2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) | 978-1-6654-2463-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/COMPSAC51774.2021.00057

TimeRadar: Visualizing the Dynamics of Multivariate Communities via Timeline Views

1st Ngan V.T. Nguyen
Computer Science department
Texas Tech University
Lubbock, TX, USA
Ngan.V.T.nguyen@ttu.edu or 0000-0001-5039-8019

2nd Jon hass
Office of the CTIO
Dell Technologies
Texas, USA
Jon.Hass@dell.com

3rd Tommy Dang
Computer Science department
Texas Tech University
Lubbock, TX, USA
Tommy.Dang@ttu.edu or 0000-0001-8322-0014

***Abstract*—Analyzing temporal event sequences play an important role in many application domains, such as workload behavior analysis, hardware fault diagnosis, and natural disaster resilience. As data volume keeps growing, real-world temporal event sequences are often noisy, missing, and complex, thus making it a daunting task to convey much of the information from a comprehensive overview for analysts. This work proposes a visual approach based on clustering and superimposing to construct a high-level overview of sequential event data while balancing the amount of information and its cardinality. We also implement an interactive prototype, called *TimeRadar*, that allows domain analysts to simultaneously analyze sequence clustering, extract distribution patterns, drill multiple levels of detail to accelerate the analysis. This work aims to provide an abstracted view of temporal event sequences where significant events are highlighted. The *TimeRadar* is demonstrated through case studies with real-world temporal datasets of various sizes.**

I. Introduction

A variety of visualization techniques and methods have been proposed in the literature in response to the needs of temporal event sequence analysis from the storytelling of personal events [1], events flow [2], multiple sequences view [3], comparing variants [4] to reducing volume and variety of data [5], [6]. The data mining technique is also adapted to facilitate sequential patterns from large and high dimensional data [7], [8]. Although much effort has been made to accelerate temporal event sequential data analysis, the challenge to create a simple, intuitive, comprehensive overviews and easy to interpret visual layout still remains. For example, the Sankey diagram [2] was not able to handle a large volume of noisy data with high dimensions due to the vertical alignments and space. Sequential Pattern Mining algorithms [7], [8] often provide excessive redundant patterns, which in turn require analysts more time to find patterns of interest. Patterns based on clustering techniques may produce clusters that are difficult to convey information [9], [10]. In this paper, we propose a visual approach based on two strategies: *clustering* for grouping similar multivariate statuses into major groups of interests using popular clustering methods, such as k-means [11]. *superimposing* overlays multivariate representations on top of the clustering bundles, which allow users to explore high-dimensional data in a given period and individual sequences. Our technique, called *TimeRadar*, visually summarizes the original temporal event sequences with clustering while recovering individual sequences from the stacked radar chart. The challenge is to identify a set of clusters for a meaningful visual summary without imposing redundant patterns and inducing information loss. To tackle this issue, we first define a small set of operating statuses within the multivariate data and then represent them onto the timeline where repeated statuses are compressed into a color-coded horizontal line. To summarize, the main contributions of this paper include:

- We propose a visual approach for representing large multivariate time series. Our approach consists of multivariate clustering, visual abstracting, and superimposing high-dimensional data points.
- We compare our *TimeRadar* visualization to another timeline visualization technique that has been widely studied and used for various applications. The comparisons are made in both forms of qualitative and quantitative feedback from a user study on 30 participants.
- We provide the use cases on real-world datasets and user feedback from experts and general users to assess the usability and effectiveness of our *TimeRadar* approach.

The rest of the paper is organized as follows: Section **??** summarizes similar work. We present our system design in Section II. A user study on *TimeRadar* is presented in Section III by using the Technology Acceptance Model. We conclude our paper with future work in Section V.

