Understanding User Behaviour through Action Sequences: from the Usual to the Unusual

Phong H. Nguyen, Cagatay Turkay, Gennady Andrienko, Natalia Andrienko,
Olivier Thonnard, and Jihane Zouaoui

Abstract—Action sequences, where atomic user actions are represented in a labelled, timestamped form, are becoming a fundamental data asset in the inspection and monitoring of user behaviour in digital systems. Although the analysis of such sequences is highly critical to the investigation of activities in cyber security applications, existing solutions fail to provide a comprehensive understanding due to the complex semantic and temporal characteristics of these data. This paper presents a visual analytics approach that aims to facilitate a user-involved, multi-faceted decision making process during the identification and the investigation of "unusual" action sequences. We first report the results of the task analysis and domain characterisation process. Then we describe the components of our multi-level analysis approach that comprises of constraint-based sequential pattern mining and semantic distance based clustering, and multi-scalar visualisations of users and their sequences. Finally, we demonstrate the applicability of our approach through a case study that involves tasks requiring effective decision-making by a group of domain experts. Although our solution here is tightly informed by a user-centred, domain-focused design process, we present findings and techniques that are transferable to other applications where the analysis of such sequences is of interest.

Index Terms—action sequence, event sequence, sequential pattern mining, visual analytics, cyber security, user behaviour.

✦

1 INTRODUCTION

THE analysis of user actions for a better understanding of how users of a digital system "behave" during their interaction with a system has significant prominence in several domains such as systems design, cyber security and education [1]. Within cyber security in particular, understanding, modelling and monitoring user behaviour are highly critical for effective mechanisms that can detect complex and often human-induced "insider" threats to cyber infrastructure [2]. Action sequences, where atomic user actions within a system are represented in a labelled, timestamped form, are often gathered to develop such methods [3]. Through the analysis of the sequences, the activities of users are monitored and identified for further investigation, and eventually are decided (by analysts) whether they are indeed "suspicious" [3]. Achieving these effectively requires a thorough understanding of how users behave.

User behaviour, however, is complex by nature, and so are the resulting sequences that capture them. The sequences often comprise several semantically related *patterns* (series of actions) that are driven by *user intent* which varies significantly with users and time. These characteristics (e.g., users, time, varying behaviour and user roles) often lead to a high level of uncertainty within the fully automated analysis of such data, making it highly challenging for analysts to make well-informed, robust decisions while evaluating activities. Effective decision-making through the analysis of

data with such complex characteristics requires a comprehensive understanding of all the facets of the data concurrently. We are motivated by these challenges and present a visual analytics approach that aims to facilitate a user-involved, multi-faceted decision making process during the identification, investigation and evaluation of "unusual" action sequences. The approach includes visualisation designs and interaction techniques with integrated computational methods that provide analysts *multi-level overviews* and ways to conduct *multi-faceted comparative investigations* into large collections of action sequences. We also propose a novel interaction technique, *multi-semantic linking*, to seamlessly link data facets with complex semantic links. Some initial ideas and results of this work were discussed in the 2017 EuroVA workshop [4].

Our complete approach is designed and developed through a user-centred process as a collaborative team of visualisation and cyber security experts. We carry out a series of workshops to identify the characteristics of the data and to elicit the domain requirements. We externalise our findings as a taxonomy of goals and tasks, which are then addressed through novel designs and computational methods that are iteratively developed through further workshops. The resulting prototype, named **U4**, is then evaluated by security experts in terms of its effectiveness.

To summarise, the contributions of this paper include:
- Observations, findings and lessons learnt from a user-centred design process within a cyber-security context
- A method to identify high-level, semantically relevant patterns, *activities*, from raw action sequences
- Multi-level representation of user sessions and activities (Section 5) combined with *multi-semantic linking* interaction (Section 6) to enable multi-faceted overview, in-depth exploration and comparative analysis

- *Phong H. Nguyen and Cagatay Turkay are with City, University of London, UK. E-mail: {p.nguyen, Cagatay.Turkay.1}@city.ac.uk.*
- *Gennady Andrienko and Natalia Andrienko are with Fraunhofer IAIS, Germany and City, University of London, UK.*
  E-mail: {gennady,natalia}.andrienko@iais.fraunhofer.de.
- *Olivier Thonnard and Jihane Zouaoui are with Amadeus, France.*
  E-mail: {olivier.thonnard, jihane.zouaoui}@amadeus.com.

- A user-centred evaluation with security experts using our tool to analyse data from a user management system and make decisions that require a comprehensive understanding of action sequences (Section 7)

2 DOMAIN PROBLEM CHARACTERISATION

This section presents the first stage of our user-centred design process: understanding user problems and eliciting requirements. First, we describe a use case where analysis of action sequence data is essential. We then discuss characteristics of the data involved in such analysis. Finally, we present the important goals in the analysis and the specific tasks to achieve those goals.

2.1 Motivating Use Case

In this paper, we are motivated by a use case identified during a preliminary phase of a multi-disciplinary research project on enhancing security information and event management systems [1]. In this use case, the cyber security experts are interested in detecting possible misuses or fraudulent activities that are carried out using the administrative interface of a login and security server. This application manages user authentication, access control and more sophisticated user rights. Because of the severity of the application, it is crucial for the experts to understand how it has been used. Examples of unusual activities are as follows.

- User X in one session deletes many account profiles, whereas he normally does not perform such an action.
- User Y searches for hundreds of account details, indicating that he may want to steal personal information.
- User Z performs actions in a very high rate, suggesting that her account may be used for automatic execution.

To enable this investigation, the experts capture and analyse actions that users perform on the system. The log data is split into sessions, each containing an ordered list of timestamped actions performed by a user in that session. Fig. 1 shows an example of such a session. A probabilistic model was previously built to compute anomaly scores of sessions. We do not have detailed knowledge about the model because it is part of a commercial product by the experts' team. However, we were told that the model is imperfect and in the early stage of its development. The model accuracy has not yet been measured systematically. The model also works as a black-box, lacking detailed account for its scoring mechanism. Therefore, when the anomaly score of a session is high, an investigation into that session is required to *seek clarification and validate the score*. In this paper, we do not aim to explain the scoring mechanism of the model because it is unknown to us. Instead, we provide flexibility for the analysts to explore different aspects of the data to seek a probable explanation that correlates with the score, if any.

Status Quo. Currently, this investigation can only be done manually without the support of built-for-purpose solutions. The analysis usually starts with a *working set* of sessions that are the result of a query, for instance, all sessions from the last 24 hours, which are filtered to keep only those with high scores. These sessions are then displayed in a table
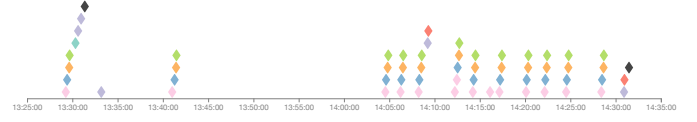
1. DiSIEM: http://www.disiem-project.eu/

Fig. 1. An example of a session. Actions are shown as glyphs along a horizontal axis at when they happen and are colour-coded based on their types. In this session, actions follow a pattern: sequence *pink → blue → orange → green* appears 12 times.

format and the ones with highest score are examined with a series of pie charts showing summary statistics of the most common action types.

This analysis process is limited in two fundamental aspects (discovered during the workshop with analysts, described later in Section 4). First is the lack of informative overviews of sessions. Analysts need to examine hundreds of actions through a generic data table, making it challenging to explore and identify patterns, such as action repetitions or temporal distributions of actions, that provide an in-depth understanding of sessions. Second is the lack of efficient ways to explore and compare sessions. Analysts often need to evaluate a session within the context of all the other sessions of the same user, or in comparison to users with the same organisational role. This can currently only be done through multiple table comparisons which makes the analysis even more challenging. Adding to these limitations is the fact that the above analysis relies entirely on the anomaly scoring model which carries high levels of uncertainty and not safe-guarded for activities that are not yet modelled.

2.2 Data Characteristics

This section describes the characteristics of the data mentioned in the motivating use case. Within the context of this work, we analysed a dataset spanning 31 days on approximately 15,000 sessions performed by 1,400 users with 300 different action types. Each user can perform multiple sessions. Each session comprises an ordered list of actions. The longest session contains 893 actions and each session has 15 actions on average. Specifically, each *session* consists of the following attributes.

- meta-information:
  - *time*: when the session began and ended
  - *user*: who performed the session and their organisational affiliation: *office* and *organisation*
  - *IP address*: through which the session was completed
- actions: an ordered list of what actually happened, including
  - *time*: when an action was performed
  - *type*: providing meaning to the action such as `SearchUser` and `DisplayOneUser`
- anomaly score: an anomaly measurement computed by a model ($0 \rightarrow 1$ with 1 as highly anomalous)

We derive further session attributes from the original ones:

- *length*: the number of actions
- *duration*: the interval between the first and last actions
- *action rate*: the ratio of duration to length, showing the average time gap between two consecutive actions

Here, we list the characteristics of the data that make the analysis and modelling challenging:

C1 – High number of action types. The application in this use case supports many different types of actions. There are 300 unique action types in this sample dataset.

C2 – Varied session lengths. Sessions can have arbitrary lengths ranging from a few to a few hundreds actions.

C3 – Multivariate. Many data attributes from both actions (time and type) and sessions (length, user and IP address) can contribute to the anomaly of a session.

C4 – Complex semantics. Actions have little meaning but they are the execution of higher level semantics such as tasks and user roles.

C5 – Noise. Unintended actions (due to incompetence and mistake) make it challenging to understand user intention.

