This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

IVESA – Visual Analysis of Time-Stamped Event Sequences

Jürgen Bernard
University of Zurich
Digital Society Initiative

Clara-Maria Barth
University of Zurich
Digital Society Initiative

Eduard Cuba
University of Zurich

Andrea Meier
University of Zurich

Yasara Peiris
University of Zurich

Ben Shneiderman
University of Maryland

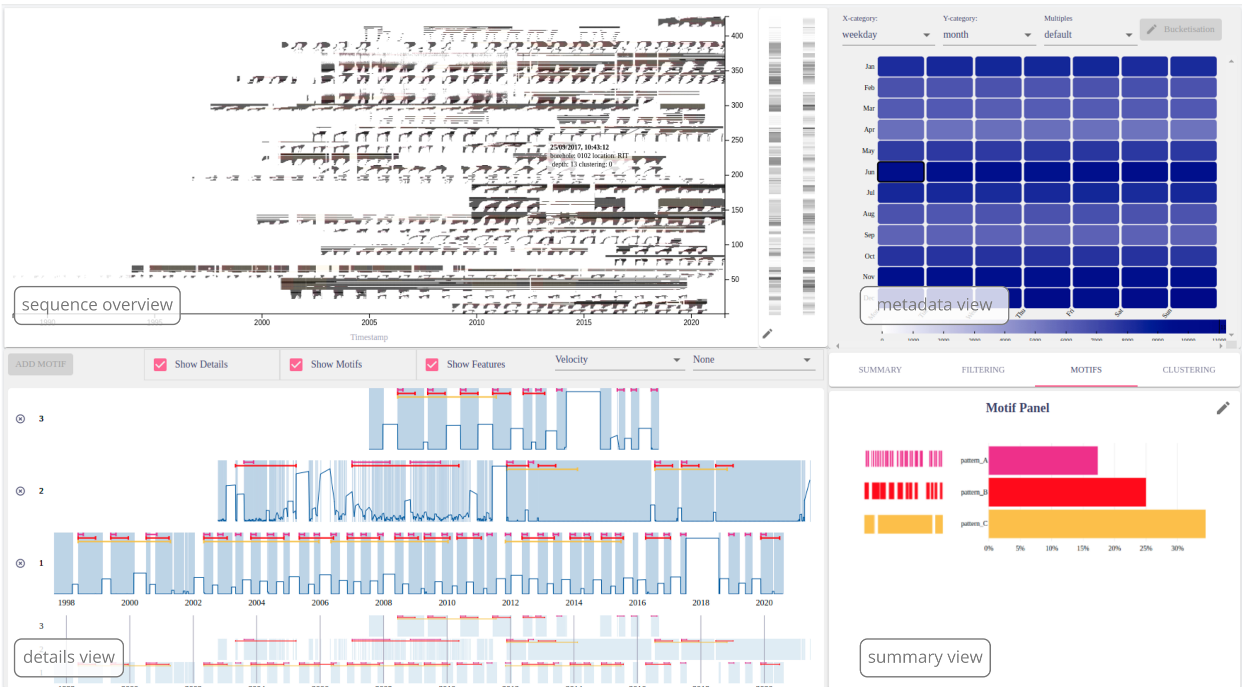

Fig. 1. Overview of IVESA. On the left, the *Sequence Overview* and *Details View* primarily enable the analysis of the TSEQs content, i.e., events, event sequences, groups of event sequences, motifs, and features. On the right, the *Metadata View* supports the analysis of metadata attributes and the TSEQs contextualization, whereas the *Summary View* includes the entry point to auxiliary views for filtering, motif configuration, feature analysis, and clustering.

Abstract—Time-stamped event sequences (TSEQs) are time-oriented data without value information, shifting the focus of users to the exploration of temporal event occurrences. TSEQs exist in application domains, such as sleeping behavior, earthquake aftershocks, and stock market crashes. Domain experts face four challenges, for which they could use interactive and visual data analysis methods. First, TSEQs can be large with respect to both the number of sequences and events, often leading to millions of events. Second, domain experts need validated metrics and features to identify interesting patterns. Third, after identifying interesting patterns, domain experts contextualize the patterns to foster sensemaking. Finally, domain experts seek to reduce data complexity by data simplification and machine learning support. We present IVESA, a visual analytics approach for TSEQs. It supports the analysis of TSEQs at the granularities of sequences and events, supported with metrics and feature analysis tools. IVESA has multiple linked views that support overview, sort+filter, comparison, details-on-demand, and metadata relation-seeking tasks, as well as data simplification through feature analysis, interactive clustering, filtering, and motif detection and simplification. We evaluated IVESA with three case studies and a user study with six domain experts working with six different datasets and applications. Results demonstrate the usability and generalizability of IVESA across applications and cases that had up to 1,000,000 events.

Index Terms—Time-Stamped Event Sequences, Time-Oriented Data, Visual Analytics, Data-First Design Study, Iterative Design, Visual Interfaces, User Evaluation

1 Introduction

Time-oriented data is among the most frequently analyzed data types. Many representations of time-oriented information contain both a timestamp and a value, leading to time-value pair relations [4, 5]. If values are of categorical type, researchers often refer to this data as *event sequences*, such as different types of treatments given to patients over

- *Jürgen Bernard is with University of Zurich. E-mail: bernard@ifi.uzh.ch.*
- *Clara-Maria Barth is with University of Zurich. E-mail: cbarth@ifi.uzh.ch*
- *Eduard Cuba is with University of Zurich. E-mail: eduard.cuba@uzh.ch.*
- *Andrea Meier is with University of Zurich. E-mail: andrea.meier5@uzh.ch.*
- *Yasara Peiris is with University of Zurich. E-mail: yasara.peiris@uzh.ch.*
- *Ben Shneiderman is with University of Maryland. Email: ben@cs.umd.edu*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

time. For numerical values (univariate or multivariate), the term *time series* is most often used, e.g., to track stock prices or sensor data over time. In contrast, this paper focuses on a sequential data type that has received surprisingly little attention so far: event data with only time information available *but no values*, i.e., the exploration of *temporal occurrences of events* is the primary analysis goal. We refer to this type as *time-stamped event sequences*, hereafter, in short: TSEQs. TSEQs exist in many application domains, such as the commit behavior in code repositories, heartbeats in healthcare, exercise times for personal workouts, communication events on the Internet of Things, tweets in social media, or Email communication.

One distinction between TSEQs and time series lies in the spacing of time stamps. Time series data are typically evenly spaced or preprocessed towards regular spacing to enhance comparability, with central analytical questions on *changes in values*. In contrast, for TSEQs central questions revolve around *changes to spacing over time*, opening up an interesting space of temporal analyses in ways that evenly spaced time series cannot. Another difference is the volume of data, which for TSEQs often significantly exceeds that of traditional time series, encompassing longer sequences with millions of events and possibly extending to thousands of TSEQs. The vast volume introduces complex analytical challenges, particularly in ranking TSEQs, according to developed metrics like changes in event frequency. For instance, one might seek to identify earthquakes by analyzing the rapid deceleration in aftershocks, ranking these natural events by the urgency or decline of their occurrences. Such analyses require sophisticated methods to manage and interpret large volumes of data, underscoring the unique challenges and opportunities presented by TSEQs.

One commonality of interest to domain experts is the temporal patterns of temporal event occurrences, which we call *motifs*, borrowed from motif discovery for time series [67] and classical event sequences [6]. Making sense of motifs for TSEQs is rooted in the application context. For that purpose, domain experts use environment parameters, sensor information, and other contextual attributes, which we call *metadata attributes*, inspired by the principle of separating time-oriented data content and auxiliary metadata [10]. Since metadata attributes co-exist with the TSEQs content, they can be used to characterize, explain, and contextualize motifs to draw conclusions. A simple example would be one's particularly healthy sleeping motif that may only be observed in the context of weekends and vacations. Given the plethora of real-world applications and cases, decisions made based on TSEQs can benefit our everyday life. Metadata contextualization of motifs can facilitate knowledge generation.

However, domain experts working with TSEQs are confronted with challenges. First, the data can be complex, both in the size of the dataset (sequences) and the size of individual sequences (events per sequence). Given these two problem dimensions, data collections can easily have millions of events, exceeding the human capabilities for manual processing. Also, existing tools for the exploratory analysis of time-oriented data do not typically enable users to respond to questions arising from these magnitudes of temporal event occurrences and motif patterns, but rather support the analysis of changing values, which do not exist for TSEQs. A second challenge comes with the data characteristics and features of TSEQs. Statistical metrics for time series exist from the data mining domain and would allow ranking TSEQs to ease their analysis, but have not been studied in the context of the interactive analysis of TSEQs. When domain experts combine visual exploration with metrics designed not for the assessment of value changes but temporal characteristics only, they are more likely to discover important features such as regularity, density, acceleration, or entropy of event occurrences. Third, since TSEQs have no values, domain experts can use metadata attributes to relate data findings to the contextual information, which would allow them to explain interesting patterns. However, solutions for relation-discovery would require including and combining with both solutions for TSEQs pattern exploration and interactive metadata analysis. Fourth, since the simplification of TSEQs through appropriate aggregation and data reduction methods makes the analysis more effective, domain experts must use the rich portfolio of machine learning methods to find meaningful simplification strategies. For TSEQs, this is beyond what related interactive machine learning solutions offer.

This work combines technique-driven (with general problems) and data-first visualization design study (with case-specific problems) methods to systematically investigate TSEQs. Through the in-depth study of problems remaining in related works, as well as six real-world cases and datasets, we systematically learned about data complexity and challenges with respect to underlying data characteristics. Through the observation of the domain experts, we abstracted 12 analysis tasks and 22 metrics for TSEQs, common across the six real-world cases. In an iterative and collaborative process, we designed and implemented IVESA, a visual analytics (VA) approach that can support the analysis of TSEQs of all six heterogeneous real-world cases and associated tasks. The VA tool consists of linked views, showing TSEQs from different perspectives, including an overview of up to 1,000,000 events and corresponding sequences, sorting and filtering interfaces, details-on-demand support, as well as visualizations of sequences in the context of metadata either as a large single or small-multiples visualization. In addition, we present views to support data analysis workflows that simplify TSEQs, either at the granularity supported through interactive clustering and filtering methods for sequences, or at the event granularity facilitated by motif detection and substitution. We validate our approach through three case studies and a user study, conducted again with the same domain experts. Results show that our approach supports experts across cases and application domains to analyze TSEQs in an interactive visual way. IVESA is generalizable with respect to the underlying datasets with heterogeneous characteristics, analysis tasks, and metrics that are common across cases and applications. Our evaluations laid the basis for reflections on the design process, a critical discussion of limitations, and the outline of future work.

2 Related Work

We review interactive visual analysis approaches for time-oriented data, separated by dataset type into time series and classical event sequences data. We discuss related VA approaches for data simplification with respect to grouping, filtering, and motif detection.

2.1 Interactive Visual Approaches for Time Series Data

We draw connections between TSEQs and time-oriented data with numerical values (time series), which can be univariate (such as a temperature measurement) or multivariate (such as the measurement of temperature, precipitation, humidity, etc.). Time series analysis typically focuses on the temporal change of values, which is fundamentally different to analyses conducted with TSEQs, where no value information exists. Consequently, most visualization techniques for time series explicitly encode these values, and are thus not directly usable for TSEQs. In fact, commonalities and inspiration can be found for the aspects of application domains, analysis tasks, and interaction techniques. Visual time series analysis approaches [4, 5] can be found in diverse application domains, e.g., in employee assessments [103], stock prices [50, 120], electronic signals [56], work behavior [57], geosciences [3], network device management [71], or anomaly detection in sensor networks [94]. Task taxonomies for time series are available in visualization [4, 5, 8] and time series data mining [38, 55, 74]. Some tasks, such as the identification of trends, are special and cannot be easily transferred to TSEQs. However, with a certain degree of abstraction, some user goals and tasks for time series analysis can also be adopted for the analysis of TSEQs [82]. Examples include preprocessing [13, 14, 21] (segmentation and alignment in the TSEQs case), the representation of data with features through descriptors [38, 55, 120] (through metrics in the TSEQs case), the discovery of patterns/motifs/subsequences [64], content-based similarity, search, and retrieval [76, 86], segmentation [12], prediction [74], exploratory analysis [118], and clustering [35, 62]. From these tasks, we support clustering based on features derived from metrics, as well as the discovery of motifs based on an interactive query-by-example [109] search strategy. A particularly relevant task for TSEQs is to relate [91] the data content with other attributes or metadata [16] or vice versa [15], sometimes also referred to as relation seeking [8, 100], i.e., the search for occurrences

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

of relations between data characteristics and references [4]. A groundbreaking relation-seeking example is the inspiring calendar view by van Wijk and van Selow [103] and its extensions [94], nicely demonstrating how patterns in the time-series content can be contextualized through auxiliary metadata attributes (such as days within a week or a year). Finally, we were inspired by interaction techniques for time series analysis, such as browsing [60, 63], zooming and/or panning [71, 117], dynamic queries [2], timebox-widgets [50], filtering [95], (feature) selection [30], query-by-example [11, 29], and focus+context [56].

2.2 Interactive Visual Approaches for Event Sequences

Classical event sequences include sequences of time-category pairs. We follow the survey by Guo, et al. [48], and structure the work by application domain, analysis task, and design space. Classical event sequences are strongly present in the medical domain, particularly in electronic health records (EHR). Here, prevalent analysis tasks include sequential pattern mining [47, 83, 96], clustering sequences based on similarity characteristics [43, 85], comparing individual event sequences [45, 51, 107], and comparing cohorts of patients [17, 69]. Approaches scalable to tens of thousands of events with hundreds of event categories [61] and techniques to reduce the data load using progressive analytics [96] were proposed. However, this does not translate well to TSEQs, as the data volume of TSEQs grows along the dimension of the number sequences and their length (up to millions of events), not through the cardinality of categories. In the domain of web data exploration, a substantial part of the related work is dealing with clickstreams. Clickstream data can be characterized by the high cardinality of the event set, long sequences, and multivariate events [66]. The abstracted analysis tasks often comprise pattern mining [65, 104], aggregation/clustering [105, 108], comparison of multivariate sequences [119], and detecting anomalies [113]. Other application domains include the analysis of user behavior using internet data [27, 79, 116], detecting anomalies in manufacturing [114], and using event sequences for predictive analysis and recommendations [32, 33, 46].

Classical event sequences in are often large-scale, high-dimensional, irregular, and heterogeneous, with diverse event types and multiple attributes. The presence of categories in the event sequences enables the utilization of color and branching in visual techniques based on Sankey diagrams [47, 51, 83] or matrix [33, 119] encodings. A stronger focus on the temporal information of events comes with timeline-based visualizations, such as EventFlow [73], LifeFlow [110], or LifeLines2 [106]. We compare to these approaches by introducing an overview for a higher-level summary of event sequences aligned by sentinel events. We also shared the challenge of visualizing thousands of events side-by-side while preserving the ability to identify motifs of interest. This challenge guided us to revisit pixel-based encodings [54] in combination with zooming and panning interaction, to enable users to explore larger datasets with up to 1,000,000 events in a Web-based environment.

