

EventColumn: Integrating Event Sequences into Tabular Visualizations

Jakob Zethofer*
Pro2Future GmbH

Andreas Hinterreiter†
JKU Linz

Lukas Schiefermüller‡
voestalpine Stahl GmbH

Belgin Mutlu§
Pro2Future GmbH

Marc Streit¶
JKU Linz



Figure 1: EventColumn provides visual decision support for event-based data, for example, in production logistics, where it helps users estimate the storage duration of production items. EventColumn displays event sequences alongside conventional tabular attributes (e.g., warehouse location) in a single table row. Events are displayed on a shared timeline (1). Grouping event rows by type summarizes them in heatmaps (2). A boxplot shows the distribution of specific event stages (3), in this case the storage duration of already produced items relative to the current production event. The column summary (4) shows the distribution of all events. EventColumn allows comparison of multiple event sequences by aligning and sorting them by event type (5). When users select an item in the main table, a linked table (6) is filtered to show details for items represented by the boxplots. Hovering over an item reveals detailed event timestamps in a tooltip (7).

ABSTRACT

We introduce EventColumn, a new column type that integrates event-sequence data with heterogeneous tabular attributes into a single unified table. EventColumn lets analysts compare event sequences alongside numerical, categorical, and temporal attributes at both instance and group levels, offering a compressed overview, heatmap group summaries, alignment by event types, and boxplots of similar historical items. We developed EventColumn together with collaborators from the steel industry to facilitate the analysis of production events and warehouse logistics, but the solution applies to other event-sequence domains. Unlike most existing ap-

proaches that compare either event sequences or tables, EventColumn supports simultaneous comparison of both. We demonstrate its integration with Taggle and Microsoft Power BI on data from steel production logistics and on a public e-commerce dataset.

Index Terms: Event sequence visualization, tabular visualization, visual analytics, decision support, production logistics.

1 INTRODUCTION

In multi-stage manufacturing settings, successive production stages are often distributed across different locations and points in time, with products stored in warehouses between stages. When a warehouse reaches capacity, items must be relocated to external storage facilities. Optimal relocation decisions require identifying items that will be stored the longest to reduce the total number of transports and avoid damage from additional handling. Such relocation decisions can benefit from a visual decision support system.

We developed our approach in collaboration with a steel manufacturer, focusing on logistics between the hot rolling and pickling (i.e., coil surface acid treatment) production steps. In the current process, logistics employees assess which coils to move to outdoor storage by evaluating the remaining storage duration based on the

*E-mail: jakob.zethofer@pro2future.at

†E-mail: andreas.hinterreiter@jku.at

‡E-mail: lukas.schiefermueller@voestalpine.com

§E-Mail: belgin.mutlu@pro2future.at

¶E-mail: marc.streit@jku.at

coil’s characteristics and relevant events, such as the shipping date and the hot rolling date. The expected storage duration before pickling is often unclear, as pickling is typically scheduled only about three days in advance, and coils require the same cooling period after hot rolling. Training a machine learning model for predicting storage duration did not yield reliable results. We experimented with multiple different clustering algorithms, regression and classification models, and inspected the most promising results with a detailed SHAP analysis [13]. The results suggested that human factors and non-standardized planning decisions were not fully captured in historical data. We therefore designed a visual analytics approach that keeps the human in the loop by providing domain experts with a dashboard that helps them estimate a coil’s storage duration.

Supporting this task requires comparing candidate coils across two types of data: (i) *event sequences* containing the dates of production-related events such as hot rolling completion, planned galvanizing, or shipping, and (ii) *heterogeneous tabular attributes* such as steel category, coil dimensions, and warehouse location. Comparing each candidate against similar historical coils whose actual storage duration is known further aids decision-making. While event sequence visualization [9] and tabular visualization [4] are both well-researched areas, existing tools do not display event sequences and heterogeneous tabular attributes together in a single table. Instead, they separate them into different views or tabs [21, 5], which makes comparison of items more difficult.

We present **EventColumn**, a new column type that displays event sequences alongside numerical, categorical, and temporal attributes. To demonstrate our approach, we integrate EventColumn with Taggle [6], a tabular visualization technique, and Microsoft Power BI [14]. We showcase the functionality of EventColumn on a steel production logistics dataset and demonstrate its generalizability on an e-commerce dataset.

2 RELATED WORK

We position our work at the intersection of visual analytics for decision support in manufacturing, event sequence visualization, and tabular data visualization.