These challenges and characteristics of the data in this context resonate well with the event sequence data diversity aspects discussed by Plaisant and Shneiderman [5], indicating the generalisably of the problems faced in this context. In fact, an action is a special kind of event – it is an event generated by some actor. Therefore, all research on analysing general event sequences applies, in particular, to action sequences.

3 RELATED WORK

Besides application logs [6], event sequence data in many other domains such as web clickstreams [7], health records [8] and analytic provenance [9], [10] have been studied. For generality, we use *sequence of events* instead of *session of actions* to denote ordered lists of events. For instance, in medical domain, a *patient record* is a sequence of events, consisting of all health events related to the patient. This section reviews approaches and techniques in visualisation analysis and design used for exploring and understanding different types of event sequence data.

3.1 Event Sequence Data Visualisation

Shneiderman [11] likens temporal events to the well-known Anscombes Quartet (1973) and argues that complex patterns within multiple timelines with many event types can only be revealed by effective visual designs and interactions; some of which are discussed here. A conventional approach to visualising temporal events is to display them along a time axis according to their *temporal* information, using icons for discrete events, and bars for continuous ones [12]. To show the *category* or *type* of an event, we can colour code the event glyph [13] or spatially group events according to their (multiple) types [14]. However, both approaches do not scale well with a high number of event types (**C1 – type**).

To handle a large number of events, a common approach is to provide an extra view showing a summary of the whole dataset and allow the selection of a subset of interest for examination in a more detailed view. For example, LifeLines2 [15] uses a stacked bar chart to summarise patient records by their types. Another approach is to visualise aggregates of event sequences rather than individual ones. Depending on the data, sequences can be aggregated into either a tree (visualised as an icicle plot [16]) or a directed acyclic graph (visualised as a matrix [7]). The

former design can reveal common sub-sequences and their volumes more effectively, but the later design has a higher scalability. Alternatively, EventFlow [17] offers interactive simplification by filtering and combining events based on their time and types. In this paper, to handle a large number of sessions, we also include an overview to provide an overall understanding and support examination of selected ones in detail (Section 5). To address the large number of action types, we apply a user-involved clustering algorithm (Section 6.1) and colour code actions based on these clusters.

3.2 Sequential Pattern Mining and Visualisation

A more analytic approach to analyse large datasets is to mine and visualise relevant patterns in the data, revealing higher level semantics (**C4 – complex semantics**). A domain where sequences of actions are mined and modelled is the broad problem area of Process Mining [18] in which trends, behaviour and patterns are derived through mining of the data [19], using sophisticated methods such as transition models and Petri-nets [18]. In this work, given our interest in simplifying sequences, we resort to sequential pattern mining techniques, which are also widely used in process mining. These techniques extract ordered lists of events that co-occur frequently [20], such as common visited paths in a website and health issues that likely to follow in chronological order. The patterns can be mined using an Apriori-based algorithm with a parameter controlling how much "frequent" a pattern should be [21]. Constraints to the patterns can be included such as temporal context and concurrency [22]. Visualisation is then used to explore the mining results, commonly showing each pattern as a sequence of visual glyphs indicating event types [23]. Chronodes [24] focuses on relationships among a small set of patterns of interest, displaying patterns that appear before, between and after the selected ones.

One issue with sequential pattern mining is the large resultant set of similar patterns (e.g., $A \rightarrow B \rightarrow C \rightarrow D$ and $B \rightarrow C \rightarrow D \rightarrow A$), making it challenging to interpret. Dev and Liu [25] propose a ranking technique to prioritise more *cohesive* patterns, i.e., supported by shorter sequences. Liu et al. [26] suggest to exclude patterns if they share similar support sets. However, it is also interesting to investigate further if two different patterns are supported by the same set. Liu et al. [27] extract key events and build a tree of common patterns based on only those events. Wongsuphasawat and Lin [28] extract all n-grams and aggregate infrequent ones having the same prefix, which is similar to maximal sequential patterns [21] that we use. Chen et al. [29] find an optimal set of summarising patterns that minimises the difference between the patterns and the underlying event sequences. In this paper, we apply a classic sequential pattern mining algorithm and propose new constraints to facilitate making sense of the mined patterns (Section 5.1.2, a different set of constraints from Frequence [22]).

3.3 Query-based Event Sequence Data Analysis

To handle large datasets, an alternative to the "overview-first" approach is the "search-first" approach, which can be effective when analysts already know what they want

to investigate. DecisionFlow [30] allows interactive construction of queries, specifying event types and temporal relationships between occurrences of these event types. Also taking a visual query approach but for building regular expressions, such as "$(A|B)C+$", (s|qu)eries [31] enables exploration of complex event sequences matching the expressions, e.g., A or B is followed by one or more C.

A different type of search without explicit query parameters is search by *similarity*. Similan [32] allows finding patients having health records similar to the target. The similarity measurement favours pairs of records with a high number of matched events and a low number of mismatched events (which can be extra, missing or wrong order). In this paper, we also compute a domain specific measure to support comparison tasks. Instead of comparing two event sequences in isolation, we evaluate each event in a sequence within the context of a particular set of sequences (e.g., those performed by the same user) to derive how expected that event is (Section 5.2.2).

4 UNDERSTANDING ANALYSIS GOALS & TASKS

4.1 Methodology

We conducted a series of four workshops with five analysts, who all work in the same company that provided the data and are familiar with the type of investigation mentioned in the motivating use case. Three of them have more than 10 years of experience, and the other two have around 5 years of experience. The goals of the workshops were to gain a deep understanding of the current investigation process and to discuss iterative design prototypes. Each workshop lasted around 2 to 3 hours and involved 3 to 5 analysts. In the first workshop, an analyst demonstrated and explained their current practice in anomaly investigation. We observed the demonstration, asked for clarification, and followed up with a semi-structured interview to gain additional insight. We then agreed with the analysts on an initial set of key leverage points that could improve the existing process, and built prototypes to address them. In subsequent workshops, we demonstrated a prototype showing its capability in revealing potentially interesting insights in the data. The analysts then commented on its usefulness and suggested what information and patterns might be relevant to the investigation that are of their interests. We then made changes to the prototype and prepared for the next iteration.

4.2 Analysis Goals & Tasks

The following goals and tasks are the abstraction of our observations of the difficulties and limitations in the current way that analysts perform their work. We present the tasks using domain-unspecific terms to highlight their transferability to other applications, however, we also list specific examples noted during the workshops that informed these tasks to put them in the context of the application area.

Goal 1 – Overall Understanding. Help analysts gain a multi-perspective and multi-level understanding of the monitored sessions, facilitating the identification of highly unusual ones for further investigation.

Task 1.1 – Multi-perspective exploration of sessions. Currently, the analysts can explore the sessions based on only the anomaly score. It can be biased because the score can be imperfect. Therefore, it is necessary to complement the score with other information, such as duration and length, to increase accuracy in identification of potentially suspicious sessions. Specific examples: *"We know some scores are due to the way the algorithm is developed; thus, we would like to consider these with caution.", "We rank sessions according to the score, but score alone could be misleading."*.

Task 1.2 – High-level semantic summary of action sequences. Besides exploring at the session level, it is useful to have an overall understanding of what actually happened in those sessions before diving into particular ones. Currently, analysts rely on the frequency of action types, which is a semantically poor summary of the content of sessions. Specific examples: *"... want to see different groupings of actions at once ...", "... dont actually know what the end users usually do, so a summary is very helpful ..."*.

Goal 2 – In-Depth Analysis. Help analysts gain deep understanding into particular sessions selected in the previous step. The ultimate purpose of this analysis is to search for an explanation why the score is high or why a session is unusual. Such analysis is currently completed through manual examination of actions displayed in a data table, which is time-consuming and error-prone.

Task 2.1 – Multi-scale exploration. Help analysts explore what happened in a single session at different levels of granularity: high-level abstraction for quick understanding and detailed actions for in-depth inspection. Also, the tool needs to support exploration of multiple sessions such as the ones performed by the same user or executed with a high rate. Specific examples: *"I want to know what the user has done in the past...", "... how his typical session looks like and what common series of actions he performs...", "... also like to compare to what other people in the same office/similar users did..."*.

Task 2.2 – Comparative analysis. Help analysts evaluate a session by comparing it with the past behaviour; i.e., the sessions previously performed by the same user. This is currently the most challenging task in the analysis because of comparison of multiple action tables. The tool needs to quickly reveal both the similarity and difference between a given session and the past sessions. Specific examples: *"I want compare this session to what the user has done in the past...", "... which actions are usual/unusual for this user..."*.

5 DESIGNING A VISUAL ANALYTICS APPROACH

The analysis goals and tasks identified in Section 2 stress the importance of an enhanced understanding of activities of users both at a general (to gain an overview of the overall status of the system), and at a specific level (to gain an indepth understanding on what happens in specific sessions). Aiming these goals within the context of the challenging data characteristics (**C1 – C5** in Section 2.2) leads us to design and develop a visual analytics approach that focuses on providing multi-perspective, semantic summaries of action sequences, and on facilitating the multi-scalar, comparative analysis of sessions.

