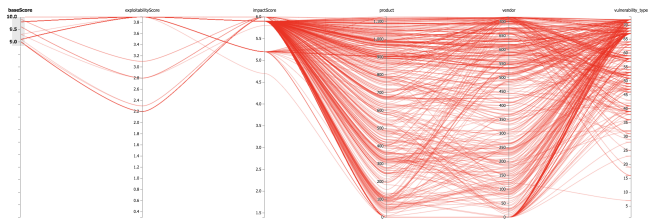

Figure 3. Parallel Coordinates of 473 CVEs in *critical* category published on NVD from January to July 2018.

CVExplorer also supports other interactions such as reordering axes by *drag and drop*, dropping a dimension, or revert its scale. The right panel in Figure 2(b) also supports group or individual CVE selection (meeting the visualization task **T3**).

B. Network

The force-directed layout is an efficient tool to reduce edge-crossings (a drawback of parallel coordinates) as the related entities can freely move closer to each other to form clusters. In this context, the entities (vendors, products, and problem types) are considered as related if they appear in the same CVE and therefore connected by a link. The link thickness indicates how often they are reported in the same CVEs. Node sizes are calculated based on the number of reported CVEs. We use a different color scheme to encode network nodes to differentiate them from the vulnerability severity color range: blue for vendors, purple for products, and brown for vulnerability types.

Figure 4 shows an example of vendors and products networks extracted from 4,541 CVEs published on NVD from January to July 2018 (answering the research question **R1**). We can easily notice the interconnections among the four most popular vendors at the netwrok center: *Redhat*, *Canonical*, *Debian*, and *Oracle*. We can also notice a strong link between *Debian* and *Mozilla* which has only one vulnerable product (*Firefox*). One might argue that the force-directed layout suffers from the "hairball" issue and that an interactive parallel coordinates plot (in Section IV-A) or alternatives might have been a cleaner way to illustrate these relationships. However, we did not implement the force-directed graph to visualize millions of entities. Rather, it aims to present a focus relational view of popular vendors, products, and problem types. Therefore, a filter (slider) is provided to allow users quickly narrow down strong correlations among important entities. We demonstrate the network filtering operations in Section V and our demo video at https://idatavisualizationlab.github.io/CVSS/.

Figure 5 shows vulnerability types network of the same data. We can easily detect a few pairs of strongly connected vulnerability types: *CWE-119* (Improper Restriction of Operations within the Bounds of a Memory Buffer) vs. *CWE-200* (Information Exposure), *CWE-264* (Permissions, Privileges, and Access Controls) vs. *CWE-362* (Concurrent Execution using Shared Resource with Improper Synchronization), and an isolated pair of *CWE-918* (Server-Side Request Forgery) vs. *CWE-611* (Improper Restriction of XML External Entity Reference). Moreover, *CWE-200* and *CWE-264* at the center of this network have the highest degrees of centrality. We will investigate the vulnerability types w.r.t vendors and their products in the use cases. When clicking on a node, all related vendors, products, and problem types are highlighted. Moreover, all CVEs containing these relationships (co-occurrences) can be inspected in a separate *json viewer*.

C. Timed Wordle

The *Timed Wordle* is implemented using the combined *Wordle* [42] and *Streamgraph* [36], [43] algorithms. *Wordle* main strength is the ability to give quick emphasis on important terms using relatively larger font sizes. It is also

