SERVIZ: AN INTERACTIVE VISUALIZATION FRAMEWORK FOR THE ANALYSIS OF SEQUENTIAL RULES AND FREQUENT ITEMSETS

by

Asal Jalilvand

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2021

Table of Contents

List of Tables

List of Figures

Abstract

Sequential Rule Mining (SRM) discovers association relationship between items in a sequence database, w.r.t. their temporal order. Often, the high number of mined rules makes their exploration challenging. Visualization of Association Rules (ARs), a closely related field in data mining, has been studied extensively to address scalability issues; however, unlike Sequential Rules (SRs), the items in ARs are not partially ordered. The small body of research investigating SR visualization enforces many constraints on the rules that make their work less generalized. We tried to address this problem by combining matrix-based visualization of ARs and the partial order between rules' items through topological sort. We developed an interactive system for mining and visualizing SRs. We experimented the effectiveness of our approach by conducting a user test and showing the reduced cognitive load for exploring SRs compared to the plain-text output of a popular off-the-shelf rule miner for a real-world dataset.

List of Abbreviations Used

AI Artificial intelligence. 21

AR Association Rule. viii

ARM Association Rule Mining. 1, 2, 7, 8

CR Causal Rules. 61

DAG Directed Acyclic Graph. vi, 28, 31, 32, 34, 59, 60

DM Data Mining. 6, 7

FIM Frequent Itemset Mining. 1, 2, 7, 8, 27, 28, 36, 38, 44, 45, 49

FPM Frequent Pattern Mining. 6, 7

GUI Graphical User Interface. 14

LHS Left-hand-side. 8, 10

NASA-TLX NASA Task Load Index. v, vii, 49–56

RHS Right-hand-side. 8, 10

RWIS Road Weather Information System. 22

SLA Service Level Agreements. 20, 23, 26, 46, 61

SPM Sequential Pattern Mining. 2, 8, 9, 16, 38

SPMF Sequential Pattern Mining Framework. vi, 14, 26, 50, 59

SR Sequential Rule. viii

SRM Sequential Rule Mining. viii, 2, 9, 10, 16, 27, 28, 41, 45, 49

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Fernando Paulovich, for his invaluable advice and continuous support throughout this research.

This research was enabled by support provided by DeepSense [1]. I would like to thank Professor Evangelos Milios and DeepSense for making my pursuit of graduate studies in Canada possible. I would like to offer my special thanks to Ms. Jennifer LaPlante, Dr. Chris Whidden and Dr. Geetika Bhatia for their insightful comments and suggestions at every stage of this project.

I want to thank Halifax Stanfield International Airport (HSIA), the partner organization responsible for this project's motivation. I wish to show my gratitude to Mr. Brian LeBlanc and all the other domain experts from Halifax International Airport Authority (HIAA), and Mr. Petr Zhigalin from Assaia for all their guidance and cooperation during this project.

I also would like to extend my gratitude to my colleagues at the lab, especially Mr. Leonardo Christino, for his support inside and outside this project.

I am deeply grateful for the support of my friends from home and the new friends I made at Dalhousie University.

I would also like to thank my mother, Mehrnaz, for her support and belief in me. I also appreciate all the support I received from my two lovely grandmothers the rest of my family.

[1] `www.deepsense.ca`

Chapter 1

Introduction

In many domains, data is stored as a sequence of discrete events, ordered by their occurrence time (e.g. stock market data, health care records, website clickstream, and customer transaction history) [29]. There are numerous ways to analyze event sequences for decision-making in real-world systems [29]. For example, the users' clickstream in websites can be analyzed for a customized product recommendation or revealing usability problems based on their navigation behaviour [29, 52]. In health care, experts can investigate disease progression from clinical records to form best practice guidelines [29, 55].

Overall, there are two kinds of analytical tasks [43]. First, predictive tasks such as prediction and recommendation. Second, descriptive tasks for analyzing past and gaining insights for decision-making [29, 43]. For predictive tasks, there are numerous state-of-the-art models, such as neural networks, that focus on accurate predictions of the upcoming events given historical data. Even though their results are promising, the downside is that these models are often designed as black-boxes, and making sense of their decisions is difficult [47, 20]. Descriptive tasks, on the other hand, discover patterns that are human-understandable [43].