5.1 Overall Understanding of Sessions

We facilitate the high-level analysis of sessions in the working set (a subset of sessions returned by a query as explained

Fig. 2. Linked visualisations in U4. The Session View (**A**) helps explore relationships between session attributes. Sessions are displayed as small rectangles with colour lightness showing anomaly score, height showing a numerical attribute such as session length, and grouped by a categorical attribute such as user's office. Both sessions and groups can be sorted in different ways to reveal patterns. Sessions in the selected office are displayed in the Activity View (**B**) and the Timeline View (**C**) for further investigation. The Activity View displays common activities (as sequences of actions) that are mined from our constraint-based pattern mining algorithm. The Timeline View displays sessions as small multiples of visual summaries (each row is a session) and grouped by user. All sessions are displayed in relative time (only temporal order is preserved), except for the annotated session. Most common actions in the selected sessions are colour-coded according to their types (see the legend in **D**). Alternatively, all actions in the dataset can be consistently colour-coded based on the result of our semantic distance based action clustering analysis.

in the motivating use case) at two different levels: *session* (i.e., summary of sessions through attributes such as score, length and duration) and *action* (i.e., the atomic events depicting what actually happened).

5.1.1 Multi-perspective Exploration of Sessions

In current practice, analysts select sessions to investigate based solely on anomaly score. This practice is highly error prone due to the inaccuracies in the scoring mechanism, thus requiring a multi-perspective approach that considers both anomaly score and other session attributes (Task 1.1). The Session View (Fig. 2A) provides an overview of sessions in the working set. This view targets the scale of 1000 sessions, which is in line with the number of sessions an analyst usually examines per a single day in our case study.

Visual Representation of a Session. To help analysts have a multi-perspective understanding of sessions, one approach is to provide a visual representation of their multiple attributes (addressing **C3 – multivariate**). For a scope of 1000 sessions, we choose to display each session as a small glyph as it can help spot and select suspicious sessions for further investigation more easily. The glyphs follow a sequential layout: from the top row to the bottom row, and from left to right in each row. It is impractical to show many attributes simultaneously in a small glyph. Therefore, we limit the number of attributes visually encoded and allow changing the encoding–attribute mappings interactively.

Overview of Session Attributes. We classify the attributes into the following classes.

- Core attribute: *anomaly score*.
- Other numerical attributes: *length, duration, action rate*.
- Categorical attributes: *user, office, IP address*.

Anomaly score is the given numerical measurement that plays an important role in identifying sessions of interest; therefore, we assign it to the core attribute and make it always visible. The design also needs to encode another numerical attribute and one categorical attribute.

Core attribute. Several options are available to encode a numerical attribute such as length, angle, area and colour lightness. For the core attribute, we choose a visual channel that can catch attention from the viewers instantly: colour lightness. Black is set as the default hue in our prototype. In practice, analysts may prefer to split the score value based on predefined ranges such as for high, medium and low scores. Therefore, besides the continuous scale, we also provide a quantised scale using several shades of black to imply the order of those classes.

Other Numerical Attributes. Besides position, which is already used for the layout, length is the most effective visual channel in encoding numerical attributes, especially aligned length [33]. Therefore, we choose a small rectangle to represent session glyph and align rectangles at their bottoms, mapping attribute values to aligned rectangle *heights*. All rectangles have the same widths.

Categorical Attributes. Different approaches are avail-

able to visualise categorical or set relationship [14] such as drawing links between items in the same set and colour coding the item glyphs according to their set memberships. However these methods are not suitable for a large number of sets. In our dataset, there are hundreds of users in every single day. A session is performed by only a single user; thus, there is no intersection between user sessions. This allows us to use spatial location to distinguish groups of sessions, following the Gestalt principle of proximity [34]. More specifically, sessions belonging to the same group are located close together in a single row, surrounded by a light border to indicate the group containment and provide a strong separation from other groups (Gestalt principle of uniform connectedness). For example, Fig. 2A shows sessions grouped by office with each group as a grey rectangle containing all of its sessions performed by users in the same office. We choose not to colour the background of the entire group because it adds too much ink to the display and it might be challenging to find a colour that does not interfere with the range of different shades of black in the foreground.

Interactions in the Session Overview. We provide the following three interaction means to support analysts in constructing new perspectives for performing their analyses: *session sorting*, *group sorting*, and *sessions alignment*.

Session Sort. Sessions can be ordered by time and quantitative session attributes, allowing another perspective to be analysed. For example, Fig. 10 shows sessions sorted by action rate, with rectangle height mapped to session length, revealing a relationship between anomaly score and those two attributes.

Group Sort. While analysing groups, one important task is to find the *unusual* ones. However, analysts might have different strategies in formulating what is unusual for groups and consider the anomaly scores of a group's individual sessions in different ways. We provide the following measurements to sort groups:

Median score. This is the median value of the scores of all sessions in each group. We choose *mean* over other central tendency measurements because of its robustness.

Medal-based score. A group of 10 sessions with an 0.9 score could catch more attention than a group of only one session with an 0.95 score. Therefore, we propose a sorting method as in Olympic medal table, prioritising the number of high scores, then medium scores, and then low scores. Note that high/medium/low scores are defined as discussed earlier in the visual encoding of score.

Division of score. Considering another example: a group of 5 sessions with only high scores might be of more interest than a group of sessions with 6 high scores and 94 low scores. Therefore, this sorting method split groups into the following bands with decreasing severity of score: only high → (only high and medium) → (high, medium and low) → (only medium and low) → only low, as in Fig. 2A.

Session Alignment. Another strategy to identify unusual groups is to compare the current behaviour and the past behaviour of the same group. For example, consider these two cases: (i) a user performs a session with 0.8 score and his sessions in the past were also typically high at 0.7 score, (ii) a user performs a session with 0.7 score but his sessions in the past were typically low at 0.2 score. The latter case could be more interesting even with lower absolute

Fig. 3. Sessions are grouped by user office with the rectangle height showing the difference between the session value (of the attribute mapped to the rectangle height) and the average value of the past sessions performed by users in the same office.

score due to the large negative change in behaviour. This sessions alignment can be considered as related to Task 2.2 (comparative evaluation for in-depth analysis) but at an earlier stage, before diving into specific sessions.

Technically, the "typical" score in the past can be measured by the average of scores of sessions that performed prior to the starting time of the working set and by the same group. The "past" window can be constrained to a specific range rather than the entire history so that the window can reflect the latest behaviour. This change of behaviour can also be applied to other numerical attributes. To visualise this change, session rectangles are aligned at the middle of the group rectangle instead of at the bottom. The height of the session rectangle corresponds to the difference between its value and the historical average value. If the difference is positive, the rectangle grows up from the baseline, and vice versa. Fig. 3 illustrates this design.

5.1.2 High-level Semantic Summary of Action Sequences

The techniques in this section aim to provide an overview of sessions at an action level, but with a richer semantic than only the frequency of action types (Task 1.2). Generating high-level abstractions of raw actions also helps simplify the sequences, making it more effective for analysts to gain deep understanding into the inspected sessions (Goal 2).

Choosing High-level Abstractions. Our initial data exploration revealed that actions in a session do not appear randomly. Instead, they often occur together as short sequences of actions that execute a higher level activity. For instance, the sequence `SearchUser → DisplayOneUser → UpdateUserDetails` may represent an activity of "updating details of a user", which could be part of a bigger "user verification" task (see Fig. 1 for another example). We do not mine user tasks in this paper, however consider the extraction of frequent user activities as a step towards dealing with the **C4 – complex semantics** of action sequence data. Note that our consideration of semantics here is based on how the actions are understood in the functional context rather than how they are labelled literally. We model an *activity* as a frequent sequence of actions and discuss the mining algorithm as follows.

Mining Activities. There are many sequential pattern mining algorithms [20] with differences in computational performance and data formats. We choose to implement the classic Generalized Sequential Patterns (GSP) algorithm [35] due to its simplicity. It takes a set of sessions as input and returns frequent patterns (i.e., those that appear more than a minimum *support*, which is defined as a fraction of total sessions supporting a given pattern). The GSP algorithm generates patterns with increasing length and start at one. In each iteration, a set of potentially frequent patterns is

generated using the set of frequent patterns computed in the previous iteration (*candidate generation*). The candidates are tested against minimum support and other constraints to become true patterns (*candidate testing*). The process stops when no frequent patterns are found.

A major limitation of sequential pattern mining is that the number of resultant patterns can be very large and the majority of them may not represent meaningful activities. To reduce the number of irrelevant patterns, we adjust the GSP algorithm with the following constraints. Note that we use $\langle a_1 a_2 \ldots a_n \rangle$ to denote a sequence or a pattern with each a_i is an action type.

Action Aggregation. Consecutive actions in a sequence are combined into one if they have the same type. It is because we assume that successively repeated actions convey similar meaning as a single action does. For instance, a series of `SearchUser` actions and a single `SearchUser` action share a unique goal: finding the right user. This is a pre-processing step prior to running the GSP algorithm. Formally, a sequence $\langle b_1 b_2 \ldots b_m \rangle$ is compressed into $\langle a_1 a_2 \ldots a_n \rangle$ if there exists n pairs of lower and upper (l_i, u_i) index integers $1 = l_1 \leq u_1 < l_2 \leq u_2 < \cdots < l_n \leq u_n = m$ where $l_{i+1} = u_i + 1, \forall i \in [1, n-1]$ such that $b_{l_i} = b_{l_i+1} = \cdots = b_{u_i} = a_i, \forall i \in [1, n]$ and $b_{u_i} \neq b_{l_{i+1}} \forall i \in [1, n-1]$. Each merged action a_i is a sub-sequence of original actions, thus having a time range instead of a time point: start-time(a_i) = time(b_{l_i}) and end-time(a_i) = time(b_{u_i}), $\forall i \in [1, n]$.