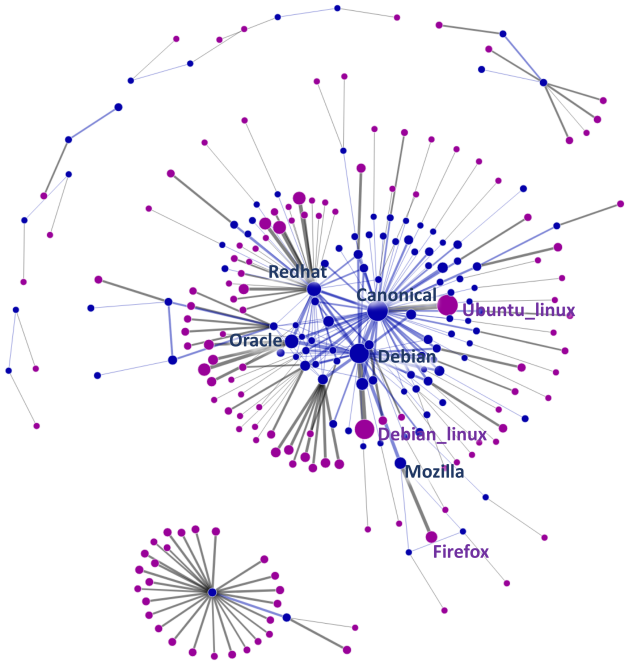

Figure 4. Network of popular vendors and products of 4,541 CVEs published on NVD from January to July 2018: blue for vendors and purple for products. Popular vendors (*Redhat*, *Canonical*, *Debian*, and *Oracle*) are interconnected (at the network center) since they share the same Common Vulnerabilities and Exposures.

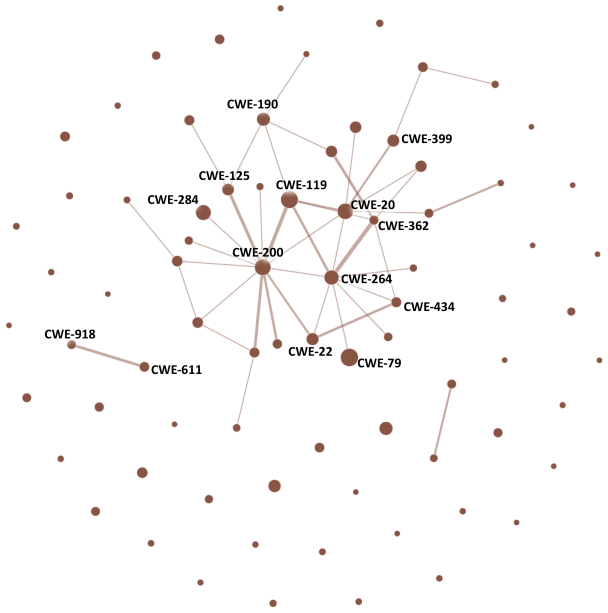

Figure 5. Network of popular vulnerability types of 4,541 CVEs published on NVD from January to July 2018: brown for vulnerability types (or problem types).

known for its algorithm to optimize its space by organizing words efficiently. However, *Wordle* lacks the ability to show the evolution of the topic over time. On the other hand, *Streamgraph* is a popular method for visualizing topic evolution. Its strength is the ability to provide a comprehensive overview of the evolution of the underlying topics over time. However, *Streamgraph* is limited in terms of the number of layers and space for each layer. Hence, its layer normally does not contain or contains only a few terms of the topic that the layer is representing, making it difficult to trace and compare the evolutions of different terms in the topic across time. Our implemented *Timed Wordle* presents a strategy to visualize the evolution of topics (i.e., qualitative severity rankings of the CVEs) and their terms (i.e., vendors, problem types, products, and descriptions) by using *Streamgraph* to represent the topic overview across time and using *Wordle* to compactly and elegantly place terms to its corresponding layer and time step and to visually emphasize important terms with relatively larger font-sizes (answering the research question **R4**). As depicted in Figure 6), we also set lower opacity to unimportant topics to reduce information density in the combined visualizations. Severity layers are ordered and color-coded according to Table I (meeting the visualization task **T4**) while keyword colors are color-coded as in the network view: blue for vendors, purple for products, and brown for vulnerability types (or *CWEs*).

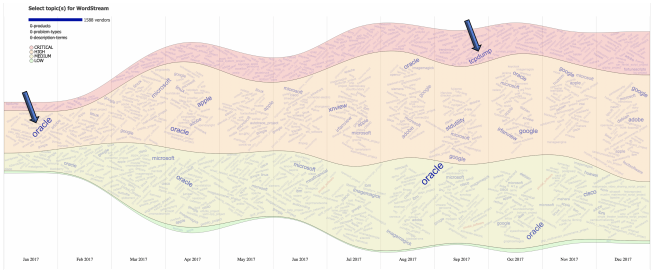

Figure 6. *Timed Wordle* visualization of popular vendors in 14,746 CVEs reported to NVD in 2017: Vertical layers in the graph are ordered and color-coded by their severity classifications. Important vendors are displayed in larger font-sizes and opacity for tracking purpose.