2.3 Visual Analysis for Data Simplification

We review interactive data aggregation and simplification approaches along the three aspects of grouping, filtering, and motif simplification [34], as provided with IVESA. Grouping and filtering help to aggregate and reduce sequences, whereas motif detection and substitution reduce the number of events. IVESA allows users to interactively steer a deglomerative hierarchical clustering algorithm. According to the taxonomy by Elmqvist and Fekete [37], our visual representation uses points to show visual aggregates in a below-traversal rendering strategy and show averages about cluster characteristics on demand. Interactive drill-down and tree exploration is achieved by a global aggregation slider by default [18] (to keep the tree balanced), as opposed to local interactive split criteria [1, 9], or focus+context interactions [20]. Other interactive hierarchical clustering approaches distribute across the domains of genome analysis [36], human motion analysis [18], spatio-temporal data [44], biological processes [101], healthcare [26], topic evolution [31], and clickstreams [115]. Other types of clustering algorithms have been used for sequential data [62], including agglomerative hierarchical clustering [103], Markov chain models [25], self-organizing maps [108], KMeans [3], and DBSCAN [19], yet only some of them are user-steerable. Our interactive clustering solution is based on a user-defined feature set, by allowing interactive feature selection and refinement. Exploratory analysis of multidimensional data based on the rank-by-feature framework, taking advantage of interactive hierarchical clustering, was demonstrated in [90]. Similar to our approach, Wang et al. [105] explored iterative feature pruning to build similarity graphs for interpretable user clusters. With INFUSE [58], users can also apply interactive feature selection tasks, here to support prediction modeling. Finding features to characterize TSEQs can often be based on work in extracting temporal features from time-series data [78], on applying metrics [92], or both. However, our task to extract features through metrics was considerably impeded by the fact that most inspiring work for time series and classical event sequences takes the value information into account, which does not exist for TSEQs.

Further simplifications on the sequence granularity can be achieved by selection filtering, category filtering, time filtering, and attribute filtering, e.g., proposed by Monroe et al. in EventFlow [73]. Our feature filtering technique has its root in the rank-by-feature framework [90], in our case applied to TSEQs. The incorporated metadata filtering technique was inspired by faceted search and browsing techniques [11, 93, 111], whereas our clustering-based filtering approach is inspired by exploratory search approaches [18]. In contrast, domain experts did not offer strong demand for event-based filtering [112].

Simplifying the data on the event granularity has been addressed using a) temporal segmentation to split the sequence into sub-sequences [17], b) motif substitution based on event motif matching [49], c) substitution based on rules and regular expressions [28], or d) algorithmic recurring pattern detection [68]. Using motifs is powerful not only for reducing the size of sequences, but also for discovering higher-level patterns across motifs. Motifs reduce the visual clutter when focusing on remaining events, ultimately leading domain experts towards anomalies after most generalizable motif substitutions have been made. Our motif simplification strategy thus includes an interactive interface for the motif definition [24], matching [49], and substitution.

3 Abstractions

We describe our methodological process, characterize TSEQs, and outline the six real-world cases studied. Next, we describe abstractions of tasks and metrics and discuss routines for preprocessing.

3.1 Methodological Process

We identify a gap in existing literature studying the special characteristics of TSEQs, and how this type of data would be used by domain experts. To overcome the lack of methodological support and practical guidance for TSEQs, we adopted two complementary sources of information for the abstraction of analysis tasks and metrics.

First, we followed the technique-driven research principle, aiming at addressing general remaining problems [89]. The in-depth study of the dataset type at hand and corresponding related work revealed remaining challenges concerning the dataset complexity, its characteristics, and its usage. Reflecting on the related work, we identify the need for algorithmic support to compute features through metrics for TSEQs, to identify interesting patterns, and to simplify the data by reducing the number of sequences and events. Also, analytical support is needed to relate the content of TSEQs to contextual attributes and to facilitate pattern contextualization, insight generation, and decision-making.

As a second source for abstractions, we took inspiration from the data-first visualization design study principle [80]. This led us to first acquire real-world data, instead of a specific stakeholder. Our abstraction method differs from data-first design study principle in one crucial aspect: our main study subject is an abstract dataset type, instead of a concrete dataset instance. Based on the characterization of TSEQs (see Section 3.2), we made an informed selection of *six* real-world datasets. In five of the six cases, we were also able to identify one domain expert per dataset (see Section 3.3) from the start of the abstractions process. Only for the host behavior case, we initially re-used a dataset (and tasks) of a scientific publication [52], and conducted the expert cast and winnow stages [89] later.

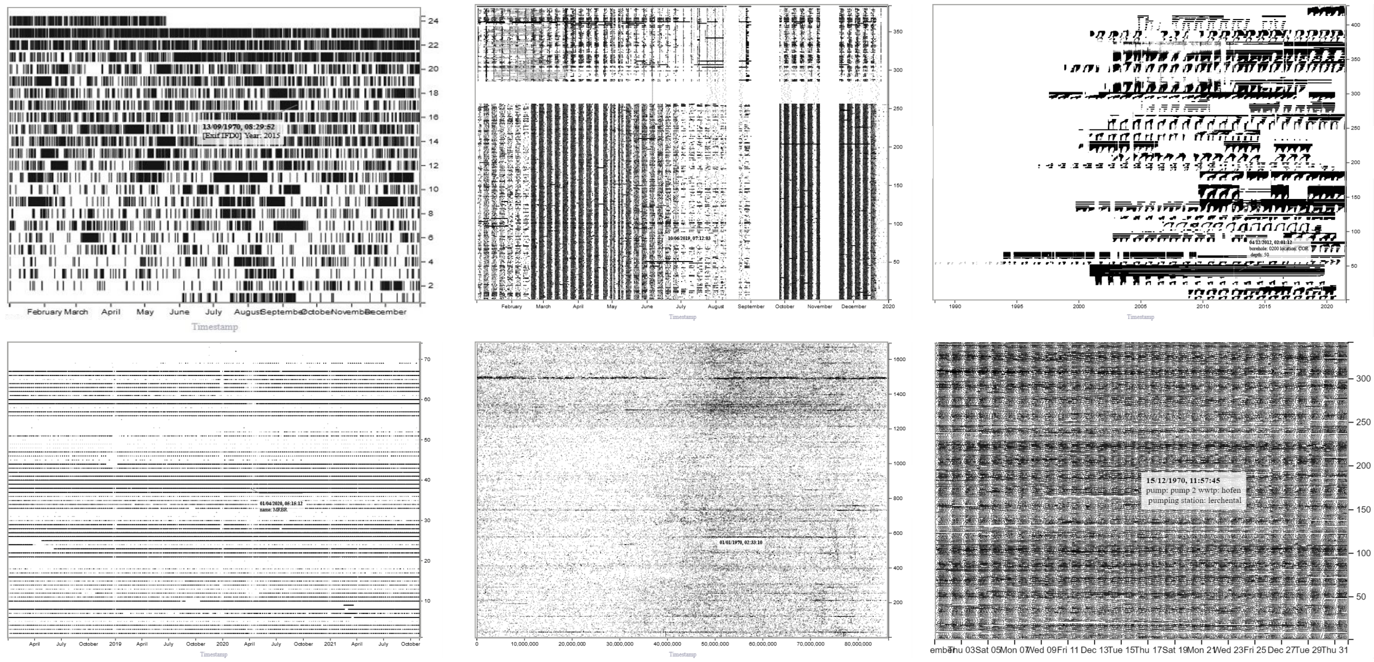

Fig. 2. IVESA's overview visualization for the six datasets and cases used throughout this data-first design study, representing six different application domains and involved experts (pseudonyms) with data of considerable heterogeneity. From top left to bottom right, the cases we studied are 1) Digital Photography (Nathan), 2) Host Behavior in a Computer Network (Andreas), 3) Permafrost Observations (Esmeralda), 4) Radiological Examinations (Reto), 5) Stock Market Tweets (Wenuka), and 6) Waste Water Treatment Activity (Olivia). The time-stamped event sequences span across the horizontal time axis, sorting by different metrics eases the visual comparison of hundreds or possibly thousands of sequences with up to 1,000,000 events.

To design a data-centric but domain-agnostic visualization tool that helps to solve problems encountered by experts across different domains, we engaged with the experts early in the process to learn more about their domain context, usage of the data, current tool usage, problems, and challenges, to identify design targets and determine appropriate validation strategies [77]. The involvement of the six domain experts in the abstraction process of both tasks and metrics helped us to understand priority levels and relevance across cases, and to develop a more holistic understanding of tasks and metrics for TSEQs. The results of these two parallel processes are presented in Sections 3.4 (tasks) and 3.5 (metrics), with a rich set of figures and tables, additionally provided with the supplemental materials for both tasks and metrics. During the design and development of IVESA, we kept the experts of the six datasets/cases in the loop, to gather feedback for further refinements. In addition, for more general problems of dealing with TSEQs, we reached out to visualization experts, to receive informed feedback from an informed outside perspective. In the validation phase of IVESA, we conducted a user study with all six datasets (again with the experts per dataset), to receive a broad spectrum of usage feedback, on task success, tool usage, and conducted workflows, as a basis for reflection, discussion, and future work (see Section 6). Three instances of these think-aloud observational runs are also presented as case studies in Section 5, to demonstrate IVESA's usefulness.

3.2 Time-Stamped Event Sequences

TSEQs contain events with time information only, without value information; a fresh direction in the analysis of time-oriented data [4, 5]. The absence of values per timestamp is the main difference compared to time-oriented (time-dependent) data such as time series (numerical values) and classical event sequences (categorical values). The insights users gain from TSEQs are derived only from the event occurrences within sequences, i.e., the temporal characteristics, event behaviors and motif patterns are at the center of the analytical focus. Conversely, when dealing with other types of time-oriented data, the focus is placed on analyzing data values and the changes in these values over time.

TSEQs can be quite long and contain hundreds or thousands of events, which is why segmentation can be useful to separate complex TSEQs into more manageable units and make the sequences comparable. Due to the large number of TSEQs, a central question is how to rank them, according to what users deem interesting. To support this process, metrics for TSEQs exist, allowing the assessment of TSEQs by their regularity, speed, rate of acceleration/deceleration, etc. Users typically relate TSEQs to *external attributes* (metadata) such as location, cohort, or other environmental parameters. While TSEQs form the data *content*, metadata attributes form the application *context*, which is useful to explain patterns found in the sequences and to make sense of them. We limit the scope of external attributes to mixed sets of numerical, ordinal, categorical, or binary types, syntactically represented by string, integer, boolean, or float primitives. Due to their usefulness across applications and cases, we support temporal metadata, e.g., directly derived from the temporal information of the sequences. Examples include year, month, day of the week, or hour of the day. Finally, TSEQs may relate to multiple external attributes with values changing over time. Naturally, these time-varying attributes relate to the events of a TSEQs, and may even form dependencies.

3.3 Six Real-World Datasets and Cases

In our work, we studied six datasets from six application domains (see overview in Figure 2) with considerable heterogeneity in event semantics, number of sequences, and number of events per sequence: **1)** Digital Photography, **2)** Host Behavior in a Computer Network, **3)** Permafrost Observations, **4)** Radiological Examinations, **5)** Stock Market Tweets, and **6)** Waste Water Treatment Activity. We always refer to the expert involved in a case with a pseudonym, for anonymity reasons. Data characterization, task abstraction, user goals, and screenshots of each case are available in the Supplemental Material.

3.3.1 Digital Photography

Nathan is a photographer interested in analyzing his personal photography collection of 144,436 images taken with various devices from 1999 to 2022. Each captured image is described by a timestamp representing

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

an event, and each year is considered to be a TSEQ, resulting in a total of 24 sequences. Events additionally contain metadata such as the capture device, image orientation, and Exif attributes.

3.3.2 Host Behavior in a Computer Network

In the Host Behavior case, we analyzed a publicly available dataset of network flows collected by Jirsik and Velan in their 2021 host behavior study [52]. The dataset on the communications flow of a university network consists of 348,574 time-stamped ingress (incoming) and egress (outgoing) communication events. Every event reflects the activity of a workstation within an hourly interval, which is what Andreas, a cybersecurity expert, is most interested in. A workstation is considered to be a TSEQ, with a total of 384 workstations. The data was gathered between January and December 2019, with a total of 5,064 hours per workstation. Most relevant metadata attributes are the workstation names, communication flow (ingress or egress), and their device category in the network (administration, international services, students). In the evaluation process, we involved Andreas, a researcher with six years of experience in the field of cybersecurity threats.

3.3.3 Permafrost Observations

Esmeralda is an Earth observation researcher interested in analyzing the permafrost in Switzerland. The dataset was obtained from the PERMOS network [98] and is publicly available. The measurements originate from 613 temperature sensors placed in 29 boreholes with multiple measurement depths per borehole, situated in different permafrost regions of Switzerland. Depending on their varying installation dates, the longest TSEQ are spanning over 30 years of daily measurements. Following the recommendation from Esmeralda, the dataset was reduced to days (events) where melting occurred (temperature above zero), producing a total of 744,633 events. Events contain metadata about borehole location, depth, and temperature.

3.3.4 Radiological Examinations

Reto is a radiologist interested in analyzing radiological equipment operation records of a Swiss hospital. The dataset consists of examination timestamps (events) collected over 46 months across 74 devices and includes 538,763 of individual examinations. The device occupancy is given in 15-minute intervals, and each device represents one TSEQ, leading to 74 TSEQs. Events contain metadata about the device and examination type.

3.3.5 Stock Market Tweets

Wenuka, a data scientist with a passion for the stock market and plenty of industry experience was the subject of the first case study. His dataset consists of 120,093 timestamped tweets (events) including #apple or #csco, each with more than 100 likes, between January 1, 2017, and August 20, 2021. One day of tweets was considered an TSEQ, to support between-day comparison, i.e., we segmented by days. The segmentation resulted in 3,158 TSEQs, which we aligned globally. The metadata included the hashtag of tweets for the differentiation of stocks and information about temporal seasonality. Before the session, the dataset was extracted from the Twitter API [1], preprocessed, and segmented into days according to UTC.

3.3.6 Waste Water Treatment Activity

Olivia, the operations manager at a wastewater treatment plant (WWTP) with 25 years of experience, was the frontline analyst of the second case study. Olivia's goal was to analyze events of three pumps, to identify expected and unexpected behavior, as well as compare pumps by their event characteristics. The dataset with the three pumps consists of 1,101,703 events, each representing minute-wise sensor readings showing when a pump is active. The data was collected from 2012 to 2021, and we chose a monthly segmentation with a left alignment, to make the analysis more manageable. The wastewater level in cm and discharge in liters/second are the temporal metadata attributes. The pump name, month, and year are the available static metadata.

[1]https://developer.twitter.com/en/products/twitter-api

3.4 Task Abstraction and Higher-Level Categorization

For the task abstraction, we used the typology of analysis tasks proposed by Peiris et al. [82]. The typology guided our abstraction process by offering a long list of 23 tasks at a comparable abstraction level, that the authors derived from 65 interviews with non-experts and 16 design studies related to TSEQs. From this encompassing task list, our technique-driven data-first abstraction method finally revealed 12 abstract tasks to support the analysis of TSEQs across six real-world cases. Two tasks relate to upstream data processing ($\mathbf{T}_{A,S}$), six tasks primarily focus on insight generation and decision-making ($\mathbf{T}_{1\text{-}6}$), whereas the four remaining tasks allow data-simplification ($\mathbf{T}_{7\text{-}10}$). This categorization is in line with Brehmer and Munzner's typology of abstract visualization tasks [23] and separates tasks into *consume* ($\mathbf{T}_{1\text{-}6}$) and *produce* ($\mathbf{T}_{7\text{-}10}$) categories. This categorization was useful in three ways: to get a better understanding of how to support users in their goals, to design the VA tool appropriately, and to validate the approach. This high-level perspective on tasks also forms the principal workflow that the IVESA approach supports, shown in Figure 4, and reflects observations made about the experts in the six real-world cases.

In Figure 3, we present task abstraction results using the target-criteria crosscut method [82]. With the triples consisting of actions, targets, and criteria, we arrived at a fine-grained level of task description, to be leveraged in the iterative design phase of IVESA. Examples of tasks include *rank event sequences by feature*, *group event sequences by metadata attribute,* or *group event sequences by features*.