Unique Actions. We assume that an activity only contain actions with different types. Therefore, in a pattern $\langle p_1 p_2 \ldots p_k \rangle$, $p_i \neq p_j, \forall i \neq j \in [1, k]$. This constraint is applied in the candidate generation procedure.

Consecutive Actions. A sequence $\langle a_1 a_2 \ldots a_n \rangle$ supports a pattern $\langle p_1 p_2 \ldots p_k \rangle$ if there exists an integer i such that $a_i = p_1, a_{i+1} = p_2, \ldots a_{i+k} = p_k$. We apply this constraint because we focus on consecutive, short sequences of actions. This constraint also simplifies the implementation of candidate generation and candidate testing.
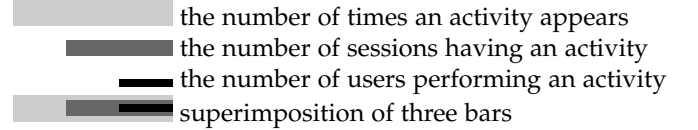
Time Gap. An activity reflects a small unit of intention and is likely to happen within a short amount of time. This constraint limits the maximum time gap between every two adjacent actions in a supporting sub-sequence. If a sub-sequence $\langle a_1 a_2 \ldots a_k \rangle$ supports a pattern $\langle p_1 p_2 \ldots p_k \rangle$ then start-time(a_{i+1}) − end-time$(a_i) \leq$ *time-gap*, $\forall i \in [1, k-1]$. This constraint is applied in the candidate testing procedure.

Maximal Patterns. A pattern is maximal if it is not contained in any other patterns [21]. We only consider maximal patterns because if a pattern is frequent, its sub-patterns are also frequent. For instance, if `SearchUser` $\rightarrow$ `DisplayOneUser` $\rightarrow$ `UpdateUserDetails` is an activity (to update user details), it is unnecessary to consider `SearchUser` $\rightarrow$ `DisplayOneUser` as another activity.

Acyclic Patterns. Sequential pattern mining algorithms return cyclic copies of the unique pattern when a pattern is repeated consecutively. For instance, in this sequence $ABCABCABCABC$, besides the 4-time pattern ABC, two other patterns, BCA and CAB, are also detected as frequent with 3 times. We check all cyclic patterns and only remain the one with the highest support. Both maximal and acyclic constraints are applied as a post-processing step, filtering patterns mined from the GSP algorithm.

Sequential pattern mining algorithms are sensitive to parameters including support and time gap. Together with domain analysts, we explored the effect of parameter values on the resultant patterns, through a simple graphical interface. For the dataset used in the testing, setting support as 3% and time gap as 60 seconds yields a meaningful and manageable set of frequent patterns. The analysts confirmed that these patterns represent common series of actions that users would perform in their work.

Visualising Activities. The Activity View (Fig. 2B) shows the most common activities produced by the mining process, helping analysts gain an overall understanding of the sessions. The visualisation consists of multiple rows, each for an activity and is split into two parts. The right part shows the sequence of actions in an activity as contiguous colour-coded squares such as ▮▮▮▮, each for an action type. This colouring method can only show a few distinct action types such as the most common actions in the selected sessions. We will discuss our approach to address this limitation in Section 6. This representation enables determining the meaning of an activity through its actions. The left part shows how "frequent" an activity is through three-level support values, each is shown as a grey bar with length representing the value. These bars are then superimposed one on another. Because #times $\geq$ #sessions $\geq$ #users, i.e., the darker bars are always shorter than or equal to the lighter bars, we superimpose the bars to show this property more clearly.

the number of times an activity appears
the number of sessions having an activity
the number of users performing an activity
superimposition of three bars

5.2 In-Depth Analysis of Unusual Sessions

This sections discusses support for in-depth analysis of unusual sessions that are discovered in the previous stage (Goal 2). When sessions are identified in the Session View (Fig. 2A), they are made available in the Timeline View (Fig. 2C) with the following features for further analysis.

5.2.1 Multi-scale Exploration

In the following, we first discuss the exploration of a single session and how the approaches are then extended to multiple sessions.

Single Session. To help analysts gain understanding of a single session, we provide a visual summary of its actions. Both the time and action type greatly contribute to the understanding of the session, thus being included in the visualisation. Actions are represented as coloured rectangles (consistent with the visual representation of an activity) and are displayed sequentially along a horizontal time axis based on their chronological order. We then provide visual representations of actions at four levels of detail to enable analysts to examine a session with different purposes: a high level summary or a detailed examination. This also helps address the spatial scalability when a session contains a large number of actions (**C2 – length**).

Action. At the highest level of detail, actions are shown separately (Fig. 4a), providing a sense of the session length

(a) Each action is shown individually as a rectangle.

(b) Each action is shown individually as rectangle and consecutively repeated actions are aggregated.

(c) Activities are shown instead of individual actions.

(d) Consecutively repeated activities are aggregated.

| 00:45:00 | 01:00:00 | 01:15:00 | 01:30:00 | 01:45:00 | 02:00:00 | 02:15:00 | 02:30:00 | 02:45:00 | 03:00:00 | 03:15:00 | 03:30:00 | 03:45:00 | 04:00:00 |

(e) The same activities in Fig. 4d are shown at absolute temporal values. A black dot indicates when the first action in an activity starts.
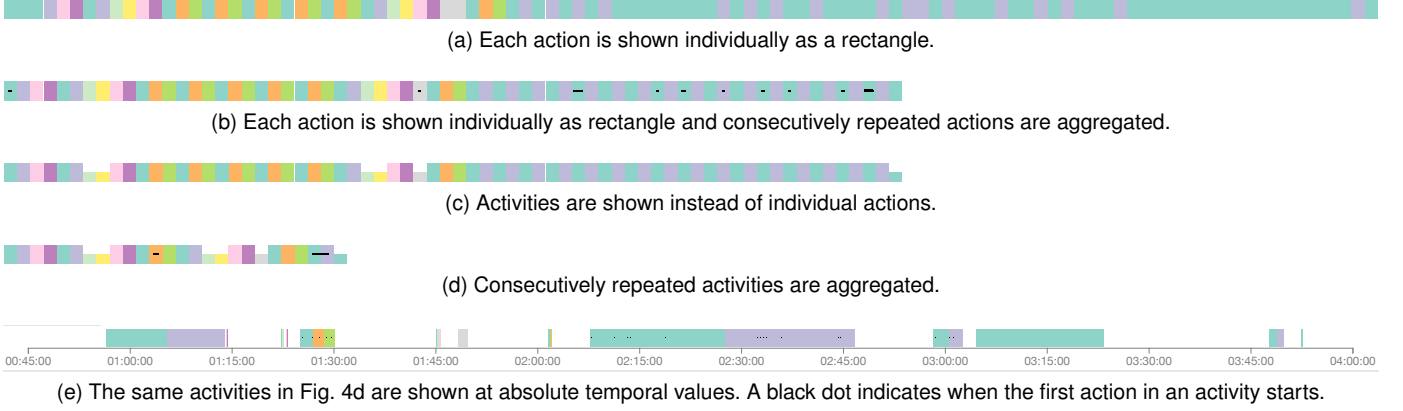
Fig. 4. Different representations of a session. A black horizontal line in an action or activity indicates the size of an aggregate.

(in terms of the number of actions) and facilitating detailed examination of individual actions.

Action Aggregate. At the second highest level, consecutive actions having the same type are combined into one with a superimposed subtle horizontal line indicating the size of the aggregate (Fig. 4b). This aggregation is based on an assumption that consecutive actions carry the same meaning as a single one, such as searching for a user many times until finding the right one.

Activity. At the third level, actions are replaced by mined activities whenever possible (Fig. 4c). This simplifies the visual summary and helps understand the entire session faster. An activity is represented as a contiguous block of colour-coded rectangles, leaving no padding in between them. This representation is consistent with the Activity View. The height of non-activity actions is reduced to half to distinguish them with activities. Alternatively, a border can be added to an activity, but the visualisation could be a bit busy with a lot of activities such as the one shown in Fig. 2. Also, as actions/activities are placed sequentially with a small gap (one or a few pixels), it can be a bit confused when an action is placed between two activities.

Activity Aggregate. At the lowest level of detail, consecutively repeated activities are combined into one with a superimposed subtle horizontal line indicating the size of the aggregate, similar to the *action aggregate* level (Fig. 4d). This most compact representation provides an effective way to visually identify "noise" in a session. In Fig. 4d, infrequent actions "pop out" from other frequent activities and can visually attract analysts for further investigation.

We also provide a layout option to position actions horizontally proportional to their temporal values. This may cause visual clutter when actions happen close together but could reveal interesting temporal patterns such as time gap between activities (Fig. 4e). Another example is the temporal regularity discovered in the user evaluation we describe later (Fig. 11 in Section 7).

Multiple Sessions. Analysts often need to examine multiple sessions simultaneously. They can be sessions having similar attributes discovered in the previous stage (Overall Understanding) such as those having high scores and coming from the same IP address. Currently, this task is time-consuming and error-prone due to the lack of visual support (analysts have to examine multiple tables of session actions).

We support this exploration by providing small multiples of visual summaries of sessions, each shown as a separate row (Fig. 2C). The representation of each session is the same as the representation of a single session discussed earlier.