Frequent pattern mining algorithms are an example of descriptive analysis tasks which are useful for finding hidden relationships in data and present them in the form of easy-to-understand patterns [51, 2, 43]. Frequent Itemset Mining (FIM) is one of the prevalent tasks in pattern mining. It was originated in the 1990s for discovering the frequently bought-together items in transaction databases for market basket analysis [20]. As an example, FIM on a set of customer transactions might show that a group of customers buy organic foods frequently. In contrast, another group is merely interested in junk food. An application of such insight could be designing group promotions to increase revenue at the given store. A closely related task that was introduced around the same time is Association Rule Mining (ARM), which discovers patterns with strong association within their items. It processes the frequently co-occurring itemsets and discards the ones with weak co-relation. These patterns can potentially find cause-effect relationships between the items and help data analysts and domain experts with decision-making [37]. To give an example, a study by Benhacine et al. [5] used association rules to investigate the relationship between mothers' socio-economic background and getting babies vaccinated on time. One of the patterns they found was that if a woman has at least four children and has a liberal profession (the *antecedent*), she will likely not respect the vaccination dates (the *consequent*). Such patterns can help the health authorities reduce the causes of low immunization.

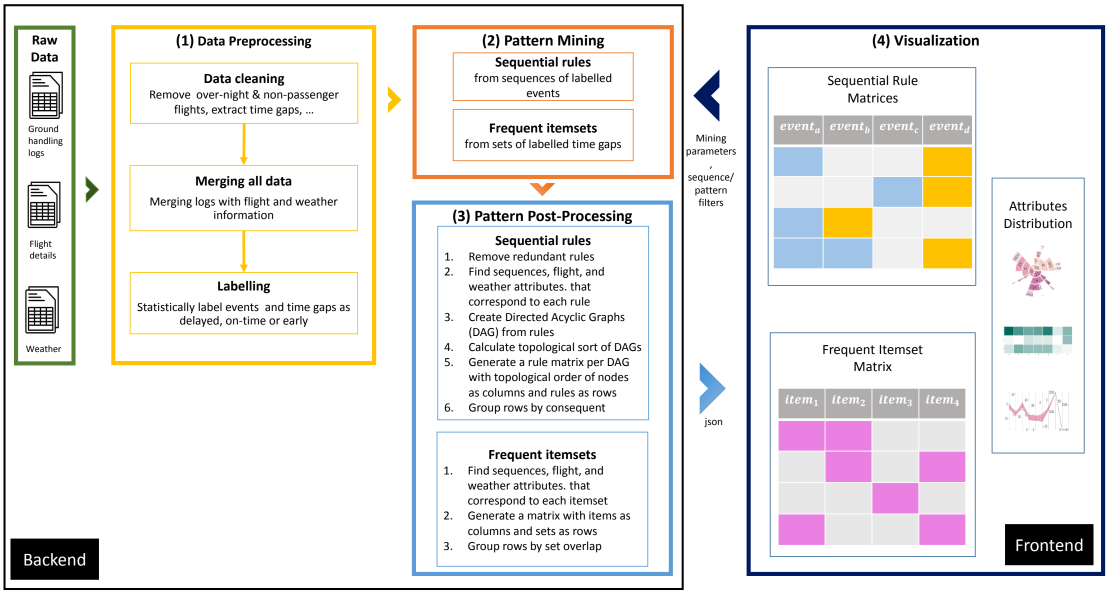
Variations of FIM and ARM apply to sequence databases, namely SRM and Sequential Pattern Mining (SPM) [20, 2, 23], respectively. An example of SPM used on a sequence database is studying user navigation behaviour on a telecommunication company's website [52]. After analyzing frequent subsequences in clickstreams, the data analysts found out customers are more likely to browse phone and data plans while visiting the website with a desktop computer. This pattern revealed a need for improving mobile site design to increase product views. Sequential rules are frequent subsequences with a high correlation between the events. A real-life example is a study by Vu et al. [73] on Australians' travel history. One of the observed travel behaviours was that people who take a trip to Laos (antecedent) are likely to visit Thailand next (consequent) with a high probability. This information can help tourism managers with travel package development.