$\mathbf{T}_A$ **Align**: The alignment of TSEQs eases the sequence comparison. Dominating criteria are based on the granularity of events and associated timestamps or the start/end of (sub-) sequences.

$\mathbf{T}_S$ **Segment**: Many domain experts prefer segmenting extra-long TSEQs into sub-sequences, prior to their in-depth analysis. Naturally, the only data target of segmentation actions is TSEQs, whereas dominating criteria are either patterns in TSEQs, or seasonal phenomena represented through event criteria or temporal metadata.

$\mathbf{T}_1$ **Overview**: Users need an overview of the underlying data collection. Especially if parts of the TSEQs are undiscovered, understanding principal structural characteristics of both *TSEQs* and *events* is crucial. Another target for overview actions is associated *metadata* attributes.

$\mathbf{T}_2$ **Sort/Rank**:Changing their order enables users to analyze targeted TSEQs from different perspectives. Criteria for sorting TSEQs include features, metadata, and clusters of TSEQs.

$\mathbf{T}_3$ **Compare**: The comparison of individual TSEQs enables users to identify commonalities and differences, including fine-grained events, metadata attributes, and features.

$\mathbf{T}_4$ **Details**: Users need support for detailed analysis of individual TSEQs and included events. Relevant criteria include events, metadata, and temporal features.

$\mathbf{T}_5$ **Relate**: The dominant task is to relate a subset of TSEQs with metadata attributes, aiming at identifying interesting means to contextualize findings. Special cases that we identified are the need for relating single TSEQs clusters to metadata or feature criteria.

$\mathbf{T}_6$ **Detect Outlier/Anomaly**: Many experts require support for the identification of outliers or anomalies. The predominant target of outlier/anomaly detection is for TSEQs, often assessed through fine-grained event-based or sequence-based criteria.

$\mathbf{T}_7$ **Analyze Features**: Features are the entry point to algorithmic support, such as clustering, dimensionality reduction, or motif detection methods. Feature sets are required at both the coarse TSEQs granularity and the fine-grained event granularity. Interested users also need a means to assess, manipulate, and (un-)select features. Criteria for metrics to reveal features are based on events, TSEQs, or metadata.

$\mathbf{T}_8$ **Group**: Assigning TSEQs to groups is useful to reduce the number of items at the sequence granularity, from multiples to a few groups. Criteria to facilitate this aggregation can be based on content-based clustering or metadata-based data partitioning.

$\mathbf{T}_9$ **Filter**: Filtering enables users to reduce the number of TSEQs from many to a few most relevant. Relevant criteria may be filtering by clusters, filtering by features, or by individual metadata attributes.

$\mathbf{T}_{10}$ **Substitute Events Motif**: With motifs, we refer to TSEQs subsequence "patterns" of interest. Motifs may be revealed by algorithmic

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

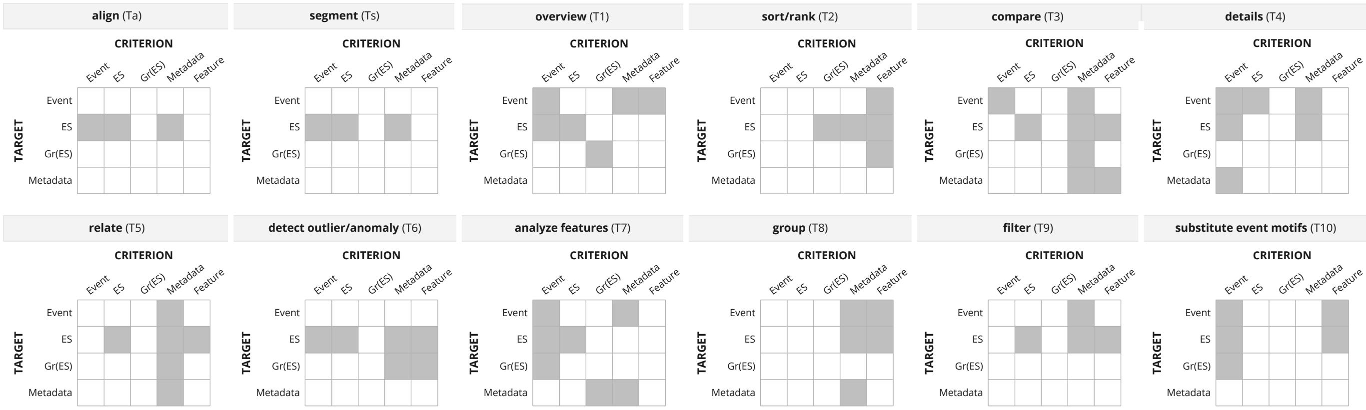

Fig. 3. Task abstraction using the target-criteria crosscut method [82]. Overall, the domain experts are mainly interested in 12 task actions, applied to different data targets and based on different data criteria. Types of targets and criteria are Events, Event Sequences (ES), groups of event sequences Gr(ES), Metadata, and Features. Example: experts were interested in segmentation (T_S) actions of event sequences (targets), based on different criteria: events, event sequences, and metadata. The target-criteria crosscut served as a fine-grained recipe for design targets.

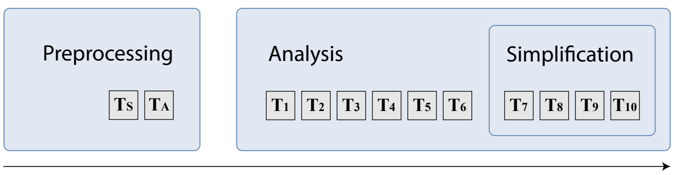

Fig. 4. Higher-level task categorization and workflow, supported with the IVESA approach, reflecting all six real-world cases studied.

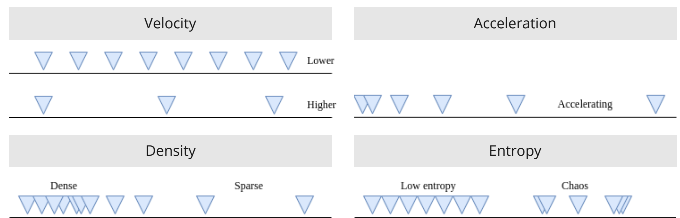

Fig. 5. Examples of temporal features that can be identified through the use of metrics. The space of characteristics includes velocity, acceleration (and slowdown), density, or entropy [92].

support, or be identified by users directly. Motifs form the basis for event-based simplification: when a motif of events is identified multiple times, it can be simplified through substitution, e.g., by a symbol.

3.5 Metrics

Based on related work, the observation of the six experts, and meetings and interviews, we developed 22 metrics for TSEQs. With metrics, users can characterize TSEQs, leading to a compact set of features ($\mathbf{T}_7$) that can faithfully represent the data. With IVESA, users can visualize, select, normalize, sort, rank, and filter by these features. In general, we differentiate between two types of metrics: Event-based metrics compute exactly one numeric feature value *per event*, i.e., they have changing values over time (temporal/dynamic features). In contrast, TSEQs-based metrics compute a single numeric feature value *per TSEQ*, i.e., the features are global/static.

Expert interviews emphasized the relevance of a series of metrics and how vital they are to achieving their goals. For example, *"changes in TSEQ frequency"*, *"analyze regularity to detect anomalies"* are some important metrics observed. Experts also pointed out gaps, outliers, periodicity, subsequence length, and dense regions as important. The awareness of their relevance also helped us to prioritize metrics identified in the literature: Related works come from the domain of statistical metrics for time-series [78], where we identified a subset of metrics, applicable for TSEQs. Summary metrics such as the number of events or minimum, maximum, and mean length of sequences form a source for features at the TSEQs granularity. Finally, time-based metrics such as duration between events, regularity, or slowdowns [70, 92] can be applied to TSEQs. Examples further include peak detection [39], outlier score within a sequence [99], density [59] or similarity metrics (between two or more sequences) [42]. We present four of the metrics at the event-based granularity in Figure 5, a complete presentation of all metrics at both granularities (event-based and TSEQs) is presented in the supplemental material in detail. This overview includes tables with metric names, formal descriptions, as well as visual examples, like the examples shown in Figure 5. Overall, our portfolio of metrics in IVESA contains 8 temporal and 14 static features.

Event-Based Metrics: all temporal metrics put a single event at the center of computation. Our strategies to compute features at this fine-grained granularity are mainly based on three approaches. Differential metrics at the event granularity use a focused timestamp and direct neighbors, i.e., the gap size to neighboring events. The velocity (the average of the gap size before and after the event) is one important metric and yet forms the basis for several more complex metrics, such as the acceleration (change of the velocity). An alternative type of metrics is based on the sliding window principle, with examples like density or entropy. Finally, we exploit date-time characteristics for some metrics, such as year, quarter, month, week, weekday, hour, minute, and second of the event.

TSEQs-based Metrics: two strategies for the computation of static features exist. Some metrics directly use all the time information of all events, and compute statistical aggregates to arrive at the coarse TSEQs granularity. Examples include statistical metrics, such as the mean and variance of the timestamps. The second type of these metrics uses already calculated event-based features to calculate statistical aggregations. Examples include skewness, entropy, or regularity period.

3.6 Preprocessing

Segmentation ($\mathbf{T}_S$): In all six cases, domain experts required exactly one persistent *segmentation* criterion, often dictated by the natural periodicity or seasonality of the data, such as years when observing and comparing snow melting events. In general, segmentation criteria can be a) semantics-driven, such as natural periodicity like the solar-dependent intraday awareness, b) domain-specific requirements expressed by experts, or c) based on data characteristics such as gaps. In the six cases, we did the segmentation of TSEQs ($\mathbf{T}_S$) as a pre-process, based on the observation that we could determine dominating segmentation criteria already in the data abstraction and design phase. A striking

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

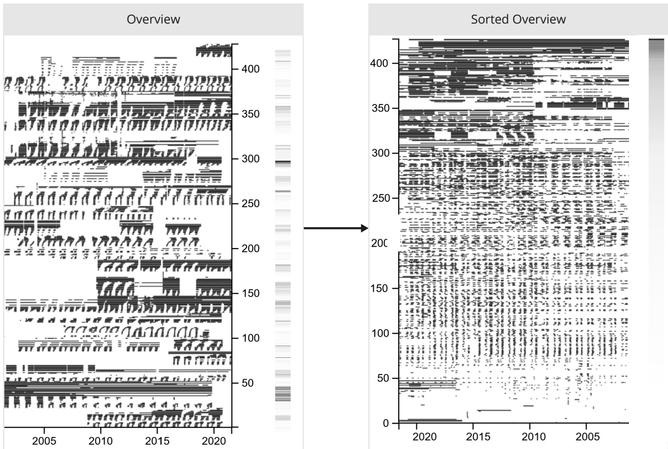

Fig. 6. Effect of a sorting action applied by the user. On the left, TSEQs were ordered by the device type, on the right the user changed the order to a content-based criterion: the longest common subsequence feature. All user-selected sorting criteria are always shown as columns on the right of the TSEQs display, obviously the values of the sorted criterion perfectly correlate with the new TSEQs order.

Fig. 7. Small-multiples support of the *Metadata View* as an alternative to large singles. A user control allows the selection of a partitioning criterion. The calibration of the criterion leads to the partition recipe into (small) multiples. In the example, a user leverages the small-multiples technique to compare the events of four boreholes, by yearly and weekly criteria.

benefit of segmenting the data in a pre-process is a significant increase in computational scalability at runtime, as many pre-computation and database management steps could already be scheduled upstream.

Alignment ($\mathbf{T}_A$): Similarly, we identified that the dominating sequence *alignment* ($\mathbf{T}_A$) strategies can often be determined in the design phase. Based on the observed real-world cases, three sequence alignment strategies [84] are necessary to support comparison.

- *Left alignment* aligns every TSEQ with respect to its first event, to align for a common starting point, e.g., for yearly segments.
- *Absolute time alignment* defines a global time interval in absolute time, and arranges every TSEQ on that interval, respectively.
- *Relative alignment* scales each TSEQ so that all TSEQ have the same relative length, allowing their comparison by event motifs.

In summary, we observed that segmentations of extra-long sequences into more manageable sub-sequences in combination with a meaningful alignment strategy often helped to ease visual comparisons considerably.

4 The IVESA Visual Analytics Approach

We present IVESA, a VA approach for interactive and exploratory analysis of TSEQs. We describe the views, visual encodings, interaction techniques, and algorithmic support, based on the abstractions made.

4.1 System Overview

Figure 1 shows an overview of IVESA, structured into four regions: the *Sequence Overview* (top left), the *Metadata View* (top right), the *Details View* (bottom left), and the auxiliary *Summary View* (bottom right). Our design rationale was to have visual support for all *consume tasks* ($\mathbf{T}_{1\text{-}6}$, see Section 3.4) always visible and active in the first three regions. In turn, with the fourth region, IVESA supports produce tasks ($\mathbf{T}_{7\text{-}10}$) with auxiliary views that pop up on demand through the *Summary View*. These on-demand views include the extended display of features in the *Details View* ($\mathbf{T}_7$), the *Clustering View* ($\mathbf{T}_8$), the *Filtering View* ($\mathbf{T}_9$), and the *Motif View* ($\mathbf{T}_{10}$).

In general, IVESA implements linking techniques where appropriate. Brushing leads to selections of TSEQs, which are always highlighted across views, with blue being the global selection color. Also, interactive clustering results are propagated across views, using categorical colors. Finally, if users have simplified TSEQs with motif shapes, these motifs are visible through a shape-based linking strategy.

4.2 Sequence Overview

The *Sequence Overview* in Figure 1 shows an overview of all events and sequences of the TSEQs dataset. Design targets have been to enable users overviewing structural characteristics of TSEQs ($\mathbf{T}_1$), sort TSEQs according to metrics and features of interest ($\mathbf{T}_2$), relate TSEQs with metadata ($\mathbf{T}_5$), identify motifs ($\mathbf{T}_{10}$), and detect outliers and anomalies ($\mathbf{T}_6$). A design requirement was to scale for up to 1,000 sequences, each with more than 1,000 events, i.e., to scale for at least 1,000,000 atomic events in a web-based display. In the example in Figure 1, the *Sequence Overview* shows daily measurements of snow melting events in 29 permafrost regions, measured with 613 sensors for up to 30 years, a scenario that spans a resolution of 6,700,000 possible melting events.

The *Sequence Overview* utilizes the strength of the double-position encodings of scatter plots with thin black vertical line marks encoding individual events. Events of all TSEQs are positioned horizontally, according to a global unifying time axis. This time axis is the result of a user-defined segmentation ($\mathbf{T}_S$) and alignment ($\mathbf{T}_A$) strategy, to allow meaningful visual comparisons (see Section 3.6). All TSEQs are listed along the vertical display axis, the (rank) order automatically responds to users' sort actions ($\mathbf{T}_2$) on features, clustering results, or metadata. In the *Sequence Overview*, we further encode motifs, using horizontal line marks, the algorithmic support for motif computation is described in Section 4.5.2. A color coding helps users distinguish different motifs, due to the categorical nature of the hue channel. Additionally, we allow semantic zooming and panning interactions for closer inspection, as well as the selection of interesting TSEQ that can subsequently be analyzed in the *Details View* for further comparison, described in Section 4.4. Finally, tooltips provide details on events, TSEQs, and associated metadata.

Users can sort/rank ($\mathbf{T}_2$) all TSEQs, using the sorting criteria on the right of the *Sequence Overview*, as depicted in Figure 1. Clicking on one of the criteria automatically executes the sort operation, and positions of TSEQs change, accordingly. Figure 6 demonstrates the effectiveness of a simple sort operation, to support different information needs of users. In the *Sequence Overview*, each sorting criterion is displayed as a column with grayscale luminance for numerical values and hue colors for categorical values. Criteria can be based on features, clustering results, or global (non-temporal) metadata. With the encodings of sorting criteria, users can easily relate ($\mathbf{T}_5$) sorted TSEQs with the different sorting criteria displayed, e.g., for value correlations. In IVESA, users are in control of the set of displayed sort/rank criteria: by using the edit icon on the lower right of the *Sequence Overview*, a config menu pops up and allows adding/removing sorting criteria. This design decision was made in favor of simplicity, to only show the small user-specified subset of criteria, from possibly hundreds of provided features and metadata.