II. *TimeRadar* Design

In this section, we present the motivation and design choices for our proposed *TimeRadar* application that facilitates the identification and exploration of temporal event patterns over a given period of time. We first studied existing work to collect high-level task requirements, as suggested by Plaisant and Shneiderman [12]. Second, we gather requirements from experts in a specific domain: High-performance computing systems. This application domain requires constant monitoring of a large number of computing nodes via various health metrics, such as CPU temperatures, fan speeds, CPU usage, and power consumption. At each time step, a computing node is characterized by these nine dimensions; we call it the operating status of a computing node. We summarized and categorized high-level tasks into the four most common ones

978-1-6654-2463-9/21/$31.00 ©2021 IEEE
DOI 10.1109/COMPSAC51774.2021.00057

and centered our design around these tasks [13]. They are as follows:

- **T1**: Provide an overview of temporal event sequence of a large number of computing nodes in the data center [14].
- **T2**: Capture the major operating statuses and display them in aggregated views which allow users to grasp their meanings at a glance quickly.
- **T3**: Automatically characterize interesting events (e.g., identifying sudden change on the time series and detecting multivariate outliers) and highlight them on the timeline view.
- **T4**: Customize visual layouts (e.g., reordering events, running jobs, and grouping computing nodes with similar timelines) based on user's preferences.

We design *TimeRadar* to support the above tasks while following the general considerations of timeline visualizations [15]. Starting from the left of the high-level overview (**T1**), users can identify unusual events which are signified by the encoded color timeline and the superimposed radar chart (**T2**). Status changes (from one radar group to another) can be visually tracked along the timeline. Aggregated views presented at the end of the timelines allow a user to summarize a subset of events in a given time (**T3**). The radar charts superimposed on top of the timelines enable analysts to identify the behavior pattern of many profiles. Similar profiles can be ranked, ordered vertically, and grouped to focus on the entries with different temporal behaviors (**T4**).

A. *The timeline view*

The timeline window allows an analyst to have a holistic overview of the evolution of multiple event sequences (Visualization task **T1**). Each sequence of events is represented as a horizontal line (from left to right) since tracing lines is easier compared to tracing curves [16], especially when the number of timelines increases.

TimeRadar superimposes color-coded glyphs onto the timeline event (Visualization task **T2**) on the timeline to indicate group changes [17] while unchanged statuses are represented as straight lines. This allows users to focus on important events and highlight the causal relationships (triggering similar group changes) across various entries: A new job is allocated on multiple machines, or a hurricane affects neighboring states.

B. *The control panel*

The control panel allows analysts to customize/adjust the visual space. In particular, options in this panel include:

- Reorder the entries vertically by their groups (such as computers allocated to the same jobs/users) and the similarity of their timelines [18].
- Show/hide a group of entries. For example. Users may want to remove the normal entries (with the metrics within a usual range) or stable entries (never change their status group) to focus on the unusual behaviors.
- Adjust the timeline scales or the number of radars appearing on the timeline by a given threshold.

The radar settings: This panel allows users to customize the order of dimensions which is considered to be significant in perceiving the *morphology* of radar layouts [19]. Radar plots are used for quickly discerning abnormal dimensions [20].

Clustering: We adapt the existing clustering algorithms, such as k-means [21] to summarize the major operating statuses (or clusters) of computing nodes. The parameters of k-means are customizable via the control panel. The k-means method requires an appropriate convergence criterion such as no reassignment of patterns to new cluster centers or minimal decrease in squared error [22]. We avoid the infinity loop by inputting the maximum number of iterations to stop the computation when no further improvements are made.

III. USECASES

This section demonstrates our proposed visualization tool on the two datasets: the High-Performance Computing Center dataset, US employment dataset, and Rest Useful Life dataset.

A. *High Performing Computing Center data*

The High-Performance Computing Center data includes 467 computing nodes with nine metrics: CPU1 vs. CPU2 Temperature, Fan speeds, Memory usage, and Power consumption.