Fig. 2C shows sessions performed by users in the selected office and are grouped by user. To the left of the user name is a simplified box plot, showing the median and the interquartile range of the scores of all sessions performed by the user in the past. It provides a historical context of how a typical session score would look like. Having multiple compact visual summaries of sessions allows comparison of sessions performed by the same user and across users. User *Agent Cheesecake* seems to be a help-desk user doing a lot of "unlock" and "reset password" activities. Whereas, users *Jolt* and *Jack of Hearts* deal with "right management" activities. Finally, sessions performed by *Hall, Franklin* and *Geirrodur* involve heavily with "delete user" and receive quite high scores.

5.2.2 Comparative Analysis

One essential step in anomaly investigation is to compare what a user does in a given session against what he or she did in previous sessions. Currently, this manual comparison is time-consuming and error-prone because a user can perform thousands of sessions in the past. Therefore, it is necessary to automate this comparison task (Task 2.2).

We formulate the problem as follows. First, all actions happening in a given session s are extracted into a set $C = \{a\}$ with c_a be the number of times action a happens in that session. Then, all actions that the user did in his past sessions, prior to s, are also extracted into a set $P = \{a\}$ with p_a be the number of times action a happened in each session on average. The goal is to derive how *expected* each action a in C is by comparing c_a and p_a. Intuitively, action a is considered to be expected in session s if it happens the same number of times as it happened in each session that the user did before; i.e., $c_a = p_a$. Also, if action a happens, for instance, five times more or less than the number of times it happened in the past, we assume that these two cases have the same amount of *expectedness*. Therefore, we compute the expectedness e of action a in session s as follows.

$$
e(a, s) = \begin{cases} \frac{\min(c_a, p_a)}{\max(c_a, p_a)}, & p_a > 0 \\ \frac{\min(c_a, p_a)+1}{\max(c_a, p_a)+1} = \frac{1}{c_a+1}, & p_a = 0 \end{cases}.
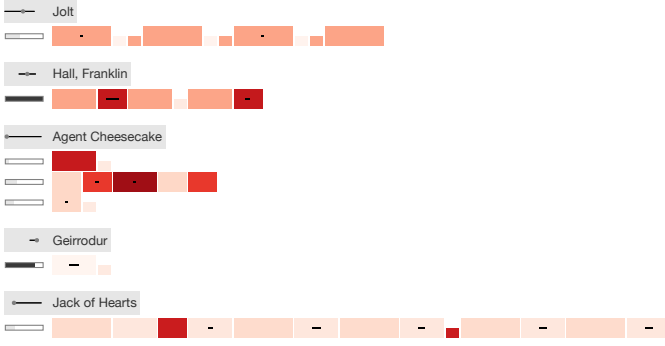$$

Fig. 5. Comparative analysis between each session and its past sessions performed by the same user. This shows the same sessions as in Fig. 2C. Dark red rectangles indicate highly unexpected activities.

Fig. 6. Colour Map used in our design.

The expectedness value ranges in $(0, 1]$ with 0 indicating totally unexpected and 1 indicating totally expected. This value can be used to colour code actions in the timeline to provide a quick assessment of how unexpected the actions in the inspecting sessions are (Fig. 5). A single colour hue (red) is used and the lightness of the colour shade maps to the expectedness score (the darker the more unexpected) to emphasise unexpected actions. We provide this colouring (for comparing with the past) as an option besides the action type colouring (for understanding what is happening in a session) so that the analyst can switch between two purposes of use.

The above formula uses action frequency (c_a), thus being sensitive with the session length. For example, in the past, a user performed `SearchUser` 10 times per session. Now, in a given session, he performs that action 20 times because the session is twice longer. Therefore, this session is still very expected. To address it, we use action ratio instead, replacing c_a with $\frac{c_a}{|s|}$ with $|s|$ be the number of actions in session s. Note that in this section, we discuss how to compute expectedness of actions. However, the formula also applies for activities.

6 FURTHER CONSIDERATION ON ANALYSIS, DESIGN AND INTERACTION

This section discusses further analytical and interactive capabilities implemented, and additional design considerations made within the design and development process.

6.1 Semantic Distance Based Action Clustering

As discussed above, the data in consideration has 300 distinct action types. This is not only causing a problem in terms of colouring the actions within action sequences but also a challenge for analysts when they try to reconstruct an understanding of the sessions. Hence, we cluster action types under "semantically" relevant groups to address this challenge (**C1 – type**).

To accomplish this we adopt the *semantic distance* approach we developed in our recent work [4]. This measure captures the semantic relatedness within actions by borrowing ideas from a text-mining approach on the lexical co-occurrence analysis of words within a corpus of
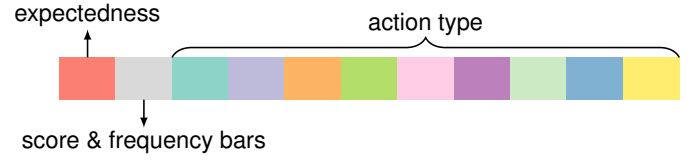
text [36]. The core idea of this measure is that frequently co-occurring actions are semantically more related than actions co-occurring rarely. The resulting *semantic distance*s serve as heuristic measures to capture semantic relatedness and suitable metrics to incorporate in a clustering algorithm.

Having established this notion of distance between individual actions, we then utilise these distances as the underlying distance measure within the density based clustering algorithm OPTICS [37]. We first generate a matrix of pairwise (semantic) distances between actions and feed this matrix to the algorithm. We then used a progressive clustering approach [38] with an adaptive cluster radius to handle the high level of variation within frequencies of actions. As a result of this process all the actions are aggregated into 36 groups. Using these 36 groups as a starting point, a domain expert with solid knowledge in the domain examines the automatic grouping and further reduces them to 8 semantically core groups. Such a user-involved approach not only enables us to arrive at domain-relevant clusters but also ensures faster adoption and increased engagement on the users' side. Next section discusses how we adopted these resulting *action type groups* as part of the colour palette we use within our solution.

6.2 Colour Map Design

Colour is used extensively in all three views to represent the following concepts.

- anomaly score (Session View, Timeline View)
- frequency bars (Activity View)
- action type (Activity View, Timeline View)
- expectedness score (Timeline View)

Our colour map is based on an 11-class Set 3 of quantitative colours from ColorBrewer [39]. #D9D9D9 is used for both anomaly score and frequency bars. We believe that there is no misunderstanding between these two uses because they are in different views, which was also confirmed in the user evaluation. #FB8072 is used for expectedness score. This leaves the rest nine colours for action types, which is about the number of colours that human can visually distinguish simultaneously [33]. By selecting colours from a single categorical colour set, we limit the chance that the colours encoding score, frequency and expectedness will be perceived similarly to action types, which may confuse the end users. Fig. 6 illustrates this colour map design.

6.3 Multi-semantic Linking

Conventionally, when elements in one view are selected, the *same information* in any other linked view is also highlighted [40]. We propose a novel *multi-semantic* coordination concept that relaxes the linking targets: elements in
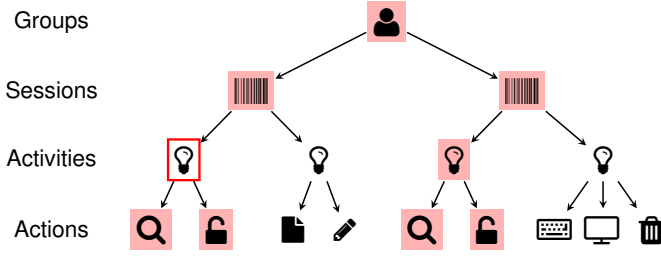
Fig. 7. An example of multi-semantic linking across four semantic levels. When an activity is mouse hovered (red border), all the same activities, sessions and users containing them, and individual actions within the activity are highlighted (pink backgrounds).

(a) Activities in the timeline with colour encoding action types.

(b) Most frequent activities performed in the past by the same user. Manual comparison with such diverse activities is time-consuming.

(c) Activities in the timeline with colour encoding expectedness, facilitating automatic comparison.

Fig. 8. Example of the *comparative analysis* usage pattern.

views are linked when they are *semantically related* to the source, rather than forcing to be the same. In our case, four different semantic levels are included in the views: *group*, *session* (Session View), *activity* (Activity View), and *action* (Timeline View), as shown in Fig. 7. A group consists of multiple related sessions (such as the ones performed by the same user), a session contains an ordered list of activities and an activity is a sequence of actions. When a semantic concept in one view is selected, the other three hierarchical semantic concepts are also highlighted in other views. Fig. 7 shows an example when activity `SearchUser` → `UnlockUser` in Activity View is mouse hovered, sessions and users (higher levels) having that activity are highlighted in Session View. The same activities (equal level) or its individual actions (lower level) in Timeline View are also highlighted. Such multi-semantic linking enables analysts to effectively explore relationship between related concepts in different views.

7 EVALUATION

7.1 Evaluation Design

As part of our user-centred design process, we conducted a user evaluation of the **U4** tool to establish an understanding of its use by target users in performing real tasks for which it was designed to support. We recruited six participants who are all working with the company mentioned in the motivating use case. They had different levels of both domain (the application from which the dataset was collected) and technical (the anomaly modelling score) knowledge. Three of them had been involved in the design workshops, thus having a better understanding about the **U4** tool.

The participants were all introduced to the tool's features in about 45 minutes before being given the task. The task was to use **U4** to first identify sessions of interest, and then investigate them in depth. We used the same dataset mentioned in the motivating use case but selected a 24-hour window different from the introduction session for the working set of sessions to avoid any learning effect. This dataset acts as a representative of a wider range of datasets and applications.