4.3 Metadata View

The *Metadata View* in Figure 1 (upper right) provides an overview ($\mathbf{T}_1$) of events (target), structured by metadata, features, and clusters (criteria). The view is designed in a way that users are in full control of the event-grouping strategy ($\mathbf{T}_8$), as a means to compare ($\mathbf{T}_3$) resulting event distributions. The *Metadata View* also helps to relate content-based findings to supplemental metadata, features, or clusters ($\mathbf{T}_5$). Finally, the view is useful to analyze features ($\mathbf{T}_7$), or to identify temporal trends, outliers, and anomalies ($\mathbf{T}_6$).

The *Metadata View* uses the heatmap idiom and allows users to systematically cross-cut event distributions by two criteria from a pool of metadata attributes, features, or clustering results. Every cell of the 2D matrix represents a particular number of events, encoded with the saturation channel of a dark blue basic color. Controls at the top enable the selection of criteria, leading to the re-creation of the heatmap. The discrete nature of the 2D matrix representation uses binned value distributions of criteria, for the unified analysis of both numerical and categorical attributes. With IVESA, users can apply three binning strategies:

- **Discrete Numerical**: to account for discrete values such as integer attributes. We apply a domain-preserving [75] binning approach to keep the span of every bin equal if possible. Years between 2008 and 2016 with 2 bins would be binned to 2008-2012 and 2013-2016.
- **Continuous Numerical**: The default binning strategy for float values is also domain-preserving [75], meaning that the span of every bin has the very same range. Binning of the floats 0.7, 0.998, 2.1 into 2 bins: (0.7, 1.4], (1.4, 2.1]
- **Categorical**: we use the individual levels of categorical attributes. If the number of levels exceeds a user-definable threshold (default: 20), we focus on the n-1 most frequent levels and add one more bin for all remaining levels. Binning of the strings A, B, C, D (sorted by cardinality) into 3 bins would be A, B, and "other: 2"

Users can interactively select criteria, and adjust the binning strategy, accordingly. By clicking the binning button on the upper right, a popup menu provides all necessary controls to steer and refine the binning process, including the number of bins. Our justification of the default number of bins (where appropriate) follows Sahann et al. [87], who studied the trade-off between the bin count in histograms and the user error rate. The authors found out that more than 20 bins do not contribute to a considerable improvement in perception anymore.

To better support the exploratory nature of experts' information needs, we gained inspiration from large singles and small multiples [102] concepts. As an alternative to the large-single display, users can also create small-multiples arrangements, by interactively defining a partition criterion ($\mathbf{T}_8$). All events are partitioned by this criterion, leading to a small-multiples display of heatmaps. The popup menu allows users to create small-multiple arrangements based on a) selecting a metadata attribute (see Figure 7), b) defining ranges of relevant feature values, c) using the currently selected TSEQs (supplemental materials), or created clustering results (see Figure 17). According to the standards for displaying juxtaposed information [41], we arrange small multiples panels in a grid display side-by-side to ease the comparison ($\mathbf{T}_3$) of partitioned ($\mathbf{T}_8$) data.

4.4 Details View

The *Details View* on the lower left of the IVESA interface (see Figure 1) visualizes selected TSEQs, events, temporal features, and motifs. This visualization allows users to compare ($\mathbf{T}_3$) and analyze details ($\mathbf{T}_4$) of selected TSEQs. Tooltips allow the contextualization of TSEQs by showing metadata on demand ($\mathbf{T}_5$). Ultimately, the *Details View* is designed to let users identify interesting event patterns and thus forms the basis for motif identification and substitution ($\mathbf{T}_{10}$).

To start with the most atomic visual element, every event timestamp is encoded with a vertical line (blue, according to the global selection color). Tool-tipping helps users to relate static and temporal metadata attributes to hovered events ($\mathbf{T}_5$). As for the *Sequence Overview*, the x-axis shows the temporal progression of TSEQs. The *Details View* consists of a global static timeline component at the bottom and a component with a scalable timeline at the center, showing all displayed TSEQs in large detail. The two timelines are coupled through brushing and linking. The global timeline also allows users to zoom into a temporal subset of events by brushing over the desired events, as demonstrated in Figure 8. The scalable timeline at the center automatically responds to the brushing event to always correspond to the brushed time interval. Both components of the *Details View* always show the currently selected set of TSEQs, with considerably more visual (y-axis) resolution reserved for the scalable timeline component.

To further support the comparison of TSEQs ($\mathbf{T}_3$), the *Details View* allows the visualization and analysis of temporal features ($\mathbf{T}_7$). We use line charts to show user-selected features of interest. As depicted in Figure 8, a combo box allows users to switch between features, here using an outlier metric as an example. In addition, users have a control to apply feature normalization to support identifying patterns within sequence and comparing across event sequences. Temporal features can be normalized sequence-wise or with respect to the whole dataset. To account for the variety of comparison scenarios, we provide five normalization techniques: L1-norm, L2-norm, max-norm, logmax, and z-score, adapted from scikit-learn [81].

Another interactive functionality of IVESA is partitioning the events of selected TSEQs into small-multiples arrangement, to provide another perspective that eases the effective TSEQs comparison ($\mathbf{T}_3$). This functionality is often triggered by experts right after the detailed temporal analysis of selected TSEQs in the *Details View* and thus builds a natural transition in the users' workflow from the *Details View* to the *Metadata View*. Finally, the *Details View* is useful to identify and define motifs, as Section 4.5.2 will describe.

4.5 Summary View

The *Summary View* on the lower right of the IVESA interface (see Figure 1) serves two main purposes. First, it summarizes all data simplification results achieved through clustering ($\mathbf{T}_8$), filtering ($\mathbf{T}_9$), motif definition ($\mathbf{T}_{10}$). Second, it defines the entry point for users aiming at simplifying the data further. In the following, we introduce the three data simplification processes in detail, enabled through the *Filtering View* in Section 4.5.1, the *Motif View* in Section 4.5.2, and the *Clustering View* in Section 4.5.3.

4.5.1 Filtering View

Users can reduce the number of TSEQs by applying filters ($\mathbf{T}_9$), based on event sequence and metadata criteria. Using the overview of active filters in the *Summary View*, users can make adaptions by opening the *Filtering View*, shown in Figure 9. The *Filtering View* is composed of three functionalities: a) selection of a criterion, b) specifying the filter, and c) assessing the effect of the filter on the dataset population even before filter application.

For the criteria selection (a), users can select from a straightforward combo box control. The filter specification (b) differs for numerical and categorical data. For numerical attributes, a range slider control allows users to specify a range of values, in the notion of a dynamic query [2]. Users can also filter by categorical metadata using an auto-complete drop-down list. The assessment of the filter effect (c) is realized through a horizontal bar chart, showing the ratio of data included in the analysis before and after the application of a filter. The x-axis legend also encodes the absolute number of TSEQs, to assess relative and absolute change. All views respond to changes of the filter state: Figure 9 demonstrates the effect of a filter on the *Sequence Overview*.

4.5.2 Motif View

The selection and detailed analysis of individual TSEQs allows users to identify recurring event patterns. We refer to patterns, i.e., non-randomly occurring distributions of consecutive events, as motifs. Motifs form a valuable source for event-based data simplification. Here, the line of approach of IVESA is to substitute the multitude of events forming a pattern by a (visual) motif representation. To define a motif, users can brush events in the scalable timeline in the *Details View*, as demonstrated in Figure 10. A click on the "add motif" button (on

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

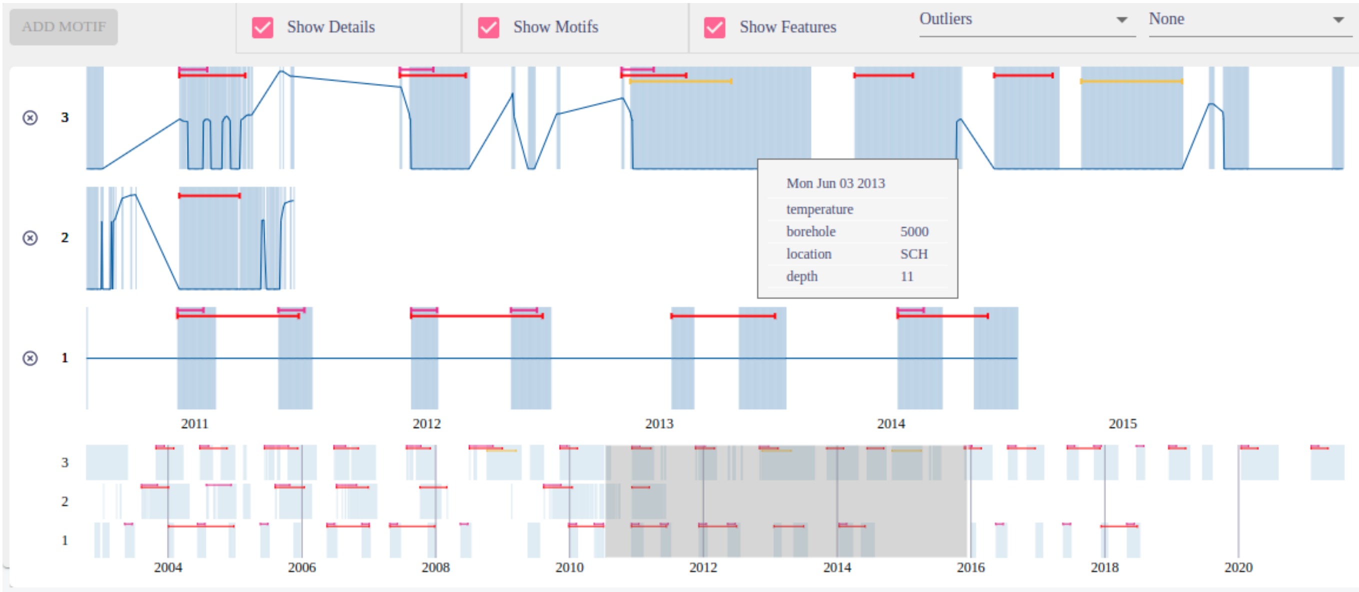

Fig. 8. *Details View* of IVESA, showing three selected TSEQs. Any TSEQ of interest can be selected from the *Overview* and analyzed in more detail. A global timeline at the bottom enables the temporal drill-down. Events are shown as vertical lines. Users can also select a temporal feature (line chart) for in-depth analysis: in this case, the Outliers feature is active, with high values coinciding gaps between events occur. Furthermore, the *Details View* is capable of showing motifs: in the example, a user has already defined three motifs (represented pink, red, and yellow line segments), that recur with some regularity in the analyzed TSEQ.

the upper left) triggers the *Motif View* to pop up. As an alternative, users can modify existing motifs by entering the *Motif View* through the *Summary View*.

The *Motif View* then allows users to specify the motif search and to execute the motif query ($\mathbf{T}_{10}$), Figure 10 shows an example. The view mainly consists of four functionalities: a) the visualization of the event pattern of the motif, b) the motif parameter configuration, c) guidance on the choice of motif-match parameters, and d) the coverage of motifs with respect to the underlying dataset.

The event signature (a) shows the distribution of events within the user-selected motif pattern, with visual encodings as in the *Details View*. IVESA provides configuration elements for all three parameter types of the motion retrieval algorithm (b). First, to account for different notions of similarity in the retrieval process, users can switch between two distance/similarity measures: Euclidean distance and dynamic time warping (implementation by Meert et al. [72]). Second, a parameter to steer the kernel of the retrieval algorithm, i.e., the type of sliding windowing strategy applied to retrieve motifs. The index-based variant preserves the number of events and is flexible in temporal comparison. In turn, with the time-based kernel the time interval of the query is fixed, possibly leading to results of different event counts. The time-based window works well with dynamic time warping. The Euclidean distance, however, is a bin-to-bin comparison method and thus can only be applied to the index-based variant [16]. In contrast, dynamic time warping finds a warping path even for uneven numbers of events for two given subsequences. Third, to control the number of search results, users can adjust a similarity threshold, i.e., a maximum distance that determines if a retrieved motif should be a part of the retrieved result set or not. Possible values are between 0.0 and 1.0 – 1.0 would be an exact match between motif query and retrieved event sequence. To guide users in their choices (c), the *Motif View* shows a histogram that puts the threshold in relation to the number of motifs that would be retrieved. Figure 10) shows an example where a considerable number of patterns are retrieved if the threshold is set lower than 0.95. At the bottom of the *Motif View*, the coverage of all motifs (d) is shown. A horizontal bar chart shows the number of sequences found for each motif that has been defined by users so far; four in the example shown in Figure 10. The numerical axis of the histogram shows a percentage scale, representing the proportion of events already assigned to motifs. A categorical colormap allows linking of defined motifs across views and eases their visual comparison ($\mathbf{T}_3$). Once a motif search configuration has been applied, the motif is added to the motif set. The data simplification through motifs is automatically shown in the *Sequence Overview* and the *Details View* (see Figure 1). The motif shape includes a line mark with two ticks indicating the start and the end of each motif occurrence.

4.5.3 Clustering View

The main purposes of the *Clustering View* are to enable the grouping of TSEQs ($\mathbf{T}_8$) through feature-based clustering, overviewing TSEQs ($\mathbf{T}_1$) by providing another perspective TSEQs through features, supporting the analysis and calibration of features ($\mathbf{T}_7$), and filtering TSEQs ($\mathbf{T}_9$) by content-based criteria. The clusters are formed using a hierarchical clustering algorithm, for two reasons: First, cluster hierarchies can intuitively be displayed in dendrograms, with the positive side effect of showing similarity relations across clusters and subtrees. Second, cluster hierarchies form a solid basis for interactive level-of-detail operations. Figure 11 illustrates a typical clustering workflow. The interactive hierarchical clustering is based on non-temporal TSEQ features. A dendrogram visualization shows resulting clustering trees, where each point mark encodes a subset of TSEQs in the individual branches. Categorical colors discriminate between clusters and ease linking and comparison across views. The size of clusters in subtrees is encoded with the size of corresponding point marks. Users can interactively adjust the aggregation level of the clustering [18] to control the level of detail, i.e., the number of clusters. As a result, domain experts are always provided with a clustering result at a user-steerable aggregation level. Here, we follow the "below traversal" rendering strategy, as described in the design guidelines for hierarchical aggregations by Elmqvist and Fekete [37]. Interactive changes to the clustering are automatically represented in the dendrogram and in linked views showing clustering information, as the example in Figure 11 shows.