2.1 Visual Analytics for Decision Support

Visualization plays an increasingly important role in industrial decision-making. Zhou et al. [27] surveyed visualization approaches for smart manufacturing, identifying the production phase—where our work is situated—as one of the key stages that benefits from visual analytics. Schreck et al. [19] discussed challenges of deploying visual data science in industrial settings, highlighting the need to cope with the four Vs of big data (volume, velocity, variety, veracity) and to integrate domain expertise. Tools like ViDX [24] and LiveGantt [12] address assembly-line diagnostics and scheduling visualization, respectively, but focus on the overall production process rather than inter-step logistics.

2.2 Event Sequence Visualization

Event sequence visualization has been extensively surveyed. Guo et al. [9] provide a comprehensive design space covering data scales, visual representations, and interaction techniques. Van der Linden et al. [20] extend this design space by including dimensions such as granularity, positioning, and alignment for comparative analysis. Yeshchenko and Mendling [25] propose the ESeVis framework, distinguishing between instance and model representations across different visualization types.

Existing tools include LifeLines 2 [22], which aligns and ranks event records by sentinel events; EventFlow [15], which aggregates and simplifies temporal event sequences; and Similan [23], which supports similarity-based comparison of categorical event sequences. Several systems augment events with additional data: ChronoCorrelator [21] enriches events with global density plots and

filter widgets, and EmoCo [26] combines event timelines with bar code charts summarizing emotions. However, existing approaches separate event visualizations from tabular attribute displays: for example, TempViz [5] uses separate tabs for events and attributes, while Guo et al. [8] show events and multivariate data in separate linked views. None of these tools unify event sequences and heterogeneous tabular columns within a single table layout.

2.3 Tabular Visualization

Dimara et al. [4] found that tabular visualizations slightly outperformed parallel coordinates and scatterplots for decision-support tasks involving multi-attribute comparison. Bartram et al. [1] highlighted the importance of tables to many users for viewing the unaggregated data. Both of these studies motivate our choice of a table-based layout. LineUp [7] introduced interactive visual analysis of multi-attribute rankings, encoding numerical values as bar charts within table columns and supporting flexible sorting and weighting. Taggle [6] extends LineUp with grouping, aggregation, and an overview mode that compresses rows to a single pixel height, enabling the display of hundreds of items simultaneously. Our work extends Taggle by introducing a new EventColumn that embeds event-sequence data directly into the table, thereby combining the strengths of tabular and event-sequence visualization in a unified layout.

3 TASK ABSTRACTION AND DESIGN GOALS

In this section, we characterize our collaborators’ existing workflow, derive abstract analysis tasks, and discuss the resulting design goals for EventColumn.

3.1 Application Context

We gathered the details of the decision-making process for coil relocation described in Section 1 through multiple meetings and a plant tour with employees from the logistics department. Based on our analysis, we identified a four-step target workflow.

In **Step 1**, the logistics employee focuses on coils stored in a warehouse approaching capacity, excluding those not suitable for relocation (e.g., due to urgency flags). **Step 2** involves identifying coils likely to require a long storage period, based on criteria such as a shipping date far in the future. In **Step 3**, the remaining candidate coils are compared in more depth: categorical, numerical, and temporal attributes are evaluated alongside the distribution of storage durations of similar coils from historical data. In **Step 4**, selected coils are scheduled for transport to the external storage location.

3.2 Task Abstraction

We performed a task abstraction [16] of the identified workflow, resulting in the following abstract visualization tasks:

T1 Filter items based on user-defined characteristics. For example, the logistics employee browses the list of current coils and identifies candidates for relocation based on known criteria, such as a shipping date far in the future or a particular product type.

T2 Select items through comparison with items from the present and past via four subtasks:

T2.a Inspect similar items from historical data based on user-defined attributes.

T2.b Summarize the target quantity (e.g., storage duration) of similar items.

T2.c Compare attributes of similar items (categorical, numerical, and temporal) to assess the reliability of the estimate.

T2.d Compare attributes of candidate items to select the most promising ones (discover–explore–compare). This includes event sequences, target quantity distribution, and other attributes.

3.3 Design Goals

We identified the following design goals based on the defined abstract tasks and target workflow.

D1 Display event sequences and heterogeneous tabular attributes in a unified table with one row per item, so events and other data can be compared in one place (**T2**).