In addition, interactive features were also added to assist the exploratory data analysis further. For instance, clicking on the term *oracle* at Figure 6, its corresponding stream layer is shown as in Figure 8(b) to show its overall contribution to the underlying topic layer. By clicking on the a term again, all the related CVEs are shown in a *json viewer* in order to further investigate the details of the associated CVEs (visualization task **T2**) such as the affected vendors, affected products, the problem types, CVE descriptions, and also the list of reference links.

V. Use cases

A. CVE Data

The NVD[5] is the vulnerability management database managed by U.S government in order to support systematic and automatic reporting and managing of vulnerabilities. As of the time of this writing, NVD contains 110,766 CVEs with different qualitative severity rankings as *low* (6,181 CVEs), *medium* (52,218 CVEs), *high* (47,182 CVEs), and *critical* (5,185 CVEs). Figure 7 shows the overview of the severity distribution of these 110,766 CVEs over time. We can easily notice the emergence of *critical* type in 2014 and the gain of its popularity as *critical* rating has been introduced recently together with CVSS v3.0.

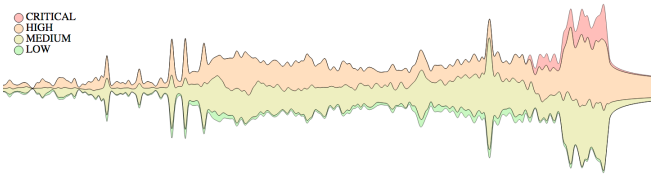

Figure 7. Overview of 110,766 CVEs reported from 1998 to 2018 obtained from NVD. Layers are severity classification: *low*, *medium*, *high*, and *critical*.

This section demonstrates our approach on two sample CVE subsets to show how this visualization solution could quickly highlight the overview of security problems with different qualitative severity rankings over time and also assists in-depth investigation if needed. The first one is to analyze security alerts in 2017 from the NVD Data Feeds[6]. The second case is the analysis of the security issues of an ASN (Autonomous System Number) with CVEs dataset collected from *Shohan* [3], [7]. While the first use-case communicates global criticality trends, the second demonstrates *CVExplorer* application to a specific ASN.

B. Use case 1: CVE-2017

The *CVE-2017* dataset collected from NVD CVE feeds contains 14,746 CVEs with qualitative severity rankings distribution: *low* (205 CVEs), *medium* (5,119 CVEs), *high* (7,509 CVEs), and *critical* (1,913 CVEs). This use-case shows how our interactive visual analytics system could help to quickly identify critical security alerts of their affected vendors, products, and problem types. In addition, users could also investigate the related references to further examine about these CVEs as well as finding patch updates for these security alerts.

We focus on analyzing the CVEs at *critical* and *high* qualitative rankings by filtering on the severity axis of parallel coordinates. Figure 8 displays several interesting views of the filtered data. At a glance to the *Timed Wordle* in the top panel (Figure 8(a)), we can easily identify *CWE-284*