Interactive hierarchical clustering comes with the downside of scalability issues, which we address in three ways. First, we pre-compute clustering results whenever possible; a database API helps to persist computed results across run times. Second, we use a dimensionality reduction method as a pre-process to project the data into 2D, leading to a significant reduction in compute time (by default: the linear PCA method [53]). Third, in line with $\mathbf{T}_7$, we enable users to interactively

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

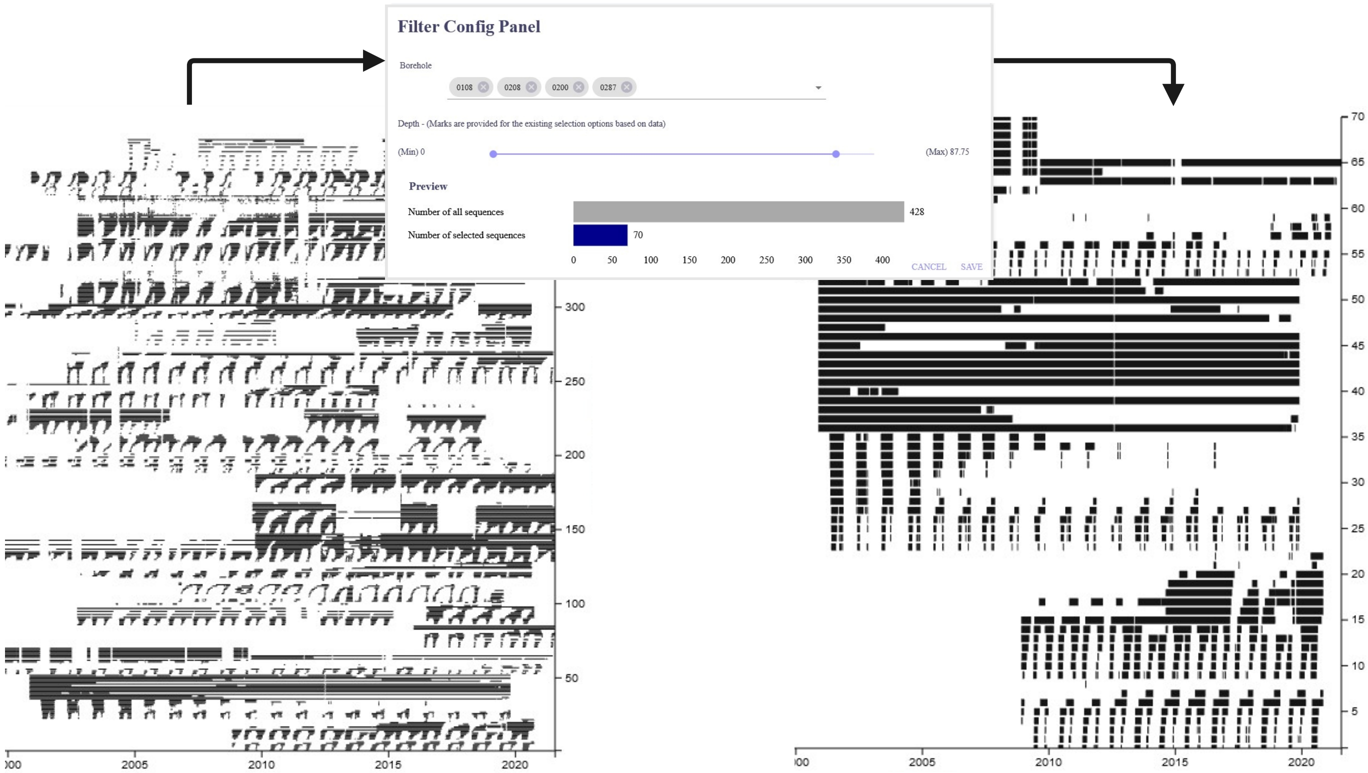

Fig. 9. *Filtering View* for a numerical metadata attribute in a filtering operation. The *Sequence Overview* on the left contains many hundreds of TSEQs, reduced by the filter to several dozens. Users can assess the filter strength already before its application, by using the preview shown at the bottom of the *Filtering View* (horizontal bar chart). In the example a user was interested in sensors located at low and medium depths of four boreholes, filtering other boreholes and sensors localized at higher depths. 70 TSEQs remained after filtering, for a more fine-grained inspection.

define the feature (sub-)set that goes into the clustering pipeline.

On the right of the *Clustering View*, we show the result of the dimensionality reduction method. In this way, users can gain an overview of TSEQs ($\mathbf{T}_1$) and structural characteristics of the global feature space ($\mathbf{T}_7$). In the scatter plot, point marks encode individual TSEQs, colored by the cluster colors. Both clustering visualizations support TSEQ selection, to be displayed and compared ($\mathbf{T}_3$) in the *Details View*. Furthermore, users can filter out TSEQs ($\mathbf{T}_9$) by right-clicking point marks, with implications on the global filter model and thus all linked views (*Sequence Overview*, *Details View*, *Metadata View*, and *Summary View*).

When users click on the "edit features" button, an auxiliary view appears, to modify the active feature set (left part of Figure 11). This list-based interface allows a) the selection of non-temporal features and b) the interactive customization of each feature through normalization methods. Histograms of the distributions of global feature values help users to make informed decisions. To foster algorithmic comparison across features of different scales, users can choose to normalize the global features with respect to the dataset using the max-norm and log-max norm. Changes made to the feature set automatically trigger the recalculation of the clustering and dimensionality reduction.

4.6 Implementation

IVESA is a React.js-based web application written in Typescript and powered by a Python backend. The user interfaces are based on Material-UI, while the interactive visualizations are built using D3.js [22] and D3FC for the data-intensive Overview. Frontend rendering is accelerated using WebGL and Web Workers. The IVESA VA approach runs as an ensemble of three Docker containers, deployed via Docker-Compose: NodeJS frontend, FastAPI-based Python backend running on a Uvicorn web server, and a MongoDB database. Frontend-backend interactions are facilitated by FastAPI endpoints using JSON-RPC format, except for the Overview endpoint which is using a TSV format optimized for frontend rendering. Data operations in the backend are performed using standard Python scientific libraries as Numpy, SciPy, Pandas, and scikit-learn [81].

5 Case Studies

We demonstrate the usefulness of IVESA in three case studies: (1) with TSEQs data about tweets on stocks in Section 5.1, (2) with TSEQs for pump activity in waste-water treatment in Section 5.2, and (3) on host behavior in a computer network in Section 5.3. The case studies describe three detailed workflows as observed from the user study presented in Section 6, with six experts for the six real-world cases. The selection of the three case studies was for most representative examples that showcase all main features of the system. In each case study, an expert was asked to freely explore the dataset, while thinking aloud. We introduce each case by briefly characterizing the dataset (detailed description in the supplemental material), the goals of the involved experts, and the main upstream preprocessing steps.

5.1 Apple & Cisco Stock Tweets

Wenuka is a data scientist with a passion for the analysis of the stock market involving tweet messages, as described in Section 3.3.5. With IVESA, Wenuka first focused on getting an overview of tweets, to assess whether the data distribution matched his expectations. Also, he was interested in the detection of outliers of the Apple stock from the beginning. For that purpose, Wenuka filtered TSEQs for #apple, and started using the *Metadata View*, to distribute tweet events by year (x-axis) and weekdays (y-axis), depicted in Figure 12. This enabled him to detect the expected: there are fewer tweets on weekends, as the stock market is closed. Wenuka further observed more tweets in 2020, which can be explained by the stock market crises due to the Covid-19 outbreak.

Next, Wenuka switched to the *Sequence Overview*. He sorted the TSEQs by the stock name and date (both are metadata attributes) such that he was able to infer a temporal trend on the y-axis for the tweets

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

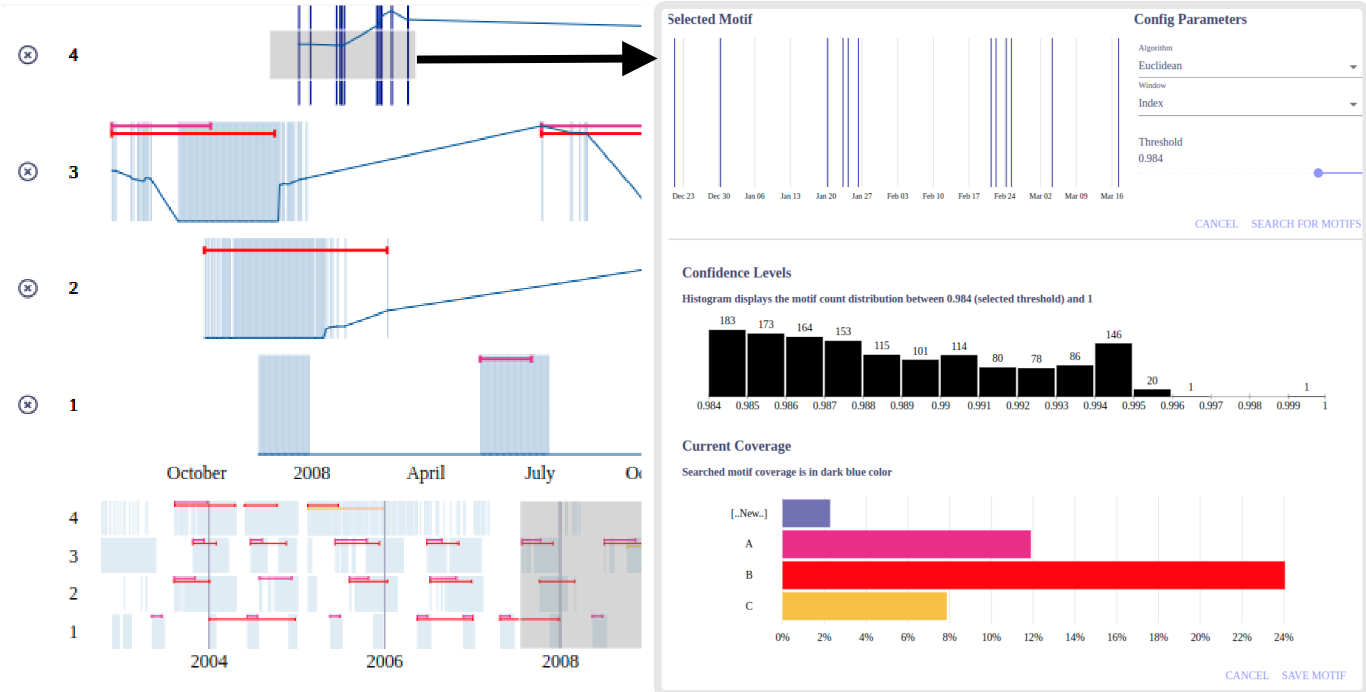

Fig. 10. The *Motif View* allows the guided definition of motifs, with a user-defined event subsequence as input. A black histogram guides users in the definition of the three motif configuration parameters. A colored bar chart at the bottom shows the percentage to which events of the dataset are covered by the different motifs, four in this case. In this example, the user was interested in this sequence of two brief melting periods, two extended melting periods, and again two shorter melting periods. They selected the pattern as a motif in the details view and searches for similar patterns withing the dataset using the index-based euclidian algorithm and a similarity threshold of 0.984.

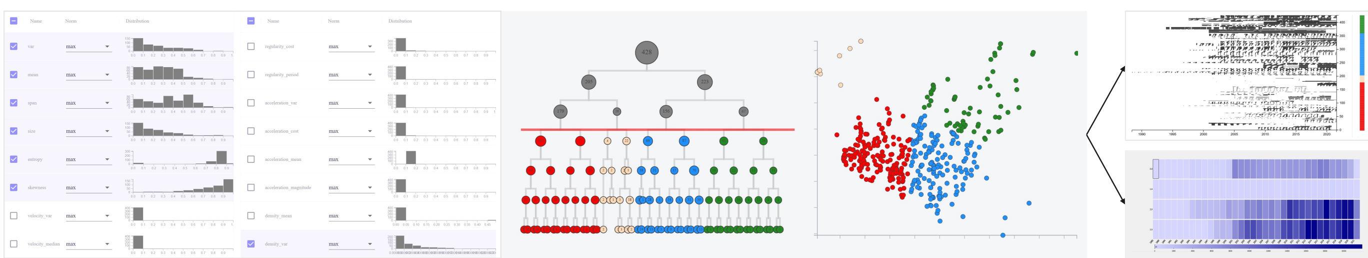

Fig. 11. Cluster linking workflow of IVESA. Based on a user-selected feature set (left view), the user creates a clustering result, here with four clusters (center view). The clustering result is automatically linked to the Sequence Overview (sorted by clusters) and the Metadata View (views on the right)

of each stock. In doing so, he noticed that there were some days when many Apple tweets were sent, with an interesting event pattern throughout the day with three dense phases standing out. He decided to mark this pattern and create a motif, to highlight these types of event signatures across the entire dataset: “My ultimate goal would be to find stock tweet movements similar to today’s movements.”. To find meaningful motifs with some degree of variety of the targeted pattern, Wenuka chose dynamic time warping in combination with a time window kernel, as parameters for motif search. However, with the default threshold, no similar patterns were found. This is not surprising to Wenuka: he explained this result through the heterogeneity of patterns that may exist across daily tweet signatures. Wenuka therefore iteratively refined motif-creation parameters using the subject distribution histogram as a guide, as it can be seen in the upper left part of Figure 13, to arrive at a meaningful subset of the dataset matching the motif search. The application of the motif-simplification routine highlights the motifs in the *Sequences Overview* (motif with pink color). To study the identified motif patterns in detail, he subsequently added these TSEQs to the *Details View* for closer inspection, depicted in the lower part of the figure.

To further analyze the days marked as motifs in the context of features, he selected the temporal entropy feature in the *Detail View*, to evaluate the unpredictability of tweets on social media data referring to the stock market. This entropy feature is also shown in Figure 13. He remarked that the resulting entropy per studied sequence was surprising, since the values did not meet his expectations for entropy outside of trading hours. After a double check with entropy values for other TSEQs, he came to the conclusion that his targeted motif was indeed special, if not unique.

To summarize, Wenuka used IVESA to (1) gain an overview of the tweets, (2) relate observed patterns to stock market events, (3) discover stock-price-related tweeting behaviors, (4) find similar daily tweeting motifs, and (5) compare tweeting behavior between different days.

5.2 Pump Activity in a Waste Water Treatment Plant

Olivia, the frontline analyst interested in three pumps for wastewater, is interested in expected and unexpected behavior for the dataset characterized in Section 3.3.6. Looking at the *Sequence Overview* at the start of the session, Olivia identified a pattern of considerably low pump activity at nighttime, depicted in Figure 14. This was in line with his prior knowledge: ”This is because people use their toilets less frequently at night, and hence we have less wastewater.” Next, Olivia

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

sorted the *Sequence Overview* by the pump name attribute, to detect trends for each of the three pumps separately. By leveraging the tooltip functionality in the overview, he compared several TSEQs with a focus on high event frequency. In this phase, he also selected TSEQs of interest, for a more detailed analysis in the *Detail View*. An interesting question would be if all interesting TSEQs stem from the same pump.

Olivia zoomed into the *Detail View* and looked at the nighttime measurements, depicted in Figure 15. He focused on one night, where one of the pumps was turned on much earlier than the other two pumps, which was quite surprising for him. In general, pump activity at night should be alternating, i.e., he identified a behavior that was atypical and not to explain semantically.

Finally, Olivia studied the metadata visualization. He was interested in seasonal effects, so he analyzed event distributions using year as the x-axis and months as the y-axis attribute. He identifies more pump events during the summer and explains that this is to be expected, as currently the WWTP lacks a separate wastewater system: normally, there is a sewage system for rainwater and one for household wastewater, but in his canton this is not the case. As it rains more frequently and for longer periods in the summer, it is natural that the pumps are activated more frequently and for longer periods of time.

In summary, Olivia used IVESA to (1) find and confirm daily patterns in the overview, (2) compare pumps to detect anomalies in pumping behavior, and (3) detect and confirm seasonal trends in the overview.