D2 Show the distribution of the target quantity from similar historical items along with the current item's events (**T2.b**).

D3 Enable one-to-many comparison between items for event sequences and other attributes (**T2.c**, **T2.d**).

D4 Enable aligning and sorting by event types for better comparison of all attributes (**T2.c**, **T2.d**).

4 VISUALIZATION DESIGN

Over multiple iterations with our collaborators, we designed and implemented a decision-support dashboard that addresses the design goals outlined above. We first describe the entire dashboard, before providing details on EventColumn.

4.1 Dashboard Design

Filters at the top of the dashboard allow filtering by warehouse, shipping date range, and an urgency-flag to exclude coils not suitable for relocation (**T1**). Below the filters, the main EventColumn visualization (Figure 1 top) displays all candidate coils including their events and attributes (**T2.d**) as well as a distribution of the target quantity (**T2.b**) from similar historical items (**T2.a**). Similar items are passed to EventColumn as lists of IDs per row. For the steel production use case, similar items were precomputed using a proprietary similarity metric based on multiple coil properties. When the user selects a coil in the main visualization, another instance of the same visualization at the bottom of the dashboard displays the selected coil's past similar items, allowing direct comparison of their events and attributes, and assessment of the reliability of the storage duration estimate (**T2.c**).

4.2 EventColumn Design

Figure 1 shows the main visualization including the EventColumn, which is the central contribution of this work. It extends the Taggle [6] tabular visualization technique with a new column type that embeds event sequences directly in the table (**D1**). Each row represents one item. Events are drawn as colored circles in each cell, with color encoding the event type. A shared axis is displayed in the column header. A boxplot of storage durations for similar historical items (**T2.a**) is overlaid in the same event column (**D2**). As all attributes and events share one row per item, they can be compared by visual juxtaposition [20] (**D3**). Colors are taken from qualitative ColorBrewer palettes [3], using saturated colors for event types and unsaturated colors for categorical columns.

Data scale and visual representation. As data scale we chose *sequence* [9] for comparing all events. The visual representation is timeline-based and instance-based [25], using a *fixed* shared time axis as in LifeLines 2 [22] and ChronoCorrelator [21]. EventColumn expects each event type to occur at most once per item. We discuss this design choice and the resulting limitations in Section 7.

Overview mode. Enabling overview mode (bottom visual in Figure 1) compresses each row to one pixel in height, allowing hundreds of items to be visible simultaneously (**D3**). A selected row remains at full height for detailed inspection, whereas all other rows are condensed. Compressed event sequences are rendered as small rectangles, and only a single user-chosen boxplot statistic (e.g., median) is shown per row to avoid clutter.

Grouping and aggregation. Rows can be grouped by any column; grouped event columns are summarized with per-event

heatmaps, as shown in Figure 1-2 (**D3**). The time axis is binned and event frequency is encoded by color intensity, inspired by EmoCo's per-channel plots [26]. Categorical and numerical columns are aggregated as histograms and boxplots, respectively.

Alignment, sorting, and interaction. By default, events are placed relative to the current time. Users can realign the axis to any event type (**D4**), similar to LifeLines 2 [22] and EventFlow [15], making durations between specific events directly comparable across rows. The boxplot can be aligned to any event type, showing the distribution of similar items' storage duration relative to that event. The event column can be sorted by any event type or boxplot feature, such as shipping time or median storage duration of similar items, to rank items (**D4**). The event column further supports the interactions *scaling* (zoom and pan on the time axis), *segmentation* (show/hide event types), and *emphasis* (column highlighting and per-event color remapping) [9, 20].

Design alternatives considered. An earlier design displayed similar coil events in a tooltip on hover rather than in a separate visualization, but was rejected because the second visualization enables more effective user interaction and supports more detailed exploration. A second draft separated events and tabular data into linked views as in TempViz [5]; this was rejected because it breaks the perceptual link between a coil's events and attributes and prevents column-based sorting and grouping (**D1**).

5 IMPLEMENTATION

The implementation comprises two main parts: extending the Taggle implementation with the EventColumn and integrating the result into a Power BI custom visual. Both parts were implemented in TypeScript.