Figure 1 illustrates a use case where analysts can use *TimeRadar* to capture dynamic behaviors of multiple events (host) over time. Through the overview, the presence of many vertically aligned radar charts (with clustering and ordering) shows that there are abnormal activities around 3 am, 7 pm, and 10 pm where the system changes from normal activity to high memory usage. This information can be used for further investigation, such as which job causes the issues and who owns the corresponding jobs. Knowing this information in advance can help the system administrator have better resource allocations and management strategies.

B. *US employment data*

The US employment dataset is about the number of employees in 51 states and 15 industrial fields from February 2000 to May 2019. Original data come from the US Bureau of Labor Statistics website [23]. Instead of the raw number of employees, the net change has been using in the *TimeRadar* visualization Figure 2. *TimeRadar* presents the events by the density of the radar appear on the top chart. The first orange markers in *TimeRadar* matched the Y2K crisis (2001) when the dotcom bubble collapsed. The second markers correspond with the Great Recession of 2007–2009 [24]. The final marker shows that the COVID-19 has a significant impact on the economy and the effect is even more than the Great Recession (base on the size of the radar).

C. *Rest Useful Life data*

The Rest Useful Life data includes 100 simulated turbofan engines with ten sensors and 200 timesteps. In particular, we use the run-to-failure datasets of a turbofan engine simulation model obtained from NASA's Prognostics Center of Excellence in 2008 [25]. In Figure 3 (a), data has been clustered into