One of the challenges of pattern mining algorithms is the extraction of a high number of patterns even from small datasets, which makes the exploration and analysis of these patterns exhaustive [41]. Algorithmic-based [20] and visualization-based [29] solutions have been proposed to deal with scalability issues to a certain extent. A visual analytics framework for exploring the patterns can reduce the high cognitive load in understanding the output of these algorithms [41]. Even though visualization of frequent itemsets, association rules and sequential patterns has been heavily researched over the years[41, 29], visualizing the sequential rules remains briefly investigated. Sequential patterns, or more precisely, frequent subsequences found in sequence databases, lack one computational layer of discarding subsequences with low correlation between the events (as in sequential rules) in their mining procedure[20]. Therefore, their visualization techniques lack the event correlation notion as well. Association rules, which are almost the counterpart of sequential rules, are not derived from sequence databases, but from unordered transactions [2, 43]. Consequently, their visualizations do not incorporate a notion of sequence and temporal order. A few studies [65, 17] attempted to bridge the gap between the visualization of association rules and sequential patterns. However, the proposed methods only work for a small set of rules in terms of readability [17], or are highly domain-specific with many enforced constraints that make the technique less usable for general scenarios [65].

This study proposes a new approach for interactive visualization of sequential rules that conveys the notion of temporal order between the items, scales for a relatively high number of rules, with few constraints regarding the rules so that the approach could be used for a variety of domains. We hypothesize that our visual analytics framework reduces the cognitive load of analysis and exploration of sequential rules compared to the plain-text output of off-the-shelf rule mining tools.

We developed our solution over a case study of finding hidden patterns within a log of ground handling operations at an airport. Ground handling operations can be view as a sequence of services an aircraft receives between its arrival and departure at an airport [67]. These services include loading and unloading of passengers, fuelling, parking or maintenance. We utilized data mining to analyze historical logs of these operations to find frequent delay-related patterns. These patterns can help the stakeholders understand what is adversely affecting ground handling operations' performance to find a solution for it.

Fig.1.1 illustrates an overview of our proposed visual analytics solution for exploring frequent patterns in ground handling operations log while addressing the challenges mentioned for sequential rule visualization. The first step is to preprocess the data (Fig.1.1 (1)). The exact steps can vary based on application. For our target domain in this study, we cleaned the logs based on guidance from domain experts and removed unimportant flights such as non-passenger ones. Then we merged all data sources (logs, flight details such as airline and aircraft information, and weather conditions during each flight) and used the information to label both the ground handling operations and the time gap between different services. Next, we used data mining



Raw Data
Ground handling logs
Flight details
Weather
(1) Data Preprocessing
Data cleaning
Remove over-night & non-passenger flights, extract time gaps, …
Merging all data
Merging logs with flight and weather information
Labelling
Statistically label events and time gaps as delayed, on-time or early
(2) Pattern Mining
Sequential rules
from sequences of labelled events
Frequent itemsets
from sets of labelled time gaps
(3) Pattern Post-Processing
Sequential rules
1. Remove redundant rules
2. Find sequences, flight, and weather attributes. that correspond to each rule
3. Create Directed Acyclic Graphs (DAG) from rules
4. Calculate topological sort of DAGs
5. Generate a rule matrix per DAG with topological order of nodes as columns and rules as rows
6. Group rows by consequent
Frequent itemsets
1. Find sequences, flight, and weather attributes. that correspond to each itemset
2. Generate a matrix with items as columns and sets as rows
3. Group rows by set overlap
Backend
Mining parameters , sequence/ pattern filters
json
(4) Visualization
Sequential Rule Matrices
$event_a$ $event_b$ $event_c$ $event_d$
Frequent Itemset Matrix
$item_1$ $item_2$ $item_3$ $item_4$
Attributes Distribution
Frontend