5.1 EventColumn

The *EventColumn* extends Taggle's *MapColumn*, using the event type as map key and a timestamp number as value; boxplot statistics are precomputed and stored under named keys (e.g., *median* and *q1*). A *getEventValue* method resolves each value relative to a configurable reference event and divides by a configurable time unit (default: one day), so alignment by any event type is handled uniformly for both events and boxplot entries. A custom cell renderer handles all three Taggle aggregation levels. At the *summary* level it renders the shared time axis in the column header using a D3 [2] linear scale. At the *group* level each event type is binned by time using D3's binning function, producing a per-event heatmap that encodes frequency by color intensity. At the *cell* level events are drawn as SVG circles at their scaled horizontal position. Boxplots are rendered as SVG lines and rectangles. In overview mode they are drawn as four-pixel HTML Canvas squares that Taggle can compress to fit more rows. A sort-dialogue add-on exposes all event types and boxplot statistics as sort keys, and a dedicated event-settings dialogue lets users toggle visible event types and boxplot overlay, and change the reference event.

5.2 Power BI Custom Visual

The custom visual is implemented in TypeScript using the Power BI custom visual SDK. Its data interface defines five input fields: *Data Columns* (general tabular attributes), *Event Data* (one date column per event type), *Similar Data Duration* (semicolon-separated storage durations of similar items), *Similar Data IDs* (IDs of similar items for cross-filtering), and *ID* (a unique row identifier for storing selections). On each data update from Power BI, the visual infers the data type of each *Data Columns* field and creates the corresponding Taggle column type (boolean, number, categorical, or date). *Event Data* columns are converted to milliseconds and passed to the *EventColumn* as event key-value pairs. For *Similar Data Duration*, the five-number boxplot summary (min, Q1, median, Q3, max) is computed and appended to the row's event data.

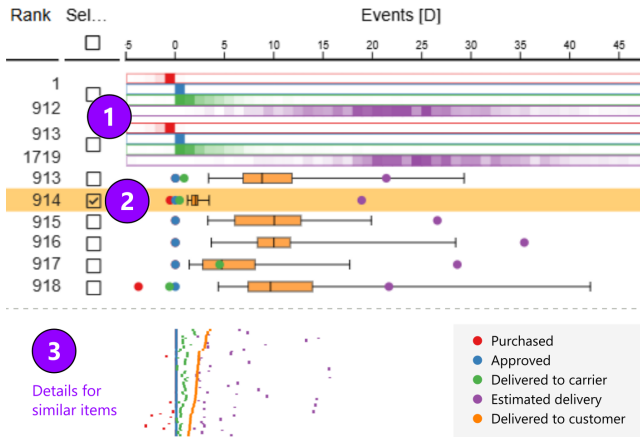


Figure 2: E-commerce analysis dashboard showing delivery events, location, price, and freight value aligned by order approved. We see that the estimated delivery date for items varies between high- and low-value freight (1). The user selected one order where the estimated delivery date was much later than for recent deliveries to the same city (2). The analysis of these recent deliveries shows that the estimated delivery date is much later than the actual delivery date (3).

Taggle’s state can be saved via a button at the bottom of the visual to preserve user interactions across data updates.

The EventColumn Power BI visual is published at the Microsoft Marketplace [18] (under the name EventTableView). The source code and example dashboards are available on GitHub [10]. Documentation and usage instructions are available online [11].

6 USE CASES

In this section, we describe two use cases for EventColumn. The first use case is based on a logistics dataset from our collaborators in the steel industry. For the second use case, we apply EventColumn to the publicly available Brazilian E-Commerce Dataset [17].

6.1 Steel Production Logistics

The primary use case motivating this work is the steel production logistics scenario described in Section 3: logistics employees must decide which coils to relocate to external storage before the warehouse reaches capacity limits, which requires estimating each coil’s remaining storage duration. The dashboard (Figure 1) allows users to filter for coils by warehouse and shipping date, and to exclude those with an urgency flag (T1). Then they can compare the displayed coils by their attributes, events (T2.d), and storage duration of similar coils (T2.b). If they are unsure about the boxplot distribution’s relevance for the current coil, they can select it to see the data of the coils behind the distribution (T2.c). Finally, they select coils for relocation based on their experience and the attributes shown in the dashboard. Because the production data is confidential, Figure 1 uses a synthetic dataset reflecting the main characteristics of the original.