5.3 Host Behavior in a Computer Network

Andreas, the frontline analyst in our case on cybersecurity threats, works with TSEQs as described in Section 3.3.2. To identify the most noticeable patterns in the dataset, Andreas began by inspecting the overview in the *Sequence Overview*, as shown in Figure 16. What stood out in Andreas's perception were the variations of behaviors of workstations during the year, with several seasonal effects. For the yearly patterns, he noticed: ”You can see some kind of gradients here, that are presumably related to the semester breaks”. With this, Andreas was referring to the summer break, when there are no classes for the students and thus less activity at the workstations is being recorded. Andreas proceeded with the analysis and made an interesting assumption: ”Workstations should be separable into three categories, depending on different usage behaviors – (i) administration, (ii) international services, and (iii) students”. Andreas further formalized these three categories by the frequency and the regularity of communication. The administration communicates frequently but irregularly, the international services infrequently and irregularly, whereas the students' workstations communicate frequently and consistently. Figure 16 is sorted by these three device types and clearly reflects this tripartite separation. By looking more closely at the students' workstations, Andreas again confirmed the strong dependency of event counts with typical on/off semester semantics, especially for the summer break.

Andreas continued by looking deeper into the students' local temporal behaviors, the dominant group at the bottom in Figure 16. He identified a dominant pattern with a weekly duration: in many cases, workstations are highly active five days with a high productivity, followed by two days of idle, obviously on the weekend. Despite the fact that this weekly student pattern changes along the season (as described earlier), Andreas is surprised by the high within-group similarity: literally all students' workstations have the same behavior almost always.

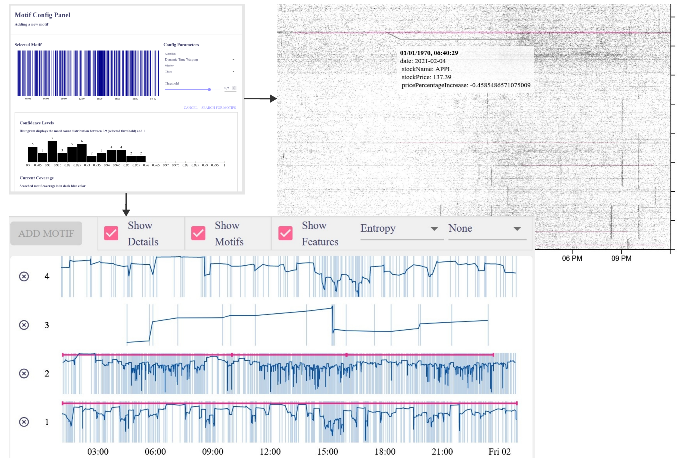

Fig. 13. Motif defined by the domain expert, with three dense phases of events (tweets) throughout a day. With the definition of this motif, IVESA automatically highlights event sequences matching this motif in the *Sequence Overview* and the *Detail View*, in this case with pink color.

Andreas continued the analysis by zooming into the workstations for international services. He discovered an unexpected anomaly: a vertical pattern where the workstations always communicate at the same time, marked with a red shape in Figure 16. For this reason, he compared a selected sequence of this category with two sequences selected from the student and administration categories. IVESA automatically adds all three selected sequences to the *Detail View*. In contrast to the anomaly sequence, he discovered that the administrative and student sequences are relatively similar in terms of their outliers.

Having the dominating patterns within the year and within weeks in mind, Andreas next used the *Metadata View*, and maps the year and day of the week attributes to the two chart axes. He chose the small multiples approach, to compare the three selected sequences by their metadata. He learned that the sequence from the international services communicates more frequently on the weekends, in contrast to the other two groups. This is a striking discovery, and he stated: ”These workstations appear to be rather odd, and workstation administrators, should definitely take a closer look”.

To contextualize and confirm the presence of the three categories of workstations, Andreas moved to the *Clustering View* provided as one of the auxiliary views in the *Summary View* in the lower right part of IVESA. Andreas selected six of the global features that he believed to be the most descriptive of the three categories of workstations. He subsequently defined the aggregation level in the cluster dendrogram in the *Clustering View* to 3, to assess the correspondence of the three resulting clusters with the three device types of the workstations, as depicted in Figure 17. He returned to the *Sequence Overview*, now ordered and color-coded by the three clusters, and noted that in fact the three clusters nicely reflect the three categories of devices. There is however some overlap between student and administrative categories. He considered this to suggest that either some administrative services have comparable communication habits to students, or the set of six selected features would require some refinement to make this classi-

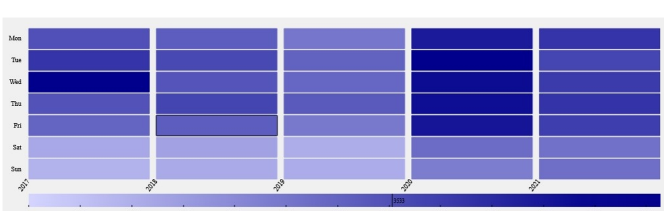

Fig. 12. Tweets featuring #apple or #csco across years (x-axis) and by weekdays (y-axis). During the weekend, activity diminishes.

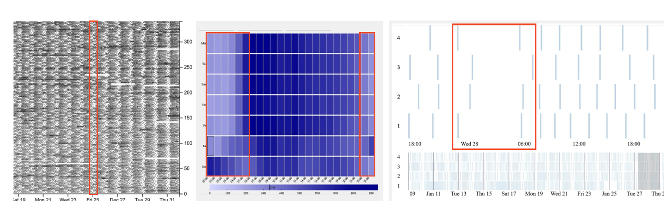

Fig. 14. *Sequence Overview*, *Metadata View*, and *Details View* showing the decreasing activity of pumps during the night.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

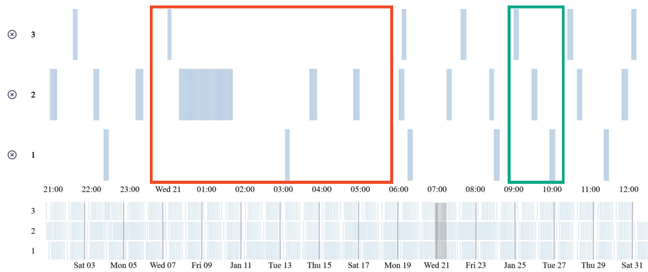

Fig. 15. Irregular pumping activity of three pumps of the WWTP that usually pump in an alternating manner.

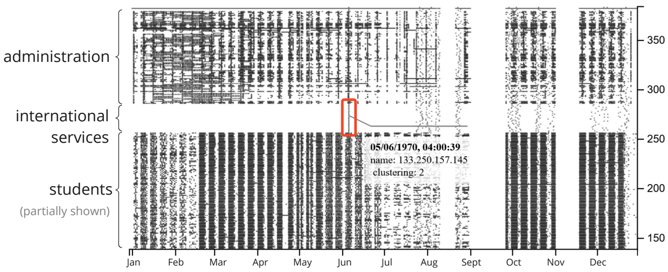

Fig. 16. Overview of incoming and outgoing communications of 384 workstations, grouped by the device category (see marks on the left). The three device categories show strong individual behaviors, as expected by the domain expert. In addition, he identified an unexpected vertical behavior of all international service devices (red mark).

fication entirely crisp. To confirm that clustering has indeed created distinct groups, Andreas utilized the small multiples functionality and created one small multiple per cluster, as shown in Figure 17. Here, Andreas chose hours and months on the x and y axes, respectively. Here, he made another finding: while most of the student activity occurs within working hours, the administrative services also communicate in the early morning, and the international services communicate quite independently of the time of day, except for a peak in June at 16:00. This is a very striking observation that he would like to pursue.

In summary, Andreas used IVESA to (1) identify weekly, yearly and seasonal patterns in the overview and contextualize them, (2) identify potential categories based on TSEQs similarity in the overview, (3) formalize identified categories by looking at the distribution of features on the TSEQs granularity, (4) confirm the categorization by using the clustering view, (5) compare clusters in the metadata view to confirm their dis-similarity, and (6) compare unexpected patterns using temporal features in the metadata view.

6 User Study

We evaluated IVESA by a summative user study with our six domain experts on the six different datasets and real-world cases, outlined in Section 3.1 and three of them exemplified in the case studies in Section 5. We briefly describe the experiment design, then present and discuss the results.

6.1 Methodology

To assess the utility and task usage of IVESA, we conducted a think-aloud observational user study that included qualitative coding of user feedback, a description of findings, and a reflection on the users' task usage and workflows. The user study lasted between 45 and 90 minutes, depending on the experts.

Participants: The user study was carried out with the previously introduced six domain experts using six datasets (overview in Figure 2):

- Nathan, the photographer who analyzed his personal photography collection
- Andreas, the cybersecurity expert with an interest in the analysis of host activity behavior
- Esmeralda, the Earth observation researcher interested in analyzing permafrost
- Reto, the radiologist focusing on radiological examinations and devices
- Wenuka, the data scientist interested in analyzing tweets about the stock market
- Olivia, the operations manager interested in analyzing waste water treatment.

Participants age ranged from 28 to 58 years (M=42, median=43, STD=9.5). Most participants were male (N=5). Participants had an average of 15 years of experience of working in their fields (M=15, median=20, STD=7.8).

Procedure: First we introduced IVESA to explain all its features and aggregation methods. The experts were asked to use IVESA to analyze their dataset while thinking aloud. Two evaluators were always present, one taking notes and the other answering questions. We analyzed the think-aloud notes using open coding [97]. We summarize the findings, with an affinity diagram that was iteratively created by two evaluators.

6.2 User Study Results

Observation of Experts Most domain experts began by gaining an understanding of the data distribution. Three experts used the *Sequence Overview* for this purpose, one used the *Metadata View* (x-axis: year, y-axis: month). "I want to get an overview over my data, therefore I am using years on the y-axis". After a familiarization phase, three branches in the experts' workflows could be identified: (1) comparisons between subsets, (2) identification of TSEQs patterns, and (3) confirmation of identified findings.

Comparisons: Some domain experts aimed at comparing subsets to a) detect differences in TSEQs or metadata, or b) to detect anomalies between the two subsets. Three approaches were used to compare subsets: a) looking for differences between two or more TSEQs, b) referring to metadata for comparison between subsets, and c) looking for similar subsets to compare to non-similar subsets. For comparisons between more than two TSEQs, the *Detail View* appeared to be useful: "Being able to compare different pumps is important for us to recognize irregularities". To compare and understand the distribution of metadata-based subsets, two domain experts sorted the *Sequence Overview* by metadata attributes. One domain expert compared metadata attributes using the small-multiple visualization in the *Metadata View*. Finally, one domain expert decided to use motifs to locate similar subsets, and to conduct similarity-based comparisons.

Pattern Identification: Using the *Sequence Overview*, *Details View*, and *Metadata View*, all experts discovered expected or unexpected patterns in their data. A primary focus of many experts was on temporal changes of event patterns. For that purpose, experts sorted the *Sequence Overview*, to find day-night, month, weekday, holiday, and metadata-related patterns in their data. "In the overview you can see that there is a gap in December and April. This is due to Christmas and Easter holidays". Two domain experts used the *Metadata View* axes to discover daily, weekday, monthly, seasonal, or yearly patterns. "There is something interesting on Wednesday in 2017 as it has double the events compared to the other weekdays". With the *Detail View*, the domain experts discovered event, hour, night-day, and seasonal patterns.

Pattern Confirmation: We observed that all domain experts wanted to leverage their previous knowledge to confirm patterns, possibly to assess the trustworthiness of IVESA. A prominent example was the stock market crisis in 2020. Most experts were able to confirm expectations. Beyond the confirmation of the expected, we identified a strong interest of several experts to find out *why* expectations were not met, or *how* an unexpected pattern can be explained.

6.3 Reflection on User Workflows

To better understand the workflow of the domain experts, we created state diagrams of the tool usage for each of the cases. The six state diagrams are shown in Figure 18. Each diagram is structured by the

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3382760

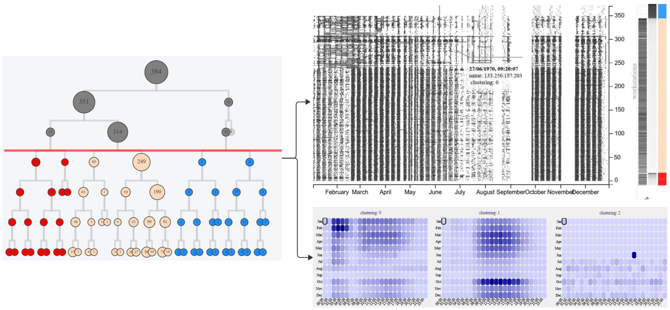

Fig. 17. Clustering result of the incoming and outgoing connections of 384 workstations (TSEQ). The separation into three groups shows that activity nicely aligns with the three device types, shown in Figure 16.

ten interactive tasks from left to right, arrows above nodes represent right-ward task transitions, arrows below nodes left-ward task transitions. We compare commonalities and differences across the six workflows using four analysis strategies: 1) *Task Coverage* by expert and case, 2) *Number of Task Transitions*, 3) *Start and End Points* of the experts' workflows, and 4) *Most Common Task Transition* between tasks.

Task Coverage: In the six cases, the domain experts covered 7.67 tasks on average. The lowest number of tasks was covered by the photography domain expert (N=5). Here, the expert was able to achieve his goals solely with overview ($\mathbf{T}_1$), detail ($\mathbf{T}_4$), and relate ($\mathbf{T}_5$) tasks (consume tasks), in combination with group ($\mathbf{T}_8$) and motif ($\mathbf{T}_{10}$) actions (produce tasks). In contrast, the stock market expert leveraged the most tasks to achieve his goal (N=10), followed by the host behavior expert (N=9). The coverage of tasks differs considerably across cases. Four tasks have always been utilized in every case: overview ($\mathbf{T}_1$), details ($\mathbf{T}_4$), relate ($\mathbf{T}_5$), and group ($\mathbf{T}_8$). Also, feature analysis ($\mathbf{T}_7$) based on metrics was relevant for 5/6 domain experts. Contrary to our expectations, the least observed task was outlier/anomaly detection ($\mathbf{T}_6$, 2x), followed by filtering ($\mathbf{T}_9$, 3x).

Number of Task Transitions: The number of task transitions reflects the complexity of a domain expert's workflow. In the user study, domain experts transitioned between tasks 14 times on average. Overall, the domain experts for stock market exchange (N=23) and host behavior (N=20) conducted the most complex workflow, with almost twice as many task transitions than other domain experts. The other workflows consisted of 9–11 transitions, with the least number performed by the domain expert for permafrost observations (N=9).

Start and End Points: We observe a dominating pattern for the preferred start of analysis: the overview of TSEQs ($\mathbf{T}_1$), conducted by four of the six experts. Interestingly, the other two experts decided for a grouping ($\mathbf{T}_8$) task to start with. In contrast, we cannot identify a dominating endpoint: five of the experts finished their analysis with different tasks.