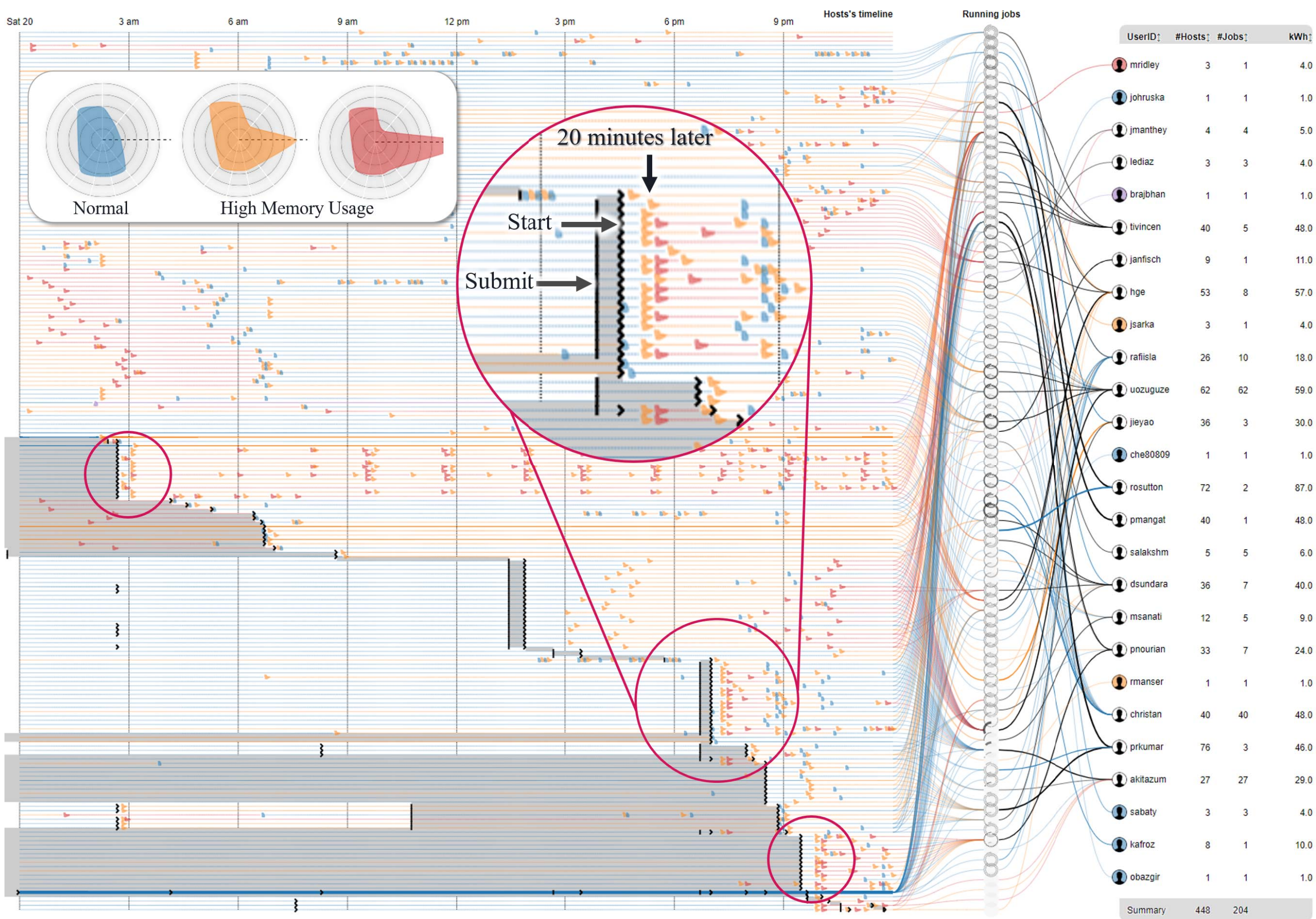

Fig. 1. A usecase of *TimeRadar* on HPCC dataset: Red circles indicates enclose the time point when many instances change their status

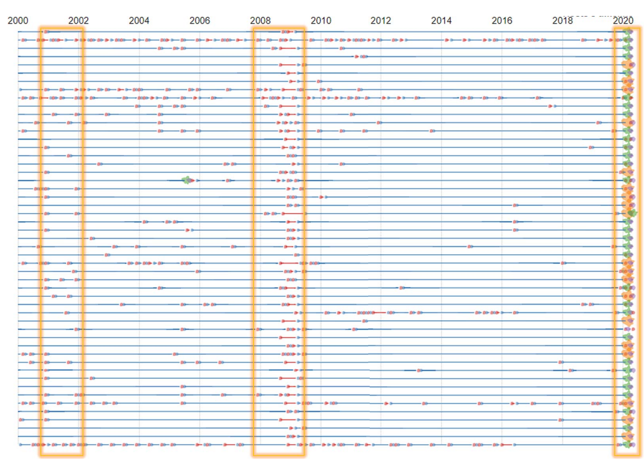

Fig. 2. The *TimeRadar* visualizations for US employment net rate: Orange marker - US crisis in 2001 and Great Recession 2007-2009 and COVID-19.

seven groups. According to Figure 3 (b), the second cluster (orange) and the third cluster (blue) is the common state of the data. *TimeRadar* clearly shows that the three last groups (in the red box) are the sign of RUL run out.

IV. User study:

Our method to evaluate *TimeRadar* was conducted using the *controlled experiment* where some factors are changed at a time. Sixteen users were asked to answer readability questions for both representations of the same dataset varying in complexity (size and dimension). Time taken to answer each question, error, and feedback were recorded for analysis.

2 Visualizations (TimeRadar, Bundling) x 3 Levels of size x 3 Levels of timepoint x 2 Tasks = 36 questions in total.

A. Participants and Apparatus

Sixteen participants were recruited in this study, including seven females and nine males, at the study site. Their educational backgrounds are ranged from undergraduate to Ph.D. (8 Ph.D. students, two undergraduates, and 6 Masters). As studied by Camilla [26], when it comes the how many experimenters to engage in the evaluation, the number of recommended experts is five; we considered our recruited users were experts since all of them took the Human-Computer Interaction classes. The study was conducted using a laptop (equipped with a 15.4-inch screen of 1920 x 1080 in resolution).

Fig. 3. The *TimeRadar* visualizations for Rest Usefull Life: (a) Major radar represent cluster. (b) Histogram of instance value in each cluster, (c) *TimeRadar* of RUL data with RUL value decrease from left to right.