These six participants were split into two groups, each carried out the task separately. We balanced the two groups in terms of domain and technical knowledge, as well as their experience with the tool. Each group spent about one hour and a half to complete the task. During a session, one group member directly interacted with the tool but all members
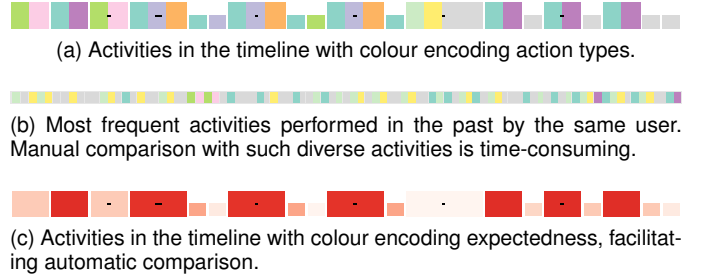
contributed and discussed out loud their thoughts on both the investigation tactics and the tool's interface. At the end of each session, we conducted a semi-structured interview with the participants to further understand their experiences with the tool and to elicit suggestions for improvement.

7.2 Examples of Use

Although the evaluation was of small scale, we observed interesting insights in both analysis sessions. Four notable usage patterns were identified to be repeated several times during the evaluation. In this section, we present one example for each pattern.

Comparative Analysis. One important strategy in anomaly investigation is to compare what a user did in the inspected session against what he did in his previous sessions (**Task 2.2**). Such comparison was observed many times during the evaluation; one example is described as follows. An analyst selected a relevant session in the Session View for further investigation. That session was then displayed in the Timeline View and another analyst quickly started reading out loud each action the user performed there. Realising the number of actions was quite high, she suggested to increase the aggregation level of actions so that she could quickly examine what happened there (Fig. 8a). This is a part of **Task 2.1**, which allowed the analyst to explore the session at different levels of granularity effectively.

After understanding what the user did in that session, the analyst asked "*So, what did that user do in the past?*". He wanted to compare the activities in the inspecting session with the activities that the same user performed in his or her previous sessions. Then, the analyst clicked on the user name to see the historical activities in detail and was overwhelmed by the large number of them (Fig. 8b). The group agreed that it could be very time-consuming to do such comparison manually. The analyst then turned on the *comparison mode* to let the timeline highlight the similarity and difference between the two sets of activities. He was drawn onto the notable red rectangles (Fig. 8c) and quickly discovered those rectangles represent frequent activities that appeared many times in the current session but never happened in the past (or much less frequent). The group commented that could explain why the score was high.

User Role Inference. One analyst expressed that she wanted to find some "interesting" users. She then grouped sessions by user in the Session View and selected the user with the most sessions. The Activity View showed that the only common activity in those sessions was `SearchUser` → `UnLockUser` (Fig. 9). This illustrates **Task 1.2**, which enabled
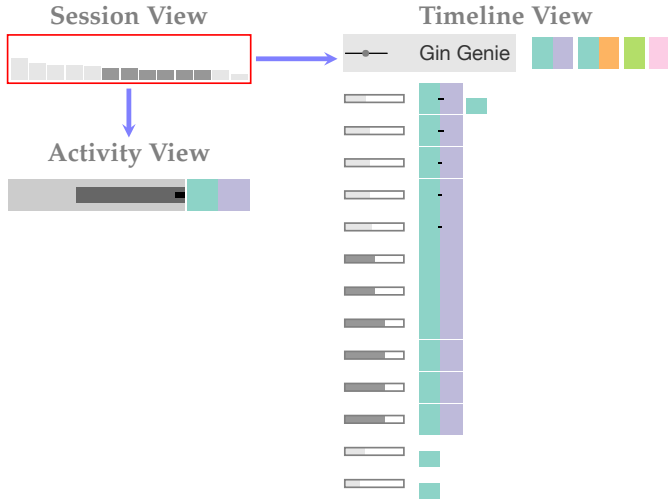
Fig. 9. Example of the *role inference* usage pattern. The Activity View quickly revealed the dominant activity and the Timeline View displayed the past activities (next to the user name) for comparison and inference.
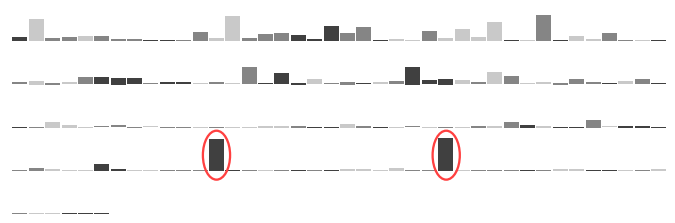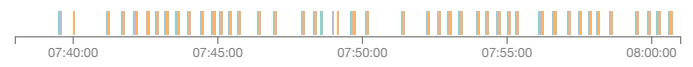

Fig. 10. The Session View configured to answer the question from the analyst. Sessions were sorted decreasingly by *action rate* and height was mapped to *length*. This made tall, black rectangles appeared at the end of the view (highlighted with red border) the most relevant ones because they represent unusual (black) sessions with many actions (high) that were performed rapidly (appeared at the end).

(a) Aggregate of consecutively repeated activities with relative time showing a strong repetitive pattern of a single sequence.

(b) Individual actions with absolute time showing a quite regular action execution pattern.

Fig. 11. Timeline of a unusual session.

the analyst to quickly understand the main activities of the user before exploring his/her sessions in detail. In this extreme case, the Timeline View also strongly supported this observation. The comparison mode revealed that the user also did this "unlocking" activity many times in the past, which might explain why all the session scores were quite low. She then examined the user's past activities and explained that she trusted the automatic comparison but still wanted to know what the user commonly did in order to learn about his/her role. "*It's good to have the past activities on demand. Looking at them, I can say that he is a help-desk guy, answering phones and helping people unlock their accounts.*". This role inference helped her make a more informed decision on the true statuses of the inspected sessions. This example also demonstrated how the tool supported the analyst to explore multiple sessions (**Task 2.1**) through a meaningful yet compact representation of sessions.

Tactic-Driven Analysis. One analyst started with a clear statement about what he wanted to focus. "*I want to look at sequences of actions that are done in a very short time. I guess some actions cannot be done by a human at a very high rate because you need to think when you do your work.*". He reconfigured the Session View towards this goal: skipped grouping because it was out of his interest, and mapped the rectangle height to the *action rate* attribute. The analyst also sorted sessions by the action rate so that he could easily select several nearby sessions (using rectangular brushing) with the fastest rate for examination. He then discovered that those sessions only contained a few actions, which made the action rate less meaningful. Therefore, he put session length into perspective by assigning it to the rectangle height as shown in Fig. 10. This analysis illustrated how the tool facilitated exploration of sessions with multiple attributes (action rate, session length) in addition to the anomaly score (**Task 1.1**).

The analyst then selected a tall, black rectangle with a fast rate (score = 0.87, length = 126 actions, and rate = 10 seconds per action) to investigate its corresponding session further. The Timeline View showed a very strong pattern in this session: it consisted of only a single sequence of actions (SearchUser → DisplayOneUser → UpdateUserDetails) but consecutively repeated 42 times (Fig. 11a). He then examined when these sequences actually happened using the *absolute* time mode and even at the individual action level (Fig. 11b). Immediately, another group member commented "*Doesn't look like a human, too regular*". Moreover, using the comparison feature, the analyst figured out that the user had not done this activity in the past, which made that session even more suspicious. "*Well, this guy hadn't done this Update Details activity in the past, and now he did that 42 times consecutively and regularly within only 20 minutes. I really want to see what exactly he updated.*". Unfortunately, such detailed information was not available at the moment, and we reported this lack of data to the data provider partner so that they can consider capturing it in the future.

Unexplainable Scores in Simple Sessions. Both groups discovered several sessions that they were unable to explain why those sessions received such high scores. For example, there was a session with a single SearchUser action and its score was 1. The comparison feature revealed that the user had done this search action in the past, and the absolute timeline showed that the action was done at normal working hour as well. A technical member in the group who roughly knew how the score was computed said that he was surprised and that score could be a false positive. He commented that it would be useful for the modelling team to be aware of those cases so that they could improve the model. **U4** can be easily customised to explore those sessions. In Fig. 12, the Session View shows sessions with scores above 0.9, ordered decreasingly by length, and the last six sessions are selected to be examined in the Timeline View. All of them might be false positives because their unexpectedness scores are very low. It is also straightforward for us to programmatically filter those high-score sessions with just a few actions but having high unexpectedness.
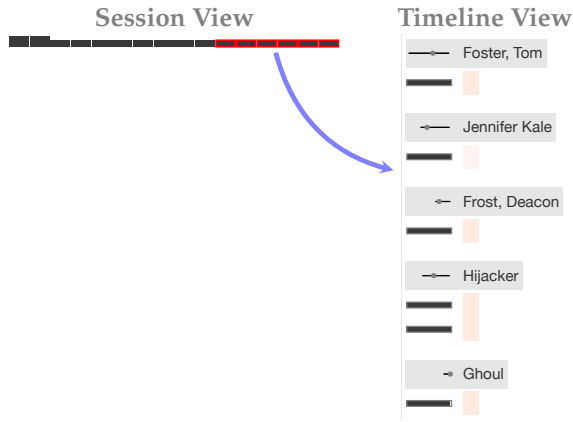
Fig. 12. Identification and examination of sessions with potential false positive scores. Session with high scores and a few actions are selected from the Session View and displayed in the Timeline View with comparison mode enabled.

7.3 Observations & Takeaways

We present the limitations of the tool **U4** that we observed during the evaluation and discuss takeaway messages.