[5]https://nvd.nist.gov
[6]https://nvd.nist.gov/vuln/data-feeds

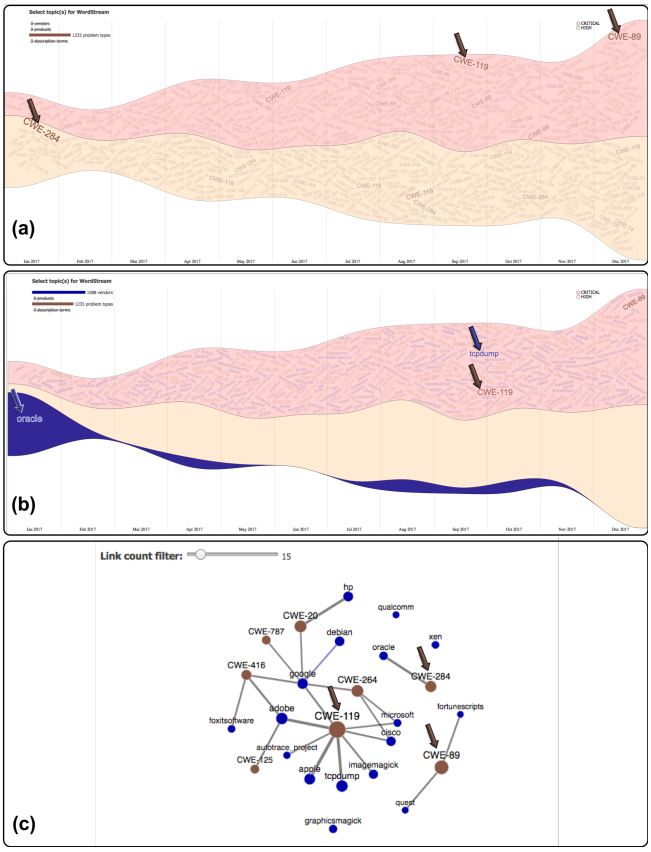

Figure 8. *CVExplorer* visualization for the NVD CVE-2017 dataset for *high* and *critical* qualitative severity rankings: (a) *CWE-284*, *CWE-119*, and *CWE-89* as the dominant security problem types in 2017; (b) *tcpdump* and *CWE-119* are the dominant vendor and problem type in September 2017 and Oracle has *seasonal* reporting schedule; (c) The network view shows the relationships among the main problem types and vendors.

(Vulnerability in the Oracle Service Fulfillment Manager component of Oracle E-Business Suite[7]), *CWE-119* (improper restriction of operations within the bounds of a memory buffer[8]), and *CWE-89* (Movable Type plugin A-Member and A-Reserve vulnerable to SQL injection[9]) as the dominant problems in 2017. The middle panel shows our investigation of an affected company (*Oracle*). It reveals the *seasonal* vulnerability reporting pattern from *Oracle*. After further reviewing the related references in the CVE *json viewer*, we found that these were *Oracle* scheduled patch updates. In particular, Oracle's critical patch updates schedules were on the Tuesday closest to the 17th day of January, April, July, and October [10].

Figure 8(c) displays the high correlations of among popular vendors and problem types by filtering via the provided slider: vendors and problem types are collocated in at least 15 *high* and *critical* CVEs. As depicted, *CWE-119* vulnerability is highly connected to *Google*, *Adobe*, *Apple*, *Microsoft*, and *Cisco* among the other vendors. In particular by inspecting 96 *critical* CVEs reported in September 2017 in the *json* viewer, we can confirm that they all belong to the *CWE-119* vulnerability type and mostly affect Apple's products.

C. Use case 2: CVEs from an ASN

The first use case shows high-level vulnerability trends (which is more suitable for someone in charge of producing executive reporting for an organization), this section provides another use case for system administrator or cyber analyst to investigate the security of an ASN over time. One sample ASN was chosen, and five pages of CVEs (100 *banners* per page) were collected from *Shodan* [3] for the ASN and 441 vulnerabilities found. Of which 439 were CVEs and the other two were of type MS17-010 (Microsoft Security Bulletin Number[11]). The CVE qualitative severity ranking distribution was *high* (157 CVEs), *critical* (56 CVEs), *medium* (187 CVEs), and *low* (39 CVEs).

The parallel coordinates Figure 9(a) show the correlations among the prominent dimensions within the data. Notice that *critical* (red) CVEs have lower *Exploitability* metrics which reflect the ease and technical means by which the vulnerability can be exploited. The *Timed Wordle* in the middle panel highlights dominant problem type, vendor, and product in 2010: *CWE-119*, *microsoft*, and *iis* versus the recent correspondings: *CWE-20* (`mod_auth_digest` does not properly initialize or reset the value placeholder in authorization headers leading to information disclosure or denial of service[12]), *apache*, and *http_server*. The network view in Figure 9(a) further confirms these correlations.

This use case shows that it is relatively easy to use our visualization system to identify critical security alerts and related references for patch updates. As of July 2018, we also selected several IPs from this ASN and scanned them with Shodan and discovered that many of these critical CVEs are still existing in the services from the specified ASN and this "security-sensitive" corporation seemed ignorant about these critical vulnerabilities.