B. Visualizations and Datasets

For our dataset, we defined the level of difficulty by the number of instances and the number of time points. As the number of instances and time points increases, the difficulty levels are linearly increased.

For each example, we generated two figures of *TimeRadar* vs. edge bundling. To ensure that our tasks were isomorphic with both *TimeRadar* and Bundling, the same data was used in both static figures. However, to avoid memorizing the result from the previous question, we shuffled the set of questions.

C. Tasks

In this study, we want to find out, which is a more efficient way to capture the dynamics of a system via two tasks.

- T1: Find the most dynamic instance (the instance that exhibits multiple statuses over time).
- T2: Find the most dynamic time point (the timestamps, when many instances change their status).

D. Procedure

We performed the study with multiple participants at a time. The study is about 20 minutes in total, with a set of 36 questions. We set up our own page on Github to collect data. Responses from users were saved on the cloud repository, which is Firebase from Google Service. JSON data format from the repository was converted into Excel (.csv) for analysis. Before the controlled experiment, all participants were given a tutorial explaining *TimeRadar* and Bundling to make sure that they understood the concepts of visual design, how data was transformed to visual encoding, how color was mapped. All participants were informed why the study was conducted and what type of information was collected before starting the survey. The tutorial was provided by means of direct conversation with the conductor or watching the introduction video created by the conductor. The video was embedded directly at the beginning of the survey page. Next, the participants used the provided computers that showed the figure represented by either *TimeRadar* or Edge Bundling method and answered question associated with it. When a user started a new question, our survey page started recording time spent on that question.

The order of tasks was dynamic - meaning that all questions were randomized in order to avoid memorization. After completing 36 questions, users were asked to provide feedback on each visualization design in terms of readability and mental efforts. Level of education and gender were also collected for categorization.

E. Hypotheses

Our study has the following Hypotheses:

- **H1**: Superimposing will outperform Bundling in completion time for discerning the dynamic tasks (T1, T2). We also believe that Superimposing requires less time for these types of tasks, as visually detecting the color-coded radars is much easier and quicker than following/tracking the curves.
- **H2**: Bundling will outperform Superimposing in accuracy and completion time for overview tasks (T3). The group sizes in the Bundling option are easier to discern as groups are ordered vertically.

F. The results

We considered time and accuracy are both equally important for the evaluation. Because when participants were given enough time, the likelihood to get the correct answers was high with enough mental effort. Since visualization helps people carry tasks easier, time should be taken into account to measure this criterion. 2 Visualizations (*TimeRadar*, Bundling) x 3 Levels of size (10, 30, 50) x 3 Levels of timepoint (10, 30, 50) x 2 Tasks (finding instance and finding timepoint). Repeated Measures Analysis of Variance (RM-ANOVA) method was used to analyze accuracy and completion time results. The analysis is within-subject design.

G. Accuracy

On average, for all cases, 64% of the total answers from both tasks were correctly answered. Users achieved an accuracy of 88.09% for identifying the dynamic instances using the superimposing technique. However, the performance is worst for the same task using the bundling visualization technique on the same data with an accuracy of 21.42%. For this task, *TimeRadar* were 66.67% more accurate than the bundling chart.

Regarding the overview task, the performance of using the superimposing technique was decreased with an accuracy of 79.52%, while the performance of using the Bundling technique was increased significantly from 21.42% to 64.56%. The accuracy difference between these two techniques was 14.96%.

H. Completion Time

On average, the task for finding dynamic instances using superimposing took 15.03 seconds (SD =10.56) to complete, and the task for finding dynamic instances using the bundling method took 20.98 seconds (SD = 12.2). Overall, the difference between means of superimposing and bundling is statistically significant, F(1, 126) = 28.68, p ¡ .0001. This result supported our hypothesis (**H1**). In detailed, the study found a difference between the level of size 10 and level of time point 30 (F(1,13) = 9.52, p = 0.009), the level of size 30 and level of time point 50 (F(1,13) = 5.67, p = 0.03), and the level of size 50 and level of time point 50 (F(1,13) = 5.68, p = 0.033) with the significant level of 0.5.