Most Common Task Transition: It was particularly interesting to analyze which task tuples have been used most often in the same order, back to back. This will be a strong indicator for guidelines on effective interface and interaction design, with more awareness of frequently applied higher-order transitions. The most common transitions between tasks were:

- (N=5): TSEQs details ($\mathbf{T}_4$) to feature analysis ($\mathbf{T}_7$), possibly to assess if metrics are able to reflect important local characteristics found in TSEQs of interest
- (N=5): grouping TSEQs ($\mathbf{T}_8$) to relate ($\mathbf{T}_5$), possibly to contextualize groups of TSEQs with explanatory criteria such as metadata attributes
- (N=4): overviewing TSEQs ($\mathbf{T}_1$) to details ($\mathbf{T}_4$), possibly in line with the common principle of analyzing details on demand
- (N=4): details of TSEQs ($\mathbf{T}_4$) to event motif substitution ($\mathbf{T}_{10}$), possibly to transform an interesting sub-sequence pattern into a motif
- (N=4): substituting event motifs ($\mathbf{T}_{10}$) to overview ($\mathbf{T}_1$), possibly to analyze the effect of a changed motif set on the entire dataset
- (N=4): analyzing features ($\mathbf{T}_7$) to grouping ($\mathbf{T}_8$), possibly to inspect

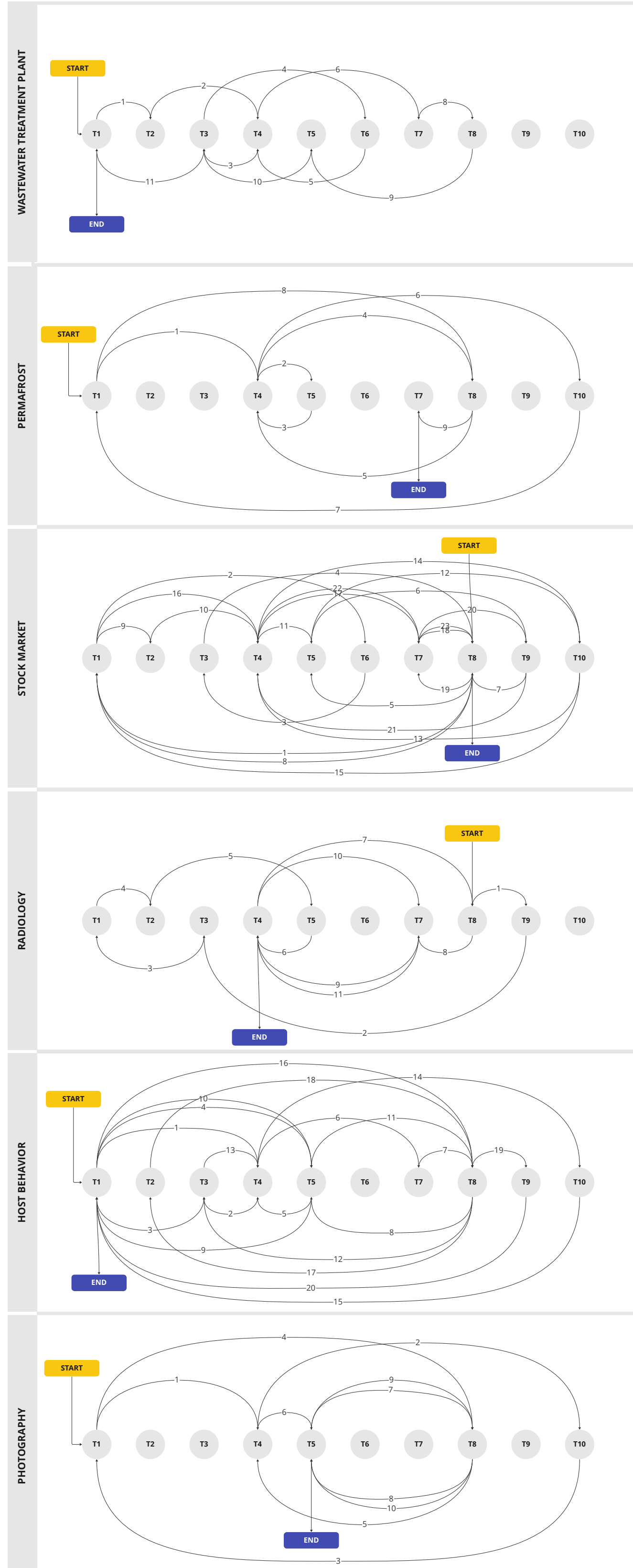

Fig. 18. State transition diagrams depicting the workflow of experts in the six real-world cases. To ease comparison, the ten interactive tasks $\mathbf{T}_{1\text{-}10}$ are aligned from left to right, orange, and purple labels mark the start and end of the experts' workflows. Arrows at the top represent right-ward transitions, arrows at the bottom indicate left-ward transitions, and arrow labels are the index of transition.

and refine the feature space to make the clustering more useful

7 Discussion and Future Work

We reflect on the design process of IVESA, discuss limitations, and outline future work possibilities.

7.1 Algorithmic Scalability

We have identified four issues that can lead to algorithmic scalability problems. The first is motif detection, as user-defined queries need to be executed at runtime. While the index-based variant in combination with Euclidean distance is quite scalable, the time-based kernel in combination with DTW is considerably slower, possibly due to known scalability issues of DTW. Future work includes a more in-depth study of implementation alternatives for dynamic time warping, such as fastdtw [88] or dtw-python [40]. Second, some metrics are computationally intensive. Specifically, temporal feature computation is time-expensive, especially if large sliding window sizes are requested by the user. As an example, computing the Connectivity-based Outlier Factor has quadratic computational complexity in terms of the event size (neighborhood) [7]. Our current strategy is to pre-compute all features, which is feasible as the input TSEQs are not changed at runtime. Third, we have decided on the benefits of a hierarchical clustering approach. To address the downside of a quadratic runtime, k-means or DBSCAN [19] would form more scalable clustering alternatives.

The final issue is the length of TSEQs in general. While IVESA scales well for 1,000 TSEQs with 1,000 events, the system performance might decrease for considerably more events, due to database communication and API call payloads.

7.2 Visual Scalability

We started with visualizing a few thousand events and applied a highly iterative design process including several technology changes to reach the design target of visualizing up to 1,000,000 events in a Web frontend. The *Sequence Overview* can take up to one second for both transporting and rendering the data points. Using GPU-accelerated graphics, re-rendering of the data can be done at 60fps while users interact with the data. With zooming applied, the approach is capable of interactive frame rates below 100ms refresh time. However, currently, only the *Sequence Overview* uses the GPU-accelerated graphics, leading to some delay regarding the other views, e.g., for single TSEQs with more than 100,000 events being displayed in the *Details View*.

7.3 Metrics and Feature Engineering

The user study with domain experts, each working on a different dataset and analysis problem, revealed insights into metrics that are applicable across applications and cases. However, we also observed that most cases would call for more domain-specific metrics, to better capture domain semantics through metrics and corresponding features. Future work in this direction is three-fold. First, we plan a more encompassing observation of users, possibly across dozens of real-world cases and datasets to collect more evidence about the relevance and generalizability of metrics. Second, it is our goal to conduct design studies for individual applications to investigate the support of domain-specific features. Finally, it would be interesting to investigate derived features more thoroughly, possibly in combination with the interactive feature selection procedures of IVESA.

7.4 User Studies

Our validation approach includes a) a generalizability strategy through the use of multiple datasets, b) intuitiveness and usability through an iterative design and refinement process, by leveraging expert feedback, and by conducting summative studies with the six domain experts, and c) usefulness through the presentation of three case studies. Future work includes more extensive user studies, to assess whether other domain experts can achieve their goals equally well. Also, working with non-experts will be an interesting avenue to assess their ability to use IVESA.

7.5 Dataset-Type-First Design Study

Our main abstraction method was applying the data-first design study principle by Oppermann and Munzner [80]. The principal idea is to acquire real-world data first and then select promising stakeholders, in contrast to classical design studies where stakeholders are identified first and datasets are subsequently provided by stakeholders [89]. Methodologically, we extend Oppermann's data-first principles in one particular aspect:

- First, we determined a dataset type to work on (uncovered by the methodologies)
- Second, we determined a heterogeneous set of (six) datasets for the dataset type
- Third, we selected promising stakeholders to work with these (six) datasets

While our abstraction results in Section 3 have been useful for designing a tool that supports domain experts in all six real-world cases, we feel that for TSEQs, our *dataset-type-first* approach would benefit from a field study with a larger number of observed real-world cases to further assess the relevance of the analysis tasks. To complement the dataset-type-first methodology, further research will include observational studies to characterize analytic strategies to cope with dataset complexities, such as conducted by Du et al. for classical event sequences [34]

8 Conclusion

We have presented a VA approach to support users with the analysis of TSEQs. Studying this data type with six datasets and domain experts enabled us to identify 12 tasks and sets of 8 temporal and 14 static metrics. The iteratively designed VA tool IVESA includes multiple linked views that enable users to explore TSEQs, relate the sequence content to metadata attributes, and produce data simplifications that help them cope with large and complex datasets. We validated IVESA through three case studies and a user study with six domain experts with six real-world cases. Results demonstrate the usability of the approach and its generalizability across cases and applications. We were pleased that the domain experts were able to interactively analyze TSEQs with up to 1,000,000 events. While the supported tasks and provided metrics were deemed useful, there are indications of the need for more individualized and domain-specific solutions. Promising directions of future work include VA support for the identification of domain-specific metrics, the synthesis of features, and the domain-specific identification of motifs.

Acknowledgments

The authors wish to thank Kevin Bohn for his support of the development of IVESA.

References

[1] J. Abello, F. van Ham, and N. Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 12(5):669–676, 2006. doi: 10.1109/TVCG.2006.120

[2] C. Ahlberg and B. Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 313–317. ACM, 1994. doi: 10.1145/191666.191775

[3] Z. Ahmed, P. Yost, A. McGovern, and C. Weaver. Steerable clustering for visual analysis of ecosystems. In *EuroVis Workshop on Visual Analytics (EuroVA)*. Eurographics Association, 2011. doi: 10.2312/PE/EuroVAST/EuroVA11/049-052

[4] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011. doi: 10.1007/978-0-85729-079-3

[5] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, second edition ed., 2023. doi: 10.1007/978-1-4471-7527-8

[6] Y. Akiyama, T. Hosoya, A. M. Poole, and Y. Hotta. The gcm-motif: a novel dna-binding motif conserved in drosophila and mammals. *National Academy of Sciences (USA)*, 93(25):14912–14916, 1996.

[7] O. Alghushairy, R. Alsini, T. Soule, and X. Ma. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing*, 5(1), 2021. doi: 10.3390/bdcc5010001

[8] N. V. Andrienko and G. L. Andrienko. *Exploratory analysis of spatial and temporal data - a systematic approach*. Springer, 2006. doi: 10.1007/3-540-31190-4

[9] D. Archambault, T. Munzner, and D. Auber. Tuggraph: Path-preserving hierarchies for browsing proximity and paths in graphs. In *IEEE Pacific Visualization Symposium (PacificVis)*, pp. 113–120. IEEE Computer Society, 2009. doi: 10.1109/PACIFICVIS.2009.4906845

[10] J. Bernard. Exploratory search in time-oriented primary data, December 2015.

[11] J. Bernard, D. Daberkow, D. Fellner, K. Fischer, O. Koepler, J. Kohlhammer, M. Runnwerth, T. Ruppert, T. Schreck, and I. Sens. Visinfo: a digital library system for time series research data based on exploratory search - a user-centered design approach. *International Journal on Digital Libraries (IJoDL)*, 16(1), 2015. doi: 10.1007/s00799-014-0134-y

[12] J. Bernard, E. Dobermann, M. Bögl, M. Röhlig, A. Vögele, and J. Kohlhammer. Visual-Interactive Segmentation of Multivariate Time Series. In *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2016. doi: 10.2312/eurova.20161121

[13] J. Bernard, M. Hutter, H. Reinemuth, H. Pfeifer, C. Bors, and J. Kohlhammer. Visual-interactive preprocessing of multivariate time series data. *Computer Graphics Forum*, 38(3):401–412, 2019. doi: 10.1111/cgf.13698

[14] J. Bernard, T. Ruppert, O. Goroll, T. May, and J. Kohlhammer. Visual-interactive preprocessing of time series data. In *Interactive Visual Analysis of Data (SIGRAD)*, vol. 81, pp. 39–48. Linköping University Electronic Press, 2012.

[15] J. Bernard, T. Ruppert, M. Scherer, J. Kohlhammer, and T. Schreck. Content-based layouts for exploratory metadata search in scientific research data. In *ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 139–148. ACM, 2012. doi: 10.1145/2232817.2232844

[16] J. Bernard, T. Ruppert, M. Scherer, T. Schreck, and J. Kohlhammer. Guided discovery of interesting relationships between time series clusters and metadata properties. In *Conference on Knowledge Management and Knowledge Technologies I-KNOW*, p. 22. ACM, 2012. doi: 10.1145/2362456.2362485

[17] J. Bernard, D. Sessler, J. Kohlhammer, and R. A. Ruddle. Using dashboard networks to visualize multiple patient histories: A design study on post-operative prostate cancer. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 25(3):1615–1628, 2019. doi: 10.1109/TVCG.2018.2803829

[18] J. Bernard, N. Wilhelm, B. Krüger, T. May, T. Schreck, and J. Kohlhammer. Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(12):2257–2266, 2013. doi: 10.1109/TVCG.2013.178

[19] D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, 60(1):208–221, 2007.

[20] G. Bisson and R. Blanch. Improving visualization of large hierarchical clustering. In *Information Visualisation, IV*, pp. 220–228. IEEE Computer Society, 2012. doi: 10.1109/IV.2012.45

[21] C. Bors, J. Bernard, M. Bögl, T. Gschwandtner, J. Kohlhammer, and S. Miksch. Quantifying Uncertainty in Multivariate Time Series Pre-Processing. In T. v. Landesberger and C. Turkay, eds., *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2019. doi: 10.2312/eurova.20191121

[22] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185

[23] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(12):2376–2385, 2013. doi: 10.1109/TVCG.2013.124

[24] P. Buono, A. Aris, C. Plaisant, A. Khella, and B. Shneiderman. Interactive pattern search in time series. In *SPIE Visualization and Data Analysis (VDA)*, vol. 5669, pp. 175 – 186. SPIE, 2005. doi: 10.1117/12.587537

[25] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. In *SIGKDD Conference on Knowledge Discovery and Data Mining*, p. 280–284. ACM, 2000. doi: 10.1145/347090.347151

[26] N. Cao, D. Gotz, J. Sun, and H. Qu. Dicon: Interactive visual analysis of multidimensional clusters. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 17(12):2581–2590, 2011. doi: 10.1109/TVCG.2011.188

[27] N. Cao, C. Shi, S. Lin, J. Lu, Y. Lin, and C. Lin. Targetvue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 22(1):280–289, 2016. doi: 10.1109/TVCG.2015.2467196

[28] B. C. M. Cappers and J. J. van Wijk. Exploring multivariate event sequences using rules, aggregations, and selections. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 24(1):532–541, 2018. doi: 10.1109/TVCG.2017.2745278

[29] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. A. Suma, C. Ziemkiewicz, D. A. Kern, and A. Sudjianto. Wirevis: Visualization of categorical, time-varying data from financial transactions. In *IEEE Visual Analytics Science and Technology (VAST)*, pp. 155–162. IEEE Computer Society, 2007. doi: 10.1109/VAST.2007.4389009

[30] L. Cibulski, T. May, B. Preim, J. Bernard, and J. Kohlhammer. Visualizing time series consistency for feature selection. 2019. doi: 11025/35593

[31] W. Cui, S. Liu, Z. Wu, and H. Wei. How hierarchical topics evolve in large text corpora. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 20(12):2281–2290, 2014. doi: 10.1109/TVCG.2014.2346433

[32] F. Du, S. Guo, S. Malik, E. Koh, S. Kim, and Z. Liu. Interactive event sequence prediction for marketing analysts. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–8. ACM, 2020. doi: 10.1145/3334480.3382971

[33] F. Du, C. Plaisant, N. Spring, K. Crowley, and B. Shneiderman. Eventaction: A visual analytics approach to explainable recommendation for event sequences. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 9(4), 2019. doi: 10.1145/3301402

[34] 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 (TVCG)*, 23(6):1636–1649, 2017. doi: 10.1109/TVCG.2016.2539960

[35] J. Eirich, J. Bonart, D. Jäckle, M. Sedlmair, U. Schmid, K. Fischbach, T. Schreck, and J. Bernard. Irvine: A design study on analyzing correlation patterns of electrical engines. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):11–21, 2022. doi: 10.1109/TVCG.2021.3114797

[36] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *National Academy of Sciences (USA)*, 95(25):14863–14868, 1998.