D. Implementation

CVExplorer is implemented in D3.js [44]. The online application, source code, supplementary materials, more use cases, and a demo video are provided via our GitHub project repository, at https://idatavisualizationlab.github.io/CVSS/.

E. Discussions and limitations

This section discusses the scalability of the proposed framework for big data: Can the prototype handle more than 110,766 CVEs (as of Angust 2018)?

[7] http://www.oracle.com/technetwork/security-advisory/cpujan2017-2881727.html

[8] https://cwe.mitre.org/data/definitions/119.html

[9] https://jvn.jp/en/jp/JVN78501037/index.html

[10] https://oracle.com/technetwork/topics/security/alerts-086861.html

[11] https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010

[12] https://www.debian.org/security/2017/dsa-3913

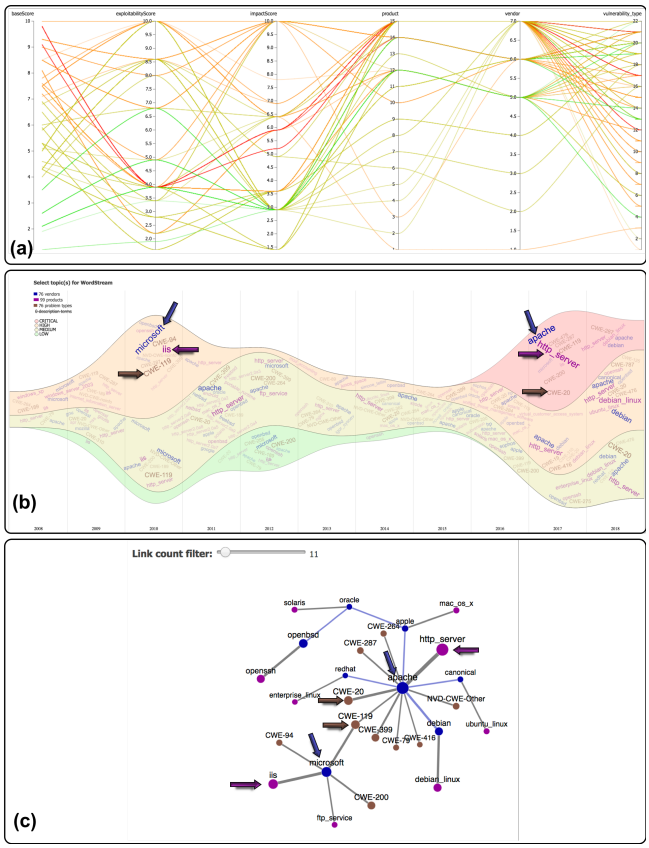

Figure 9. Visualizing features of CVEs from an ASN: (a) Parallel coordinates of prominent dimensions, (b) *Timed Wordle* highlights the recent critical problem type, vendor, and product (at the right arrows) versus those in 2010, (c) Network view confirms the stronger connections among problem types, vendors, and products in (b) via thicker links.

The force-directed graph has known limitations ("hairball" issues), and one might argue that this representation is not the most effective way to show all relationships between vendors, products, and vulnerabilities. Nevertheless, it would be hard to argue that the networks in Figure 8(c) and Figure 9(c) have successfully captured the correlations of a focused set of entities (using the slider to filter the strength of these connections). While the parallel coordinates provide an overview of correlations between sequential dimensions, the force-directed graph displays a focused view: the relationships of popular vendors, products, and security types (answering the research question **R1**).

Conventional parallel coordinates [45], [46] does not scale for big data (visual clutter and overplotting issues) since each CVE profile is rendered as a polyline. We tackle the scalability issue of parallel coordinates by adopting the following approaches:

- Asynchronous rendering method using *d3.timer* [44] which is an efficient queue for managing a large number of concurrent renders for the polylines. Please consult the demo video, published on our github repository for this effect.
- Edge-bundling method using density-based clustering for each dimension: this approach allows rendering the clustered lines using polygons, decreasing rendering time remarkably [47], [48]. Figure 10 shows an example of such approach.