Regarding the second task (the overview), there is no statistically significant difference between using superimposing and bundling techniques, F (1, 125) = 1.99 and p = 0.161 with a significant level of 0.5. The average time taken on answering questions associated with superimposing technique was 13.76 seconds, and for Bundling was 15.05 seconds. Unsurprisingly, as the number of sizes and time points increased, it took more time for users to answer. The distribution of time spent on tech type of charts is depicted in Figure 4 where the red vertical line represents the median.

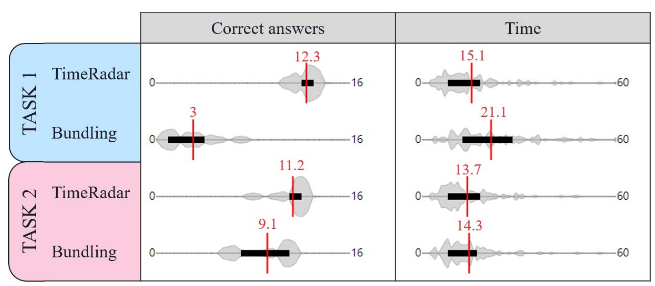

Fig. 4. The contingency table that shows the distribution of time spent and correct answer of two visualization charts

I. User feedback

Users provided feedback based on their preferences after completing all tasks. We collected all the comments and summarized them for qualitative analysis. Overall, users prefer to use *TimeRadar* to answer questions on the survey because identifying shape is easier than use color, especially when the radars were aligned horizontally and guided by a timeline.

The bundling technique needs more mental effort to trace the dynamic instance; this issue gets worst when multiple instances overlapped when they changed their statuses. One user reported that *'I gave up when I tried to answer the questions related to trace a dynamic instance using Bundling visualization, for a small figure, I can look up each instance one by one; however, when there are so many instances it is impossible for me. I just picked a random answer, so I chose the one that shows the change first'* or *'I prefer the one without the transitioning lines because you can count shapes and it is much easier to see the changes.'*, or *"A time radar chart is much easier to follow since it has less object and intersection. A time bundling chart is really hard to read in 60 seconds to find the answer*. This feedback was aligned with the accuracy result of 21.42%. For an overview task, participants still reported that superimposing visualization is more favorable because *'at least I can count the number of glyphs vertically, for bundling there are so many crossings'*, the visual clutter makes them end up with *'time radar, I made a random guess on bundle chart. Only pick the first one from the left that shows change'*.

V. Conclusion and Future work

This paper presented a novel technique for visualizing temporal event sequences. Our proposed approach was based on the clustering and superimposing techniques to construct a high-level overview of sequential event data. *TimeRadar* allows domain analysts to simultaneously capture sequence clustering, extract useful distribution patterns, drill multiple levels of detail to accelerate interactive data analysis. The *TimeRadar* was demonstrated through case studies with real-world temporal datasets. Our approach was evaluated by 30 participants to measure its efficiency and effectiveness. The quantitative result showed that superimposing technique out-performed the bundling technique in terms of completion time and accuracy for the task of tracing dynamic instances. User feedback indicated that superimposing was a better choice for exploring interesting visual patterns. Although analysts can apply *TimeRadar* on the variety of domains, it still has some limitations: since users perceived the radar chart as an area, change in one dimension will increase area, and 2) order the bundling lines is challenging because of the crossing or intersection, resulting in visual clutter. We try to address this issue in future work.

VI. Acknowledgments

The authors acknowledge the High-Performance Computing Center (HPCC) at Texas Tech University [27] in Lubbock for providing HPC resources and data that have contributed to the research results reported within this paper. This research is supported in part by the National Science Foundation under grant CNS-1362134, OAC-1835892, and through the IUCRC-CAC (Cloud and Autonomic Computing) Dell Inc. membership contribution.