Figure 1.1: An overview of the SeRVis framework and data processing pipeline.

to find sequential rules in labelled sequences of ground handling operations and the frequent co-occurrence of labelled time gaps (Fig.1.1 (2)). Subsequently, the patterns were post-processed to build a matrix of sequential rules with partial order of items displayed with topologically sorted columns and a matrix of frequent itemsets (Fig.1.1 (3)). Finally, we visualize the matrices and the distribution of flight attributes per pattern (Fig.1.1 (4)). For scalability, the rows of matrices are grouped by a similarity criterion. The user interface is interactive and provides various features such as configuring pattern mining parameters, filtering sequences (or time gap itemsets) before data mining, and filtering mined patterns.

We evaluated the proposed visual analytics framework from two aspects. First, the usefulness of the tool in aviation through individual interviews with two domain experts. Second, the framework's effectiveness in terms of reduced cognitive load in pattern exploration through a study conducted with 12 users.

1.1 Contributions

Contributions of this thesis are summarized in the following:

- A novel approach for visualizing sequential rules w.r.t. to the partial order of antecedents and consequents.

- A visual analytics approach to identify sequential rules in sequences of aviation ground handling operations, as well as frequent co-occurrence of certain intervals between pairs of different operations (for example, a *long* time gap between events A and B frequently co-occurs with a *short* time gap between events A and C).

1.2 Thesis Outline

The remainder of the thesis is organized as follows. In chapter 2, we discuss the data mining concepts we utilized in this study. Next, we review the literature regarding visualization of frequent patterns in chapter 3. After discussing our target domain and data characteristics in chapter 4, we go through details of requirement analysis and system design in chapter 5. Next, we demonstrate of the system functionality with two use cases in chapter 6. We evaluate the system chapter 7,and finally conclude the thesis in chapter 8.

Chapter 2

Background and Terminology

In this chapter, we present the background and definitions of the data mining concepts used in this thesis.

2.1 Data Mining

In the information age we are living in, the governments, scientific institutions and businesses invest in collecting an enormous amount of data [43]. But in reality, they only use a small portion of it, often because there is no predefined plan on using this data [43]. In todays' competitive world, this information-rich data is considered an asset and extracting valuable information from it is an important skill [43]. Data mining is the process of using a computer-based methodology for extracting implicit knowledge hidden in data [43].

Data mining activities are categorized into the following [43]:

- *Predictive data mining* tasks that focus on making predictions based on inference on available data. Regression and classification, for example, fall in this spectrum.

- *Descriptive data mining* tasks that describe knowledge extracted from data. There is no specific target variable; instead, the focus is more on associations, correlations, and patterns in data. Cluster analysis and association rules are examples of this type of data mining.

In exploratory research, where there is no notion of an interesting outcome, one can take advantage of descriptive data mining to extract nontrivial information from data [43]. To this end, one can utilize Frequent Pattern Mining (FPM), an important subfield of Data Mining (DM), to find interesting frequent patterns from data. FPM can be used with various data types, such as transactions, sequences, strings, and graphs, and extract various types of patterns, such as associations or frequent subsequences. The interestingness of these patterns can be objective (e.g. be based upon statistical characteristics of data) or subjective (e.g. depends on application and user preference) [53].

In the rest of this chapter, we will describe four FPM tasks related to our study.

2.1.1 Frequent Itemset Mining

FIM was introduced in the 1990s to discover frequently co-occurring products in customer transactions [53, 2]. But since then, it has become a general DM task and defined as finding the frequently co-occurring groups of nominal attributes in a database. Its application in numerous domains such as product recommendation, e-learning and malware detection has made it a popular research field. The task of FIM can be formally described as follows [21, 2]:

Definition 2.1.1 (Frequent Itemset Mining)**.** Let there be a transaction database $T = \{T_1, T_2, ..., T_n\}$, where each transaction $T_i (1 \leq i \leq n)$ is a set of items
$T_i = \{x_1, x_2, x_3, ..., x_l\}$. An *itemset* P is a set of items such that $P \subseteq T_i$. The *support* (or absolute support) of itemset P, denoted as $sup(P)$, is defined as the number of transactions that contain P, that is $sup(P) = |\{T_i | P \subseteq T_i \wedge T_i \in T\}|$. Beside absolute support, it may be reported as the ratio of transactions with a certain pattern over the entire transaction database, defined as the relative support $relSup(P) = sup(P)/|T|$. The itemset P is frequent if its support is no less than a given minimum threshold *minsup*, that is $sup(P) \geq minsup$. The discovery of all frequent sets of items (or itemsets for short) in the given transaction database is the task of frequent itemset mining.

Support is commonly used as the interestingness measure in FPM [2]. An itemset with low support may have occurred in a dataset simply due to chance, thus it is traditionally considered uninteresting [68].

2.1.2 Association Rule Mining

ARM is closely related to FIM, and both were proposed around the same time in the early nineties [2] for decision-making in market basket analysis applications such as promotional pricing and product placements. But since then, ARM has been applied to other problems such as medical diagnosis and web log mining as well. ARM can be formally described as follows [32, 68, 2]:

Definition 2.1.2 (Association Rule Mining)**.** Let there be a transaction database $T = \{T_1, T_2, ..., T_n\}$, where each transaction $T_i (1 \leq i \leq n)$ is a set of items $T_i = \{x_1, x_2, x_3, ..., x_l\}$. A rule is an implication expression in form of $X \rightarrow Y$, where the itemset $X \cup Y$ is a frequent pattern and $X \cap Y = \emptyset$. The itemsets X and Y are called the antecedent (Left-hand-side (LHS)) and consequent (Right-hand-side (RHS)), respectively. Support and *confidence* of the rule $X \rightarrow Y$ are defined as $sup(X \rightarrow Y) = sup(X \cup Y)$ and $conf(X \rightarrow Y) = P(Y|X) = sup(X \cup Y)/sup(X)$. An association rule is a rule with support and confidence no less than given minimum thresholds *minsup* and *minconf*, respectively. The discovery of all association rules in the given transaction database is the task of association rule mining.

Similar to FIM, support can be used to filter out non-frequent and uninteresting rules. The confidence interest measure, which is the conditional probability of Y given X, can be viewed as a reliability measure for a rule. High confidence for the rule $X \rightarrow Y$ means a strong co-occurrence relationship between X and Y, and a higher chance of seeing Y in transactions that contain X [68]. In general, ARM is a post-processing step for FIM where the rules satisfying confidence threshold are derived from the frequent patterns [2].

2.1.3 Sequential Pattern Mining

In FIM and ARM, the order of items in transactions is not important. However, the temporal order of items is a key element in many scenarios such as analyzing the genome, web click stream, and customer buying behavior. For such scenarios, SPM is proposed for discovering frequent subsequences in a set of sequences [20, 2]. Aggarwal et. al [2] describe a sequence and a subsequence as follows:

Definition 2.1.3 (Sequence)**.** A sequence s is an ordered list of itemsets where $s = \langle I_1, I_2, ...I_n \rangle$. An itemset $I = \{i_1, i_2, ..., x_k\}$ is an unordered set of items.

Definition 2.1.4 (Subsequence)**.** A sequence $\alpha = \langle a_1, a_2, ...a_n \rangle$ is a subsequence of sequence $\beta = \langle b_1, b_2, ...b_m \rangle$, denoted by $\alpha \sqsubseteq \beta$ (α is contained in β), if there exists integers $1 \leq j_1 < j_2 < ... < j_n \leq m$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, ..., a_n \subseteq b_{j_n}$.

SPM is formally described as follows [20, 2, 19]:

Definition 2.1.5 (Sequential Pattern Mining)**.** Let there be a sequence database $S = \{s_1, s_2, ..., s_n\}$. Support of subsequence α, denoted by $sup(\alpha)$, in sequence database S is the number of sequences in S that contain α, and is defined as $sup(\alpha) = |\{s|\alpha \sqsubseteq s \wedge s \in S\}|$. The subsequence α is frequent if its support is no less than a given minimum threshold *minsup*, that is $sup(\alpha) \geq minsup$. The discovery of all frequent subsequences in the given sequence database is the task of sequential pattern mining.