6.2 Brazilian E-Commerce Dataset

To demonstrate the EventColumn’s capabilities on a publicly available dataset from a different domain, we use the Brazilian E-Commerce Dataset [17], which contains 100,000 orders placed at multiple Brazilian marketplaces between 2016 and 2018. Each order has the following events: *purchase*, *approved*, *delivered to carrier*, *delivered to customer*, and *estimated delivery date*. These are accompanied by tabular attributes such as *customer city*, *customer state*, and *freight value*. The boxplot displays the distribution of the time between *approved* and *delivered to customer* for recent

orders to the same city. Figure 2 shows the resulting dashboard for analyzing delivery times of orders that have not yet been delivered, by adding the duration of the most recent deliveries in the same city to the boxplot data. We first group by freight value to assess whether delivery times differ between high- and low-value freight (1). This shows significantly lower estimated delivery dates for low-value freight items. We then select one order with a high-value freight where the estimated delivery date is much later than the distribution of recent delivery durations to that city (2). The analysis of recent deliveries to that city (3) shows that the estimated delivery date for these orders is later, although all items were delivered within 3 days. This indicates that the selected order will be delivered much earlier, and the estimated delivery date is unreliable for this order.

7 DISCUSSION

In this section we reflect on preliminary feedback by our collaborators and discuss potential for future work.

7.1 Preliminary Feedback

A logistics employee of our collaborator evaluated the dashboard in parallel with the existing tool on real production data for one month—a time range that typically covers decisions for several thousands of coils. They appreciated “all important columns are shown” in one view. They liked how easy it was to filter for coils in certain width ranges, steel categories, and with a planned galvanizing date. They evaluated their decisions after the coils had been stored and found that they worked well in many cases, but also encountered cases where their estimate was far off. They attributed this to manual planning of the next production steps that does not take the storage location into account. In summary, the user concluded that they would like to see “something similar integrated in a new [logistics] system”, which was in its early planning state at the time of evaluation.

7.2 Scalability & Limitations

The choice to distinguish event types per color limits EventColumn’s applicability to around 10 different types of events, and Power BI’s fixed query limit is 1 million rows. Overview mode can help avoid visual clutter in cases of large datasets. However, each item requires at least one pixel of vertical space, and scrolling is introduced if the number of items is larger than the available vertical pixel space. The width of each event mark in overview mode is constrained to four pixels, which can lead to problems when events are very close together within a very large overall time range.

Currently, the main conceptual limitation of EventColumn is that recurring events per item are not supported. Enabling alignment and sorting of rows would require more complex interactions or aggregation. As a workaround, multiple events can be aggregated in preprocessing or stored with different labels in cases of fixed event counts per item. Other minor limitations—e.g., that filtering requires adding an event as a separate date column—result directly from basing EventColumn on the Taggle system [6].

8 CONCLUSION

This paper introduced the EventColumn: a new column type for tabular visualizations that enables analysis of event sequences, numerical, and categorical data without switching between visualizations. Event sequences can be compared both at the instance and group level. Aligning and sorting by event type, in combination with duration distributions from past events, enables informed decision-making. We demonstrated the visualization’s capabilities on datasets from two domains. Our work shows that general-purpose tabular frameworks can be extended to natively accommodate event sequence data, illustrating a path toward richer mixed-type column libraries for interactive data exploration.

ACKNOWLEDGMENTS

This work has been supported by the BMIMI, BMWET, and FFG, Contract No. 911655: “Pro2Future II”.