References

[1] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman, "Lifelines: visualizing personal histories," Tech. Rep., 1995.

[2] K. Wongsuphasawat and D. Gotz, "Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2659–2668, 2012.

[3] Z. Shen, J. Wei, N. Sundaresan, and K.-L. Ma, "Visual analysis of massive web session data," in *IEEE symposium on large data analysis and visualization (LDAV)*. IEEE, 2012, pp. 65–72.

[4] J. A. Ferstay, C. B. Nielsen, and T. Munzner, "Variant view: Visualizing sequence variants in their gene context," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2546–2555, 2013.

[5] F. Du, B. Shneiderman, C. Plaisant, S. Malik, and A. Perer, "Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 6, pp. 1636–1649, 2016.

[6] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman, "Temporal event sequence simplification," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2227–2236, 2013.

[7] B. C. Kwon, J. Verma, and A. Perer, "Peekquence: Visual analytics for event sequence data," in *ACM SIGKDD 2016 Workshop on Interactive Data Exploration and Analytics*, vol. 1, 2016.

[8] Z. Liu, H. Dev, M. Dontcheva, and M. Hoffman, "Mining, pruning and visualizing frequent patterns for temporal event sequence analysis," in *Proceedings of the IEEE VIS 2016 Workshop on Temporal & Sequential Event Analysis*, 2016, pp. 2–4.

[9] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson, "Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 321–330, 2016.

[10] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao, "Unsupervised clickstream clustering for user behavior analysis," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 225–236.

[11] J. Hartigan, *Clustering Algorithms*. New York: John Wiley & Sons, 1975.

[12] C. Plaisant and B. Shneiderman, "The diversity of data and tasks in event analytics," in *Proceedings of the IEEE VIS 2016 Workshop on Temporal & Sequential Event Analysis*, 2016.

[13] 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.

[14] 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.

[15] Y. Tanahashi and K. Ma, "Design considerations for optimizing storyline visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2679–2688, Dec 2012.

[16] M. Krstajic, E. Bertini, and D. Keim, "Cloudlines: Compact display of event episodes in multiple time-series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2432–2439, Dec 2011.

[17] T. Dang and V. T. Nguyen, "ComModeler: Topic Modeling Using Community Detection," in *EuroVis Workshop on Visual Analytics (EuroVA)*, C. Tominski and T. von Landesberger, Eds. The Eurographics Association, 2018.

[18] T. N. Dang, P. Murray, and A. G. Forbes, "PathwayMatrix: Visualizing binary relationships between proteins in biological pathways," *BMC Proceedings*, vol. 9, no. 6, p. S3, 2015.

[19] M. Meyer, T. Munzner, and H. Pfister, "Mizbee: A multiscale synteny browser," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 897–904, Nov. 2009. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2009.167

[20] D. Kammer, M. Keck, T. Gründer, A. Maasch, T. Thom, M. Kleinsteuber, and R. Groh, "Glyphboard: Visual exploration of high-dimensional data combining glyphs with dimensionality reduction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 4, pp. 1661–1671, 2020.

[21] J. A. Hartigan, "Clustering algorithms," 1975.

[22] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/331499.331504

[23] B. of Labor Statistics, "State and area employment, hours, and earnings – seasonal, 2000-2020 [time series]," https://data.bls.gov/cgi-bin/dsrv?sm, 2020.

[24] E. Cunningham, "Great recession, great recovery-trends from the current population survey," *Monthly Lab. Rev.*, vol. 141, p. 1, 2018.

[25] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," *NASA Ames Prognostics Data Repository*, 2008.

[26] C. Forsell, "Evaluation in information visualization: Heuristic evaluation," in *2012 16th International Conference on Information Visualisation*. IEEE, 2012, pp. 136–142.

[27] HPCC. (2020) High Performance Computing Center. [Online]. Available: http:www.depts.ttu.edu/hpcc/