2.1.4 Sequential Rule Mining

SRM is a variation of SPM where it mines $X \rightarrow Y$ rules implying if some items X happen, they will be followed by other items Y with certain confidence (conditional probability of $P(Y \mid X)$). We can view sequential rules as association rules with the constraint of X appearing before Y [21].

The confidence value helps overcoming an important limitation in SPM. That is, a pattern might have a high frequency in the data, but if the confidence is low, it might not be the most beneficial pattern to use for decision-making. For example, if the subsequence $\langle a, b \rangle$ is frequently seen in the dataset, but b only follows a in a small portion of all the sequences that a takes place in, then $a \rightarrow b$ would have a low confidence. There are two types of sequential rules, and Fournier-Viger et all. [19] describe them as follows:

Definition 2.1.6 (Standard Sequential Rule)**.** A standard sequential rule $\langle A_1, A_2, ...A_e \rangle \rightarrow \langle B_1, B_2, ...B_f \rangle$ occurs in a sequence $\langle C_1, C_2, ...C_g \rangle$ if and only if there exists integers $1 \leq x_1 < x_2 < x_e < y_1 < y_2 < y_e \leq f$ such that $A_1 \subseteq C_{x_1}, A_2 \subseteq C_{j_2}, ..., A_e \subseteq C_{x_e}$ and $B_1 \subseteq C_{y_1}, B_2 \subseteq C_{y_2}, ..., B_e \subseteq C_{y_e}$. Absolute support of a standard sequential rule $s_a \rightarrow s_b$ in a sequence database S is the number of sequences where the rule occur, and is denoted as $sup(s_a \rightarrow s_b)$. Confidence of rule $s_a \rightarrow s_b$ is defined as $conf(s_a \rightarrow s_b) = sup(s_b \cup s_a)/sup(s_a)$.

Definition 2.1.7 (Partially-Ordered Sequential Rule)**.** A partially-ordered sequential rule $I_a \rightarrow I_b$, where I_a and I_b are two unordered itemsets such that $I_a \cap I_b = \emptyset$ and $I_a, I_b \neq \emptyset$, occurs in a sequence $s = \langle I_1, I_2, ...I_n \rangle$ if and only if there exists an integer k such that $1 \leq k < n, I_a \subseteq \bigcup_{i=1}^{k} I_i$ and $I_b \subseteq \bigcup_{i=k+1}^{n} I_i$. Absolute support of a partially-ordered sequential rule $I_a \rightarrow I_b$ in a sequence database S is the number of sequences where the rule occur, and is denoted as $sup(I_a \rightarrow I_b)$. Confidence of rule $I_a \rightarrow I_b$ is defined as $conf(I_a \rightarrow I_b) = sup(I_b \cup I_a)/sup(I_a)$.

In a partially-ordered sequential rule $X \rightarrow Y$, X and Y are unordered itemsets, and the sequential ordering only exists between the antecedent and consequent of the rule and not between the items inside the RHS and LHS sets. This property makes partially-ordered sequential rules more general than the standard form so that one partially-ordered rule can represent multiple standard rules [19].

One known problem with SRM is that sometimes a number of the mined rules might be redundant[22].The literature defines redundancy as follows [22]:

Definition 2.1.8 (Sequential Rule Redundancy)**.** A sequential rule $r_a = X \rightarrow Y$ is redundant with respect to another sequential rule $r_b = X_1 \rightarrow Y_1$ if and only if $conf(r_a) = conf(r_b) \wedge sup(r_a) = sup(r_b) \wedge X_1 \subseteq X \wedge Y_1 \subseteq Y$.