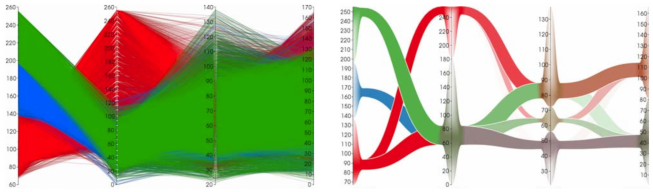

Figure 10. An abstracted parallel coordinates (right) to visualize correlations of big data (left). This figure is adapted from [47].

Timed Wordle is a new combination of visualization. As both *Streamgraph* and *Wordle* represent high-level aggregated data, *Timed Wordle* scales well with the large number of CVEs. While this hybrid visualization maximizes space usage and communicates global trends, other performance measures are still in questions: (1) Whether limitations of *Streamgraph* and *Wordle* have been mitigated and they mutually become more useful, (2) What if the labels were replaced with coloured dots? This would remove text and orientation as discussed in the *Design Decisions* Section and focus our pre-attentive perception capacities on color and size, and (3) Could each level of criticality have its own stream graph, thus allowing the use of streams for each vendors, products, and vulnerabilities? Considering these variables, formal qualitative user studies would be instrumental for deciding how to best configure the new combination of visualizations, leading to design iteration and future improvements.

VI. Conclusion and Future Work

The vulnerability of our cyber systems constitutes a critical threat to national security. This paper proposes an interactive visual analytics system for analyzing vulnerability reports from the NVD that can help enhance the protection against human-error-utilizing cyber attacks. The system has three linked components: Parallel coordinates, forced directed network, and *Timed Wordle*. While parallel coordinates are a standard technique for visualizing high-dimensional data using polylines, the force-directed layouts provide a way to highlight related entities by positioning them near to each other. Network entities are brought closer to each other (forming clusters) by forces applied on nodes and connections between nodes. Finally, the *Timed Wordle* provides a supplement view on the evolution of vendors, products, as well as types and levels of vulnerability. This research aims to reduce the cognitive workload of the cybersecurity analyst and move the user interaction into the visual space. However, this needs further studies to characterize the user's

background knowledge, reasoning process, and cognitive style based on their interactions in order to recommend suitable visual representations of potential cyber threats.

Our future work is therefore to develop a network vulnerability analysis, remedy, and alerting tool that can help enhance the defense against human-error-utilizing cyber attacks by (1) automatically interpreting and analyzing scanner output jargons, using artificial intelligent and machine learning techniques, into user-friendly graphs and animations to remind network node users of potential risks, where the user-friendliness of the graphs or animations will be designed and evaluated using human factor psychological expertise; (2) recommending users of certain actions to avoid potential risks, detecting risks and sending high severity risks not taken care of by users to network system administrators; and (3) experimentally testing, evaluating, and fine-tuning the tool on a network of computers with many users and various activities.

REFERENCES

[1] M. Evans, L. A. Maglaras, Y. He, and H. Janicke, "Human behaviour as an aspect of cyber security assurance," *CoRR*, vol. abs/1601.03921, 2016. [Online]. Available: http://arxiv.org/abs/1601.03921

[2] L. Harrison, R. Spahn, M. Iannacone, E. Downing, and J. R. Goodall, "Nv: Nessus vulnerability visualization for the web," in *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, ser. VizSec '12. New York, NY, USA: ACM, 2012, pp. 25–32. [Online]. Available: http://doi.acm.org/10.1145/2379690.2379694

[3] J. Matterly, "Shodan: The world's first search engine for internet-connected devices," 2014, http://www.shodanhq.com[Accessed date: June 5, 2018].

[4] S. Samtani, S. Yu, H. Zhu, M. Patton, and H. Chen, "Identifying scada vulnerabilities using passive and active vulnerability assessment techniques," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, Sept 2016, pp. 25–30.

[5] M. Zalewski, "p0f v3: Passive fingerprinter," 2012, http://lcamtuf.coredump.cx/p0f3/[Accessed date: July, 2018].

[6] E. Fjellskal, "Passive real-time asset detection system," 2009, http://gamelinux.github.io/prads/[Accessed date: July, 2018].