REFERENCES

- [1] L. Bartram, M. Correll, and M. Tory. Untidy data: The unreasonable effectiveness of tables. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):686–696, 2022. doi: 10.1109/TVCG.2021.3114830 2
- [2] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185 3
- [3] C. A. Brewer, G. W. Hatchard, and M. A. Harrower. ColorBrewer in print: A catalog of color schemes for maps. *Cartography and Geographic Information Science*, 30(1):5–32, Jan. 2003. doi: 10.1559/152304003100010929 3
- [4] E. Dimara, A. Bezerianos, and P. Dragicevic. Conceptual and methodological issues in evaluating multidimensional visualizations for decision support. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):749–759, Jan. 2018. doi: 10.1109/TVCG.2017.2745138 2
- [5] J. A. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, pp. 167–174, Oct. 2006. doi: 10.1109/VAST.2006.261421 2, 3
- [6] K. Furmanova, S. Gratzl, H. Stütz, T. Zichner, M. Jaresova, A. Lex, and M. Streit. Taggle: Combining overview and details in tabular data visualizations. *Information Visualization*, 19(2):114–136, Apr. 2020. doi: 10.1177/1473871619878085 2, 3, 4
- [7] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, Dec. 2013. doi: 10.1109/TVCG.2013.173 2
- [8] R. Guo, T. Fujiwara, Y. Li, K. M. Lima, S. Sen, N. K. Tran, and K.-L. Ma. Comparative visual analytics for assessing medical records with sequence embedding. *Visual Informatics*, 4(2):72–85, June 2020. doi: 10.1016/j.visinf.2020.04.001 2
- [9] Y. Guo, S. Guo, Z. Jin, S. Kaul, D. Gotz, and N. Cao. Survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):5091–5112, Dec. 2022. doi: 10.1109/TVCG.2021.3100413 2, 3
- [10] JKU Visual Data Science Lab and J. Zethofer. EventColumn GitHub repository (Event Table Viewer Power BI Visual). <https://github.com/jku-vds-lab/pro2future-event-table-viewer>, 2026. Last accessed: 2026/05/01. 4
- [11] JKU Visual Data Science Lab, J. Zethofer, and Pro2Future GmbH. Event Table Viewer Power BI Visual. <https://jku-vds-lab.at/pro2future-event-table-viewer>, 2024. Last accessed: 2026/05/01. 4
- [12] J. Jo, J. Huh, J. Park, B. Kim, and J. Seo. LiveGantt: Interactively visualizing a large manufacturing schedule. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2329–2338, Dec. 2014. doi: 10.1109/TVCG.2014.2346454 2
- [13] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon et al., eds., *Advances in Neural Information Processing Systems*, vol. 30, 2017. 2
- [14] Microsoft Corporation. Power BI. <https://www.microsoft.com/power-platform/products/power-bi>, 2026. Last accessed: 2026/05/01. 2
- [15] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227–2236, Dec. 2013. doi: 10.1109/TVCG.2013.200 2, 3
- [16] T. Munzner. *Visualization Analysis and Design*. CRC Press, Dec. 2014. 2
- [17] Olist and A. Sionek. Brazilian e-commerce public dataset by Olist, 2018. doi: 10.34740/KAGGLE/DSV/195341 4
- [18] Pro2Future GmbH. Event Table Viewer at Microsoft Marketplace. <https://marketplace.microsoft.com/en-ca/product/pro2futuregmbh1670847158359.event-table-viewer>, 2024. Last accessed: 2026/05/01. 4
- [19] T. Schreck, B. Mutlu, and M. Streit. Visual data science for industrial applications. In *Digital Transformation: Core Technologies and Emerging Topics from a Computer Science Perspective*, pp. 447–471. Springer, 2023. doi: 10.1007/978-3-662-65004-2_18 2
- [20] S. van der Linden, S. van den Elzen, and A. Vilanova. A survey on visual comparison of event sequence data. *Computer Graphics Forum*, 2023. doi: 10.1111/cgf.14856 2, 3
- [21] M. van Dortmont, S. van den Elzen, and J. van Wijk. ChronoCorrelator: Enriching events with time series. *Computer Graphics Forum*, 38(3):387–399, 2019. doi: 10.1111/cgf.13697 2, 3
- [22] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 457–466. ACM, 2008. doi: 10.1145/1357054.1357129 2, 3
- [23] K. Wongsuphasawat and B. Shneiderman. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 27–34. IEEE, 2009. doi: 10.1109/VAST.2009.5332595 2
- [24] P. Xu, H. Mei, L. Ren, and W. Chen. ViDX: Visual diagnostics of assembly line performance in smart factories. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):291–300, Jan. 2017. doi: 10.1109/TVCG.2016.2598664 2
- [25] A. Yeshchenko and J. Mendling. A survey of approaches for event sequence analysis and visualization. *Information Systems*, 120:102283, Feb. 2024. doi: 10.1016/j.is.2023.102283 2, 3
- [26] H. Zeng, X. Wang, A. Wu, Y. Wang, Q. Li, A. Endert, and H. Qu. EmoCo: Visual analysis of emotion coherence in presentation videos. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):927–937, Jan. 2020. doi: 10.1109/TVCG.2019.2934656 2, 3
- [27] F. Zhou, X. Lin, C. Liu, Y. Zhao, P. Xu, L. Ren, T. Xue, and L. Ren. A survey of visualization for smart manufacturing. *Journal of Visualization*, 22(2):419–435, 2019. doi: 10.1007/s12650-018-0530-2 2