2.2 Conclusion

We can see that there are many variations of patterns to mine from datasets based on the data structure. User task and domain context are the two main factors to consider when deciding on which type of pattern(s) to extract [51]. For example, if the order of items/events is important, we can look into sequential pattern variations. There are many other different types of frequent patterns [2] that we did not discuss in this chapter as their definition and applications are out of the scope of this study.

Chapter 3

Related Work

In this chapter, we review the visualization of the frequent patterns discussed in chapter 2. First, we will review the literature regarding visualization of association rules and sequential rules since these topics are most related to our study. Next, we look at sequential patterns visualization since we are interested in the same data structure (event sequences). Finally, we review the visualization of frequent itemsets.

3.1 Association Rules Visualization

Visualization of association rules is not a recent concept, and researchers have made efforts to deal with the exponential number of generated rules by visualizing them in a way that it is easier for the user to explore and interpret. In this study we aim at visualizing the final result of the mining algorithms; however, some studies aim at visualizing the mining procedure as well and allowing the user to modulate the interestingness and constraints during the iterative mining process [11]. This functionality is not included in our work. The following are the most common ways of visualizing association rules.

Tables are the most basic type of pattern visualization [54, 18, 10]. Basically, each rule/pattern is displayed in one column (or two columns could be allocated to RHS and LHS, respectively [10]), followed by columns about additional information such as confidence and support (Fig. 3.1(a)). It supports interactions such as sorting and filtering. Once the number of mined rules become exponential, such representation becomes hard to read and analyze.

Some studies suggested using scatter plots to provide an overview of interest measures (such as support and confidence) to deal with tables' scalability issues. Fujimoto et al. [26] proposed a method for filtering out interesting rules during the evaluation of generated association rules. The rules are plotted with their objective measures, such as Jaccard and Kappa. The user can select a subset with the objective measures in their desired range and further analyze the rules in that subset subjectively, using their domain knowledge. AssocExplorer [49] uses support and confidence for the scatter plot axes and colour-coded the rules based on rule length or the user-defined constraints such as availability of specific term(s) in the rules (Fig. 3.1(b)). DART [80] utilized a more advanced scatter plot, RadVis [39], to shows temporal attributes of the rules with different time granularities as anchors. These techniques could be helpful when the user is interested in exploring the patterns based on the distribution of interest measures. However, we are more concerned with focusing on the actual events in pattern and their temporal order in sequences.