[7] B. Genge and C. Enăchescu, "Shovat: Shodan-based vulnerability assessment tool for internet-facing services," *Sec. and Commun. Netw.*, vol. 9, no. 15, pp. 2696–2714, Oct. 2016. [Online]. Available: https://doi.org/10.1002/sec.1262

[8] D. Kennedy, J. O'Gorman, D. Kearns, and M. Aharoni, *Metasploit: The Penetration Tester's Guide*, ser. No Starch Press Series. No Starch Press, 2011. [Online]. Available: https://books.google.com/books?id=TWKLBAAAQBAJ

[9] S. Watson and H. R. Lipford, "A proposed visualization for vulnerability scan data," in *SOUPS*, 2017.

[10] R. F. Erbacher, K. L. Walker, and D. A. Frincke, "Intrusion and misuse detection in large-scale systems," *IEEE Comput. Graph. Appl.*, vol. 22, no. 1, pp. 38–47, Jan. 2002. [Online]. Available: https://doi.org/10.1109/38.974517

[11] R. F. Erbacher, "Intrusion behavior detection through visualization," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 3, Oct 2003, pp. 2507–2513 vol.3.

[12] R. Ball, G. A. Fink, and C. North, "Home-centric visualization of network traffic for security administration," in *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ser. VizSEC/DMSEC '04. New York, NY, USA: ACM, 2004, pp. 55–64. [Online]. Available: http://doi.acm.org/10.1145/1029208.1029217

[13] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju, "Visflowconnect: Netflow visualizations of link relationships for security situational awareness," in *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ser. VizSEC/DMSEC '04. New York, NY, USA: ACM, 2004, pp. 26–34. [Online]. Available: http://doi.acm.org/10.1145/1029208.1029214

[14] K. Abdullah, C. Lee, G. Conti, and J. A. Copeland, "Visualizing network data for intrusion detection," in *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, June 2005, pp. 100–108.

[15] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen, "Portvis: A tool for port-based detection of security events," in *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ser. VizSEC/DMSEC '04. New York, NY, USA: ACM, 2004, pp. 73–81. [Online]. Available: http://doi.acm.org/10.1145/1029208.1029220

[16] K. Abdullah, C. Lee, G. Conti, J. A. Copeland, and J. Stasko, "Ids rainstorm: Visualizing ids alarms," in *Proceedings of the IEEE Workshops on Visualization for Computer Security*, ser. VIZSEC '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 1–. [Online]. Available: https://doi.org/10.1109/VIZSEC.2005.8

[17] Y. Livnat, J. Agutter, S. Moon, R. F. Erbacher, and S. Foresti, "A visualization paradigm for network intrusion detection," in *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, June 2005, pp. 92–99.

[18] T. Wong, V. Jacobson, and C. Alaettinoglu, "Internet routing anomaly detection and visualization," in *2005 International Conference on Dependable Systems and Networks (DSN'05)*, June 2005, pp. 172–181.

[19] M. Lad, D. Massey, and L. Zhang, "Visualizing internet routing changes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1450–1460, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2006.108

[20] G. A. Fink, P. Muessig, and C. North, "Visual correlation of host processes and network traffic," in *Proceedings of the IEEE Workshops on Visualization for Computer Security*, ser. VIZSEC '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 2–. [Online]. Available: https://doi.org/10.1109/VIZSEC.2005.18

[21] L. Colitti, G. D. Battista, F. Mariani, M. Patrignani, and M. Pizzonia, "Visualizing interdomain routing with bgplay," *J. Graph Algorithms Appl.*, vol. 9, pp. 117–148, 2005.

[22] K. Lakkaraju, R. Bearavolu, A. Slagell, W. Yurcik, and S. North, "Closing-the-loop in nvisionip: integrating discovery and search in security visualizations," in *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).*, Oct 2005, pp. 75–82.

[23] C. P. Lee, J. Trost, N. Gibbs, R. Beyah, and J. A. Copeland, "Visual firewall: real-time network security monitor," in *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).*, Oct 2005, pp. 129–136.

[24] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen, "Real-time and forensic network data analysis using animated and coordinated visualization," in *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, June 2005, pp. 42–49.