7.3.1 Observed Limitations

Unused Interactive Features. Two features in the Session View were not used in the evaluation: group sort and session alignment. *Group sort* is designed to help analysts quickly identify the most unusual group by reordering groups based on different measurements. *Session alignment* is also designed to help detect unusual groups but by comparison of current and past behaviours. In the interview, the participants expressed their awareness of the two features but felt that the multi-perspective visual representation of sessions were sufficient to drive their analysis. Also, because the evaluation time was relatively short (one hour and a half), they wanted to focus on the analysis using a reasonable amount of knowledge they quickly learned from the tool instead of spending time to try all of its features.

We even observed a case where the analysis could benefit from the groups sort feature but the analysts did not make use of it. One analyst said "*I choose this IP address [sessions are grouped by IP address] because it seems to have the reddest sessions*". However, he made that decision based on a visual scan of the Session View instead of sorting groups using the *medal-based score* method and picking the first sorted one. Interestingly, it might imply the effectiveness of our design of the Session View at a high targeted scale (1000 sessions and 1000 groups). The visual representation allows users to identify various interesting patterns without the need of interaction such as changing group ordering.

To encourage the use of these interactive features where appropriate, we could make the text describing those features more meaningful and domain-specific to the end users. The analysts may not care about how technically the groups of sessions are ordered because it maybe unclear to them why such an ordering is important. For example, the sorting method text such as "medal-based score" could be replaced by a text describing a meaningful pattern it can reveal or a domain-specific question it can help to answer.

Steep Learning Curve. The *comparative analysis* feature required a long explanation before the participants could understand thoroughly. At the beginning, a few participants were confused when they saw the action colours being changed while the comparison feature was toggled. We explained that the comparison mode does not focus on the content of the actions but whether the actions has happened before. Therefore, we colour coded the expectedness instead of action type. The participants seemed to understand and trust this feature more after we explained roughly how the expectedness was computed. Once the participants understood, this comparison feature was used heavily.

Using the same visual channel (colour) to represent two different concepts (action type and expectedness) might not be a good idea, even when the two colour maps are distinct and there is a button to help the analysts to explicitly switch between the two representations. Other visual channels can be considered to represent the expectedness such as height. Currently, the height of visual items is used for distinguishing actions and activities. It can be used to map to the expectedness with positive value displayed above the baseline 0 and negative value displayed below the baseline 0 like . However, the baseline might not be seen easily when the absolute value of expectedness is small or when all values in a session have the same sign. Also, for multiple sessions, it becomes more difficult to quickly reveal the pattern, requiring a scan for each session. Alternatively, auxiliary lines can be added underneath the action/activity representation to indicate unexpectedness with line darkness representing the absolute value. We plan to investigate this problem further in future work.

7.3.2 Key Takeaways from the Evaluation

Putting a Session in Context. Comparative analysis, in particular, comparing a particular session to the past behaviour of the user, is a highly critical task. This was one of the tasks that our approach supported; however, we also observed that there is definitely room for improvement here and visualisations of user model and comprehensive aggregations of past behaviour are needed.

Understanding User Role. One of the key aspects to help decision-making is the inference of user roles within the organisation that the data comes from. Even though we have not explicitly designed solutions to surface the roles of users, analysts conceptualised their observations in that regard. This is a pointer for further work to develop specific views/techniques to identify user roles.

Variety in Metrics. Analysts made good use of the various session metrics to tailor the analysis along their interests. This is contrary to our reservations that the abundance of choices could overwhelm the analysts. As long as the options are designed carefully and informed by the domain needs, analysts are open to adopt them in their analysis.

Over-Engineered Solutions. Some features were not adopted by the analysts, such as grouping and alignment. Contrary to the observation where analysts made use of the extensive set of metrics, these features were less popular. One potential reason is that their existing analysis methods do not incorporate such approach, hence they prioritised

methods that resonate better with the current practice. To fully comment on the use of such approaches, prolonged studies needs to be conducted to observe for adoption.

Living with Imperfect Algorithms. One interesting observation made was how the analysts coped with the problems in algorithmic inaccuracies. In several instances, they noticed sessions with very suspicious scores, had a look at them briefly in the visualisations (or not) and immediately discarded them making a reference that they are due to limitations in the model that they are aware of, e.g., sensitivity towards short sessions (*false positives*). This is an interesting issue – over time this leads to a loss in trust in the algorithmic results and increases the chances of missing relevant cases. On the other hand, we think that by allowing multiple attribute exploration, our tool has the capability in supporting the analysts to spot *false negatives* to a certain degree. For example, examination of long and fast rate sessions as in Section 7.2 – Tactic-Driven Analysis.

8 REFLECTION & DISCUSSION

We reflect on the entire research process, covering requirement analysis, visual and algorithm design, and evaluation.

Iterative Requirement Elicitation. The iterative workshops showed the effectiveness in successive requirement elicitation sessions since it is infeasible for analysts to generate a comprehensive requirement set for such a complex problem in one single session. The iteration allows analysts to gradually understand how a visualisation solution can contribute to their workflow before making appropriate suggestions. The increasing exposure to and knowledge in visualisation through the iterations help analysts develop more valuable suggestions. In the first iteration, they tended to require duplication of existing features in their system, which is not necessarily a core capability in a visualisation tool, e.g., searching a session by its ID. However, in further iterations, they gave suggestions on more complex tasks, such as highlighting the similarity and difference between what a user does in a specific session and what that user has done in the past. Such *better-informed* requirements are essential in bringing forward what visualisation can do best.

Designs from Earlier Iterations. Before finalising on the current design of the tool `U4`, we have made different attempts to explore the data and the design space. We briefly mention those earlier designs here and refer the readers to the supplemental material for more detail. For the Session View (as in Fig. 2A), we have explored alternative layouts, coloured background to separate groups, different hues for low/medium/high scores, and shapes for approximating a numerical attribute. The final design turns out to be the simplest and cleanest, giving just enough information to the end users. We have designed a compact view to reveal the *following* or *trailing* relationship between actions within a session. For instance, how likely sequence ABC will be followed by D (becoming ABCD) or E (becoming ABCE)? This visual evidence convinced us to apply sequential pattern mining to extract patterns. To gain understanding into the mining results, we have started with a scatter plot, showing the three frequency values (occurrences, sessions, users) using two axes and circle size. It was helpful in exploring the relationship of patterns in terms of frequency. However, it was difficult to interpret the meaning of patterns as they overlap each other in the scatter plot.

Design Scalability. We discuss the scalability of all views.

Session View. This view is not designed to show the entire dataset. Instead, it targets a relatively small working set, up to 1000 sessions, which is a subset of sessions returned by a query as explained in the motivating use case. Sessions can be grouped, such as by user, and the groups are separated efficiently using space and border. Hence, our view can handle a large number of users. To handle more than a thousand of sessions, an aggregated view such as by user would be a useful starting point.

Activity View. This view enables the analysts to gain an overall understanding of a given set of sessions of interest by showing the most frequent patterns mined from them. Therefore, we think that displaying a few tens of patterns might be sufficient to achieve this purpose. Colour hues are used to encode events, which is not effective with a large number of event types. We address this by either encoding event groups or the most common event types.

Timeline View. There is enough horizontal space for about a hundred of individual actions. In practice, the view can handle much longer sessions because actions are normally repeated and aggregated. Fig. 4 shows such an example. Vertically, the view can show about ten sessions before requiring scrolling. Sessions shown in the timeline are a small number of related ones selected from the Session View, such as sessions performed by a particular user.

Uncertainty in Sequential Pattern Mining. Similar to most algorithms, our pattern mining approach also relies on effective parameters being set, i.e., the resultant patterns from the mining process are dependent on *support* and *time gap*. Our goal is to identify all important and meaningful patterns. Relaxing the mining constraints helps extract all relevant patterns but introduces many irrelevant ones (false positives). Whereas, tighter constraints could miss relevant patterns (false negatives). In this paper, we take a manual approach: tweaking parameters together with domain experts to reach a reasonable set of patterns. An automatic approach is an important future work, aiming to answer the following questions. How to reduce the number of patterns without missing important ones? How to measure and visualise the strength and importance of patterns?

Black-box Model. The model that the data provider company used to compute anomaly scores works as a black-box. Neither detailed information about the model mechanism nor explanation for the scoring output is provided to us. Therefore, our approach is not dependent on the anomaly score. Instead, we consider the score as just one of the data attributes we support the analysts to explore (Section 5.1.1). This makes our tool more general and transferable to other applications. Currently, the score is treated as a numerical attribute and encoded using colour lightness. If score is unavailable in another event sequence dataset, one can easily replace it with another data attribute.

Reliance on Metrics. In our approach, we use synthetic metrics to rank and compare sessions. On the one hand, strong reliance on such metrics may contradict the idea of discovering unexpected patterns purely visually. On the

other hand, it is hard to avoid tradeoff when working with large datasets in visualisation – without effective heuristics, it is often not possible to simplify and/or structure visual representations when data sets are large in volume. Our approach here is to present analysts with a selection of carefully designed heuristic measures that can help search/filter the sessions and carry out comparisons between them. We support the comparison tasks of visual representations that rely on perceptual processing (e.g., of individual sessions). Our hypothesis is that this will still enable unexpected pattern observations (that can then be formulated as further heuristic metrics in future revisions of such solutions).

Group Evaluation. Conducting the study with a group of participants instead of individuals showed benefits. We observed that the participants did not feel pressure and were confident in performing analysis. They did not feel they were being tested, which might happen in studies with individuals according to our prior experience. Also, the participants contributed actively in the discussions within the group in investigation strategy and how they could implement it using the tool's interface. Moreover, a group had "more eyes" to identify interesting patterns observed in the tool. We observed that conducting the study with groups of participants was effective for the purpose of understanding how the tool could be used within the deployment setting.