[37] N. Elmqvist and J. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 16(3):439–454, 2010. doi: 10.1109/TVCG.2009.84

[38] T. Fu. A review on time series data mining. *Eng. Appl. Artif. Intell.*, 24(1):164–181, 2011. doi: 10.1016/j.engappai.2010.09.007

[39] P. Geurts. Pattern extraction for time series classification. In *Principles of Data Mining and Knowledge Discovery*, pp. 115–127. Springer Berlin Heidelberg, 2001.

[40] T. Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software*, 31(7):1–24, 2009. doi: 10.18637/jss.v031.i07

[41] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011. doi: 10.1177/1473871611416549

[42] M. K. Goldberg, M. Hayvanovych, and M. Magdon-Ismail. Measuring similarity between sets of overlapping clusters. In *IEEE Conference on Social Computing*, pp. 303–308, 2010. doi: 10.1109/SocialCom.2010.50

[43] D. Gotz, J. Zhang, W. Wang, J. Shrestha, and D. Borland. Visual analysis of high-dimensional event sequence data via dynamic hierarchical aggregation. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 26(1):440–450, 2020. doi: 10.1109/TVCG.2019.2934661

[44] D. Guo, J. Chen, A. MacEachren, and K. Liao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 12(6):1461–1474, 2006. doi: 10.1109/TVCG.2006.84

[45] 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, 2020. PacificVis Workshop on Visualization Meets AI. doi: 10.1016/j.visinf.2020.04.001

[46] S. Guo, F. Du, S. Malik, E. Koh, S. Kim, Z. Liu, D. Kim, H. Zha, and N. Cao. Visualizing uncertainty and alternatives in event sequence predictions. In *ACM Conference on Human Factors in Computing Systems (CHI)*, p. 1–12. ACM, 2019. doi: 10.1145/3290605.3300803

[47] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao. Eventthread: Visual summarization and stage analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 24(1):56–65, 2018. doi: 10.1109/TVCG.2017.2745320

[48] 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 (TVCG)*, 28(12):5091–5112, 2022. doi: 10.1109/TVCG.2021.3100413

[49] K. Heins and H. Stern. A Statistical Model for Event Sequence Data. In *Conference on Artificial Intelligence and Statistics*, vol. 33, pp. 338–346. PMLR, 2014.

[50] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004. doi: 10.1057/palgrave.ivs.9500061

[51] Z. Jin, S. Cui, S. Guo, D. Gotz, J. Sun, and N. Cao. Carepre: An intelligent clinical decision assistance system. *ACM Transactions on Computing for Healthcare*, 1(1), 2020. doi: 10.1145/3344258

[52] T. Jirsik and P. Velan. Host behavior in computer network: One-year study. *IEEE Transactions on Network and Service Management*, 18(1):822–838, 2021. doi: 10.1109/TNSM.2020.3036528

[53] I. T. Jolliffe. Principal component analysis. In *International Encyclopedia of Statistical Science*, pp. 1094–1096. Springer, 2011. doi: 10.1007/978-3-642-04898-2

[54] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 6(1):59–78, 2000. doi: 10.1109/2945.841121

[55] E. J. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Min. Knowl. Discov.*, 7(4):349–371, 2003. doi: 10.1023/A:1024988512476

[56] R. Kincaid. Signallens: Focus+context applied to electronic time series. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 16(6):900–907, 2010. doi: 10.1109/TVCG.2010.193

[57] S. Koldijk, J. Bernard, T. Ruppert, J. Kohlhammer, M. Neerincx, and W. Kraaij. Visual Analytics of Work Behavior Data - Insights on Individual Differences. In E. Bertini, J. Kennedy, and E. Puppo, eds., *Eurographics Conference on Visualization (EuroVis) - Short Papers*. The Eurographics Association, 2015. doi: 10.2312/eurovisshort.20151129

[58] J. Krause, A. Perer, and E. Bertini. INFUSE: interactive feature selection for predictive modeling of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 20(12):1614–1623, 2014. doi: 10.1109/TVCG.2014.2346482

[59] M. Krstajic, E. Bertini, and D. Keim. Cloudlines: Compact display of event episodes in multiple time-series. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 17(12):2432–2439, 2011. doi: 10.1109/TVCG.2011.179

[60] N. Kumar, V. N. Lolla, E. J. Keogh, S. Lonardi, and C. A. Ratanamahatana. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SIAM Conference on Data Mining*, pp. 531–535. SIAM, 2005. doi: 10.1137/1.9781611972757.55

[61] P. Law, Z. Liu, S. Malik, and R. C. Basole. Maqui: Interweaving queries and pattern mining for recursive event sequence exploration. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 25(1):396–406, 2019. doi: 10.1109/TVCG.2018.2864886

[62] T. W. Liao. Clustering of time series data - a survey. *Pattern Recognit.*, 38(11):1857–1874, 2005. doi: 10.1016/j.patcog.2005.01.025

[63] J. Lin, E. J. Keogh, and S. Lonardi. Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization*, 4(2):61–82, 2005. doi: 10.1057/palgrave.ivs.9500089

[64] J. Lin, E. J. Keogh, S. Lonardi, J. P. Lankford, and D. M. Nystrom. Visually mining and monitoring massive time series. In *SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 460–469. ACM, 2004. doi: 10.1145/1014052.1014104

[65] Z. Liu, B. Kerr, M. Dontcheva, J. Grover, M. Hoffman, and A. Wilson. Coreflow: Extracting and visualizing branching patterns from event sequences. *Computer Graphics Forum (CGF)*, 36(3):527–538, 2017. doi: 10.1111/cgf.13208

[66] 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 (TVCG)*, 23(1):321–330, 2017. doi: 10.1109/TVCG.2016.2598797

[67] J. Lonardi and P. Patel. Finding motifs in time series. In *Workshop on Temporal Data Mining*, pp. 53–68, 2002.

[68] E. Maguire, P. Rocca-Serra, S.-A. Sansone, J. Davies, and M. Chen. Visual compression of workflow visualizations with automated detection of macro motifs. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(12):2576–2585, 2013. doi: 10.1109/TVCG.2013.225

[69] S. Malik, F. Du, M. Monroe, E. Onukwugha, C. Plaisant, and B. Shneiderman. Cohort comparison of event sequences with balanced integration of visual analytics and statistics. In *Intelligent User Interfaces (IUI)*, p. 38–49. ACM, 2015. doi: 10.1145/2678025.2701407

[70] S. Malik, B. Shneiderman, F. Du, C. Plaisant, and M. Bjarnadottir. High-volume hypothesis testing: Systematic exploration of event sequence comparisons. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(1), 2016. doi: 10.1145/2890478

[71] P. McLachlan, T. Munzner, E. Koutsofios, and S. C. North. Liverac: interactive visual exploration of system management time-series data. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 1483–1492. ACM, 2008. doi: 10.1145/1357054.1357286

[72] W. Meert, K. Hendrickx, and T. V. Craenendonck. wannesm/dtaidistance v2.0.0. Zenodo, 2020. doi: 10.5281/zenodo.3981067

[73] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(12):2227–2236, 2013. doi: 10.1109/TVCG.2013.200

[74] F. Mörchen. Algorithms for time series knowledge mining. In *SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 668–673. ACM, 2006. doi: 10.1145/1150402.1150485

[75] T. Mühlbacher and H. Piringer. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(12):1962–1971, 2013. doi: 10.1109/TVCG.2013.125

[76] M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 137–146. Eurographics Association, 2006. doi: 10.2312/SCA/SCA06/137-146

[77] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 15(6):921–928, 2009. doi: 10.1109/TVCG.2009.111

[78] A. Nanopoulos, R. Alcock, and Y. Manolopoulos. *Feature-Based Classification of Time-Series Data*, p. 49–61. Nova Science Publishers, Inc., 2001.

[79] P. H. Nguyen, C. Turkay, G. Andrienko, N. Andrienko, O. Thonnard, and J. Zouaoui. Understanding user behaviour through action sequences: From the usual to the unusual. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 25(9):2838–2852, 2019. doi: 10.1109/TVCG.2018.2859969

[80] M. Oppermann and T. Munzner. Data-first visualization design studies. In *IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*, pp. 74–80. IEEE, 2020. doi: 10.1109/BELIV51497.2020.00016

[81] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[82] Y. Peiris, C. Barth, E. M. Huang, and J. Bernard. A data-centric methodology and task typology for time-stamped event sequences. In *IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*, pp. 66–76. IEEE, 2022. doi: 10.1109/BELIV57783.2022.00012

[83] A. Perer and F. Wang. Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Intelligent User Interfaces (IUI)*, p. 153–162. ACM, 2014. doi: 10.1145/2557500.2557508

[84] R. Ruddle, J. Bernard, H. Lücke-Tieke, T. May, and J. Kohlhammer. The effect of alignment on peoples ability to judge event sequence similarity. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, pp. 1–1, 2021. doi: 10.1109/TVCG.2021.3050497

[85] R. Ruddle, J. Bernard, T. May, H. Lücke-Tieke, and J. Kohlhammer. Methods and a research agenda for the evaluation of event sequence visualization techniques, 2016. The Event Event: Temporal & Sequential Event Analysis - IEEE VIS 2016 Workshop.

[86] M. Sachdeva, J. Burmeister, J. Kohlhammer, and J. Bernard. Lfpeers: Temporal similarity search and result exploration. *Computers Graphics*, 115:81–95, 2023. doi: 10.1016/j.cag.2023.06.009

[87] R. Sahann, T. Möller, and J. Schmidt. Histogram binning revisited with

a focus on human perception. In *IEEE Visualization Conference (VIS) - Short Papers*, pp. 66–70. IEEE, 2021. doi: 10.1109/VIS49827.2021.9623301

[88] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[89] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 18(12):2431–2440, 2012. doi: 10.1109/TVCG.2012.213

[90] J. Seo and B. Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *IEEE Symposium on Information Visualization (InfoVis)*, pp. 65–72, 2004. doi: 10.1109/INFVIS.2004.3

[91] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages*, pp. 336–343. IEEE Computer Society, 1996. doi: 10.1109/VL.1996.545307

[92] B. Shneiderman. The event quartet: How visual analytics works for temporal data. In *IEEE VIS Workshop on Temporal & Sequential Event Analysis*, 2016.

[93] G. Smith, M. Czerwinski, B. Meyers, D. C. Robbins, G. G. Robertson, and D. S. Tan. Facetmap: A scalable search and browse visualization. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 12(5):797–804, 2006. doi: 10.1109/TVCG.2006.142

[94] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. *Computer Graphics Forum (CGF)*, 33(3):401–410, 2014. doi: 10.1111/cgf.12396

[95] S. Stober and A. Nürnberger. Musicgalaxy: A multi-focus zoomable interface for multi-facet exploration of music collections. In *Exploring Music Contents CMMR*, vol. 6684, pp. 273–302. Springer, 2010. doi: 10.1007/978-3-642-23126-1

[96] C. D. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 20(12):1653–1662, 2014. doi: 10.1109/TVCG.2014.2346574

[97] A. Strauss and J. Corbin. *Basics of qualitative research*. Sage publications, 1990.

[98] Swiss Permafrost Monitoring Network (PERMOS). Swiss permafrost monitoring network database (permos database), 2021. doi: 10.13093/PERMOS-2021-01

[99] J. Tang, Z. Chen, A. W.-C. Fu, and D. W.-L. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, p. 535–548. Springer-Verlag, 2002.

[100] C. Tominski, G. Fuchs, and H. Schumann. Task-driven color coding. In *Information Visualisation, IV*, pp. 373–380. IEEE Computer Society, 2008. doi: 10.1109/IV.2008.24

[101] C. Turkay, J. Parulek, N. Reuter, and H. Hauser. Interactive visual analysis of temporal cluster structures. *Computer Graphics Forum (CGF)*, 30(3):711–720, 2011. doi: 10.1111/j.1467-8659.2011.01920.x

[102] S. van den Elzen and J. J. van Wijk. Small multiples, large singles: A new approach for visual data exploration. *Computer Graphics Forum (CGF)*, 32(3):191–200, 2013. doi: 10.1111/cgf.12106

[103] J. J. van Wijk and E. R. van Selow. Cluster and calendar based visualization of time series data. In *IEEE Symposium on Information Visualization (InfoVis)*, pp. 4–9. IEEE Computer Society, 1999. doi: 10.1109/INFVIS.1999.801851

[104] K. Vrotsou and A. Nordman. Exploratory visual sequence mining based on pattern-growth. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 25(8):2597–2610, 2019. doi: 10.1109/TVCG.2018.2848247

[105] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao. Unsupervised clickstream clustering for user behavior analysis. In *ACM Conference on Human Factors in Computing Systems (CHI)*, p. 225–236. ACM, 2016. doi: 10.1145/2858036.2858107

[106] 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 *ACM Conference on Human Factors in Computing Systems (CHI)*, p. 457–466. ACM, 2008. doi: 10.1145/1357054.1357129

[107] T. D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, and M. Smith. Temporal summaries: Supporting temporal categorical searching, aggregation and comparison. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 15(6):1049–1056, 2009. doi: 10.1109/TVCG.2009.187

[108] J. Wei, Z. Shen, N. Sundaresan, and K. Ma. Visual cluster exploration of web clickstream data. In *IEEE Visual Analytics Science and Technology (VAST)*, pp. 3–12, 2012. doi: 10.1109/VAST.2012.6400494

[109] R. W. White and R. A. Roth. *Exploratory Search: Beyond the Query-Response Paradigm*. Morgan & Claypool Publishers, 2009. doi: 10.2200/S00174ED1V01Y200901ICR003

[110] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. Lifeflow: Visualizing an overview of event sequences. In *ACM Conference on Human Factors in Computing Systems (CHI)*, p. 1747–1756. ACM, 2011. doi: 10.1145/1978942.1979196

[111] K. Wongsuphasawat, D. Moritz, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 22(1):649–658, 2016. doi: 10.1109/TVCG.2015.2467191

[112] K. Wongsuphasawat, C. Plaisant, M. Taieb-Maimon, and B. Shneiderman. Querying event sequences by exact match or similarity search: Design and empirical evaluation. *Interacting with Computers*, 24(2):55–68, 2012. doi: 10.1016/j.intcom.2012.01.003

[113] C. Xie, W. Xu, and K. Mueller. A visual analytics framework for the detection of anomalous call stack trees in high performance computing applications. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 25(1):215–224, 2019. doi: 10.1109/TVCG.2018.2865026

[114] 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 (TVCG)*, 23(1):291–300, 2017. doi: 10.1109/TVCG.2016.2598664

[115] X. Zhang, H.-F. Brown, and A. Shankar. Data-driven personas: Constructing archetypal users with clickstreams and user telemetry. In *ACM Conference on Human Factors in Computing Systems (CHI)*, p. 5350–5359. ACM, 2016. doi: 10.1145/2858036.2858523

[116] J. Zhao, N. Cao, Z. Wen, Y. Song, Y. Lin, and C. Collins. Fluxflow: Visual analysis of anomalous information spreading on social media. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 20(12):1773–1782, 2014. doi: 10.1109/TVCG.2014.2346922

[117] J. Zhao, F. Chevalier, and R. Balakrishnan. Kronominer: using multi-foci navigation for the visual exploration of time-series data. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 1737–1746. ACM, 2011. doi: 10.1145/1978942.1979195

[118] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 17(12):2422–2431, 2011. doi: 10.1109/TVCG.2011.195

[119] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. Matrixwave: Visual comparison of event sequence data. In *ACM Conference on Human Factors in Computing Systems (CHI)*, p. 259–268. ACM, 2015. doi: 10.1145/2702123.2702419

[120] H. Ziegler, M. Jenny, T. Gruse, and D. A. Keim. Visual market sector analysis for financial time series data. In *IEEE Visual Analytics Science and Technology (VAST)*, pp. 83–90. IEEE Computer Society, 2010. doi: 10.1109/VAST.2010.5652530