[25] S. Özkan, "Cve details: The ultimate security vulnerability datasource," 2011, https://www.cvedetails.com/index.php[Accessed date: July 10, 2018].

[26] H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A survey of visualization systems for network security," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 8, pp. 1313–1329, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.144

[27] T. Takada and H. Koike, "Tudumi: information visualization system for monitoring and auditing computer logs," pp. 570–576, 02 2002.

[28] A. Yelizarov and D. Gamayunov, "Visualization of complex attacks and state of attacked network," in *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*. IEEE, 2009, pp. 1–9.

[29] A. Komlodi, P. Rheingans, U. Ayachit, J. R. Goodall, and A. Joshi, "A user-centered look at glyph-based security visualization," in *Visualization for Computer Security, 2005.(VizSEC 05). IEEE Workshop on*. IEEE, 2005, pp. 21–28.

[30] B. A. Cheikes, D. Waltermire, and K. Scarfone, "Common platform enumeration: Naming specification version 2.3," *NIST Interagency Report 7695, NIST-IR*, vol. 7695, 2011.

[31] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *Proc. of the IEEE Symposium on Information Visualization*, 2005, pp. 15–24.

[32] N. Andrienko, G. Andrienko, and P. Gatalsky, "Exploratory spatio-temporal visualization: an analytical review," *Journal of Visual Languages & Computing*, vol. 14, no. 6, pp. 503–541, 2003.

[33] D. A. Keim, C. Panse, and M. Sips, "Information visualization: Scope, techniques and opportunities for geovisualization," in *Exploring Geovisualization*, J. Dykes, Ed. Oxford: Elsevier, 2004, pp. 1–17.

[34] X. Zhao and A. Kaufman, "Structure revealing techniques based on parallel coordinates plot," *Vis. Comput.*, vol. 28, no. 6-8, pp. 541–551, Jun. 2012. [Online]. Available: http://dx.doi.org/10.1007/s00371-012-0713-0

[35] A. Dasgupta and R. Kosara, "Pargnostics: Screen-space metrics for parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1017–2626, 2010.

[36] L. Byron and M. Wattenberg, "Stacked graphs – Geometry & aesthetics," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1245–1252, 2008.

[37] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong, "TextFlow: Towards better understanding of evolving topics in text," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2412–2421, 2011.

[38] P. Xu, Y. Wu, E. Wei, T.-Q. Peng, S. Liu, J. Zhu, and H. Qu, "Visual analysis of topic competition on social media," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2012–2021, 2013.

[39] M. Dork, D. Gruen, C. Williamson, and S. Carpendale, "A visual backchannel for large-scale events," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1129–1138, 2010.

[40] T. N. Dang, A. Anand, and L. Wilkinson, "TimeSeer: Scagnostics for high-dimensional time series," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 3, pp. 470–483, March 2013.

[41] J. Heinrich and D. Weiskopf, "State of the art of parallel coordinates." in *Eurographics (STARs)*, 2013, pp. 95–116.

[42] J. Feinberg, "Wordle," *Beautiful Visualization: Looking at data through the eyes of experts.*, pp. 37–58, 2010.

[43] M. Wattenberg, "Baby names, visualization, and social data analysis," in *Proc. IEEE Symp. on Information Visualization*, 2005, pp. 1–7.

[44] M. Bostock, V. Ogievetsky, and J. Heer, "D3 data-driven documents," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, 2011.

[45] A. Inselberg, "The plane with parallel coordinates," *The visual computer*, vol. 1, no. 2, pp. 69–91, 1985.

[46] J. Zhang, M. L. Huang, W. B. Wang, L. F. Lu, and Z. Meng, "Big data density analytics using parallel coordinate visualization," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, Dec 2014, pp. 1115–1120.

[47] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf, "An edge-bundling layout for interactive parallel coordinates," in *2014 IEEE Pacific Visualization Symposium*, March 2014, pp. 57–64.

[48] K. T. McDonnell and K. Mueller, "Illustrative parallel coordinates," in *Proceedings of the 10th Joint Eurographics / IEEE - VGTC Conference on Visualization*, ser. EuroVis'08, 2008, pp. 1031–1038. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2008.01239.x