9 CONCLUSION

This paper presents a visual analytics approach to enable analysts to gain deep understanding into both the expected and unexpected user behaviours through an analysis of their action sequences. The approach includes novel visual designs and interaction techniques, combined with a data mining algorithm to provide analysts a multi-level overview of user sessions and ways to conduct in-depth multi-faceted comparative investigations of sessions of interest. An evaluation with cyber security experts shows the usefulness of the approach and the tool **U4**, enabling them to execute their analysis strategies and to perform essential analysis operations such as comparison and user role inference.

Even though we are motivated by a use case of "user actions", the presented approach and the techniques are not restricted to either "user/human" or "actions". They have the potential to be applied to a more general class of "event sequences" data that can be split into "sessions" such as "medical records of a patient" or "movement of cows between holdings". For future work, we plan to address the limitations identified in the evaluation regarding the unused interactive features and the steep learning curve as part of the deployment phase of the ongoing project in the form of a longitudinal study. We also plan to improve the quality of our sequential pattern mining algorithm to reduce the size of output without missing relevant patterns. Designing the layout for the Session View to display the absolute time of sessions could reveal interesting patterns as well.

REFERENCES

[1] G. I. Webb, M. J. Pazzani, and D. Billsus, "Machine learning for user modeling," *User modeling and user-adapted interaction*, vol. 11, no. 1-2, pp. 19–29, 2001.

[2] T. Fawcett and F. Provost, "Activity monitoring: Noticing interesting changes in behavior," in *ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 53–62.

[3] F. L. Greitzer and R. E. Hohimer, "Modeling human behavior to anticipate insider attacks," *Journal of Strategic Security*, vol. 4, no. 2, p. 25, 2011.

[4] P. H. Nguyen, C. Turkay, G. Andrienko, N. Andrienko, and O. Thonnard, "A Visual Analytics Approach for User Behaviour Understanding through Action Sequence Analysis," in *EuroVis Workshop on Visual Analytics*. The Eurographics Association, 2017.

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

[6] P. H. Nguyen, K. Xu, A. Wheat, B. W. Wong, S. Attfield, and B. Fields, "Sensepath: Understanding the sensemaking process through analytic provenance," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 41–50, 2016.

[7] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson, "Matrixwave: Visual comparison of event sequence data," in *Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 259–268.

[8] A. Rind, T. D. Wang, W. Aigner, S. Miksch, K. Wongsuphasawat, C. Plaisant, B. Shneiderman *et al.*, "Interactive information visualization to explore and query electronic health records," *Foundations and Trends® in Human–Computer Interaction*, vol. 5, no. 3, pp. 207–298, 2013.

[9] P. H. Nguyen, K. Xu, A. Bardill, B. Salman, K. Herd, and B. W. Wong, "Sensemap: Supporting browser-based online sensemaking through analytic provenance," in *Visual Analytics Science and Technology, 2016 IEEE Conference on*. IEEE, 2016, pp. 91–100.

[10] K. Xu, S. Attfield, T. Jankun-Kelly, A. Wheat, P. H. Nguyen, and N. Selvaraj, "Analytic provenance for sensemaking: A research agenda," *IEEE Computer Graphics and Applications*, vol. 35, no. 3, pp. 56–64, 2015.

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

[12] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman, "Lifelines: using visualization to enhance navigation and analysis of patient records." in *AMIA Symposium*. American Medical Informatics Association, 1998, pp. 76–80.

[13] P. H. Nguyen, K. Xu, R. Walker, and B. W. Wong, "Schemaline: timeline visualization for sensemaking," in *International Conference on Information Visualisation*. IEEE, 2014, pp. 225–233.

[14] ——, "Timesets: Timeline visualization with set relations," *Information Visualization*, vol. 15, no. 3, pp. 253–269, 2016.

[15] 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*, vol. 15, no. 6, 2009.

[16] 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 *SIGCHI conference on human factors in computing systems*. ACM, 2011, pp. 1747–1756.

[17] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman, "Temporal event sequence simplification," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2227–2236, 2013.

[18] W. M. Van der Aalst, *Process mining: data science in action*. Springer, 2016.

[19] W. van der Aalst and A. Weijters, "Process mining: a research agenda," *Computers in Industry*, vol. 53, no. 3, pp. 231 – 244, 2004.

[20] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.

[21] R. Agrawal and R. Srikant, "Mining sequential patterns," in *International Conference on Data Engineering*. IEEE, 1995, pp. 3–14.

[22] A. Perer and F. Wang, "Frequence: Interactive mining and visualization of temporal frequent event sequences," in *International conference on Intelligent User Interfaces*. ACM, 2014, pp. 153–162.

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

[24] P. J. Polack Jr, S.-T. Chen, M. Kahng, K. D. Barbaro, R. Basole, M. Sharmin, and D. H. Chau, "Chronodes: Interactive multifocus exploration of event sequences," *ACM Transactions on Interactive Intelligent Systems*, vol. 8, no. 1, p. 2, 2018.

[25] H. Dev and Z. Liu, "Identifying frequent user tasks from application logs," in *International Conference on Intelligent User Interfaces*. ACM, 2017, pp. 263–273.

[26] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson, "Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 321–330, 2017.

[27] 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*, vol. 36, no. 3, pp. 527–538, 2017.

[28] K. Wongsuphasawat and J. Lin, "Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at twitter," in *Visual Analytics Science and Technology, 2014 IEEE Conference on*. IEEE, 2014, pp. 113–122.

[29] Y. Chen, P. Xu, and L. Ren, "Sequence synopsis: Optimize visual summary of temporal event data," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 45–55, 2018.

[30] D. Gotz and H. Stavropoulos, "Decisionflow: Visual analytics for high-dimensional temporal event sequence data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1783–1792, 2014.

[31] E. Zgraggen, S. M. Drucker, D. Fisher, and R. DeLine, "(s|qu)eries: Visual regular expressions for querying and exploring event sequences," in *ACM Conference on Human Factors in Computing Systems*. New York, NY: ACM, 2015, pp. 2683–2692.

[32] K. Wongsuphasawat and B. Shneiderman, "Finding comparable temporal categorical records: A similarity measure with an interactive visualization," in *IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 2009, pp. 27–34.

[33] T. Munzner, *Visualization analysis and design*. CRC Press, 2014.

[34] K. Koffka, "Principles of gestalt psychology." 1935.

[35] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *International Conference on Extending Database Technology*. Springer, 1996, pp. 1–17.

[36] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 203–208, 1996.

[37] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.

[38] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko, and G. Andrienko, "Visually driven analysis of movement data by progressive clustering," *Information Visualization*, vol. 7, no. 3-4, pp. 225–239, 2008.

[39] M. Harrower and C. A. Brewer, "Colorbrewer. org: an online tool for selecting colour schemes for maps," *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.

[40] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *International Conference on Coordinated and Multiple Views in Exploratory Visualization*. IEEE, 2007, pp. 61–71.

Cagatay Turkay is a Senior Lecturer in Applied Data Science at the giCentre at the Computer Science Department of City, University of London conducting research in visual analytics. He serves as a committee member for several conferences including InfoVis and EuroVis, and part of the organising committee for IEEE VIS on 2017 and 2018. He is currently a guest editor for *IEEE Computer Graphics and Applications*, and an editorial board member for the *Machine Learning and Knowledge Extraction* journal.

Gennady Andrienko is a lead scientist responsible for visual analytics research at Fraunhofer Institute for Intelligent Analysis and Information Systems and part-time professor at City University London. Gennady Andrienko was a paper chair of *IEEE VAST* conference (2015–2016) and an associate editor of *IEEE Transactions on Visualization and Computer Graphics* (2012–2016). Currently, he is an associate editor of two journals, *Information Visualization* and *International Journal of Cartography*.

Natalia Andrienko is a lead scientist responsible for visual analytics research at Fraunhofer Institute for Intelligent Analysis and Information Systems and part-time professor at City University London. Results of her research have been published in two monographs "*Exploratory Analysis of Spatial and Temporal Data: a Systematic Approach*" (Springer 2006) and "*Visual Analytics of Movement*" (Springer 2013). Natalia Andrienko is an associate editor of *IEEE Transactions on Visualization and Computer Graphics*.

Olivier Thonnard is a Senior Security Expert for the Amadeus IT Group, the leading provider of IT solutions to the global tourism and travel industry. His R&D activities focus on machine learning and information visualization for real-world security applications, especially for fraud detection based on User Behaviour Analytics. Dr Thonnard was a Senior Principal Research Engineer at Symantec Research Labs, focusing on technology innovation and thought leadership in computer and network security.

Phong H. Nguyen is a Research Associate at the giCentre, City, University of London. His research mainly focuses on the design and application of interactive visualizations to make sense of complex datasets, with a special interest in analytic provenance, logs and general temporal categorical data. He has published papers in high-impact journals including IEEE TVCG, InfoVis, VAST, CG&A and IVS. Phong holds a PhD in Visual Analytics from Middlesex University, London, UK.

Jihane Zouaoui is a software engineer at Amadeus, Nice, France. She is leading a project on Fraud Detection built on top of big data technologies. Previously, Jihane worked as a software developer on several middleware projects mainly linked to security. Jihane holds a PhD in Computer Science from Telecom ParisTech (former ENST, Paris), and a MSc in Computer Science from CPE Lyon.