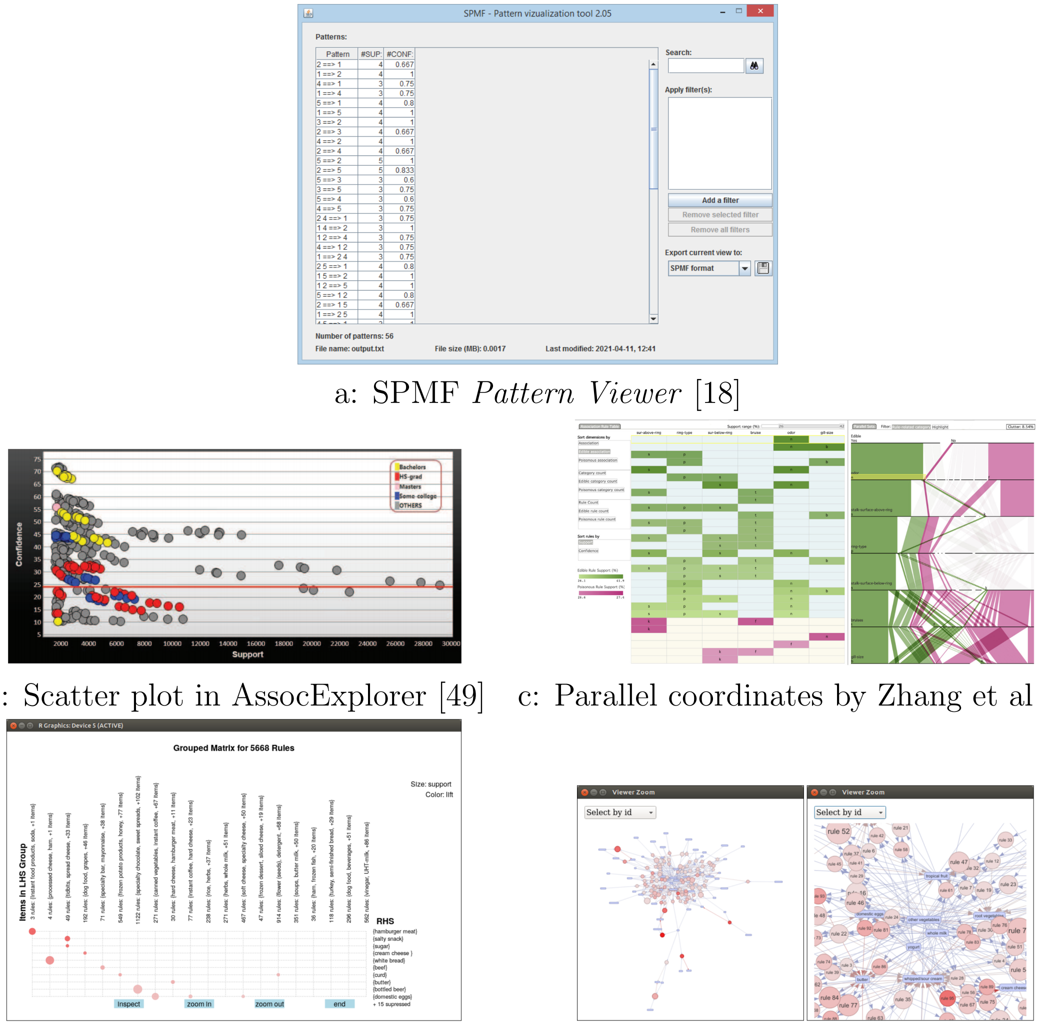
Unlike tables in which one cell displays the entire pattern as a single line, matrix-based visualization techniques organize the antecedent and consequent itemsets on the x and y-axes, respectively. Some designs such as 3DMVS [74] also used the third dimension to display support and confidence in the matrix. It is easily understood which itemsets are more frequent in the rules using matrix visualization. However, it can become challenging to read and interpret due to a high number of extracted rules [5, 32] or high-dimensional data [79]. In literature, different approaches such as defining minimum support and confidence [79], clustering [32], and extraction of the most expressive rules [5] have been explored to deal with visual clutter. Some authors proposed an extra graph-based view to keep a matrix for overview of the rules and use a graph for view details within a user-selected subset of the rules [63]. Hahsler et al. [32] proposed a clustering method for dealing with clutter and introduced grouped matrix-based visualization, which is now available in R package arulesViz (Fig. 3.1(d)). Columns (representing antecedents) of the association rule matrix grouped into hierarchical clusters based on an interest measure. Aggregated interest measures of each cluster of antecedents per consequent are visualized with balloon colour and size, at the intersection of their respective row and column. Benhacine et al. [5] proposed a solution for the clutter caused by an exponential number of rules using a colour-coded 2D matrix where the number of rules is reduced with boolean modelling. Rows and columns represent attributes and rules, receptively. The cell colour determines whether the corresponding attribute is in the LHS or RHS of the rule. These variations naturally do not incorporate the partial order attribute between RHS and LHS, as this property does not exist (or ignored) in the transaction database. However, since we are dealing with sequences, the temporal order of items is important.

Several studies investigated using parallel coordinates [40] for association rule visualization. Parallel coordinates are a method for plotting multidimensional data where each dimension is displayed with one parallel line. The association rules are visualized by connecting the RHS items within parallel coordinates [79, 75, 81]. One main problem of parallel axes is visual clutter regarding a high number of rules and high-dimensional data. Zhang et al. [79] tried to address this problem using dimensionality reduction and efficient reordering of dimensions and categories in parallel axes (Fig. 3.1(c)). However, as mentioned, the purpose of reordering is for improving the user experience and not incorporating the temporal order of the items, which is expected since temporal order is non-existent in the problem the authors addressed.

Graph-based visualizations are another popular method for displaying association rules (Fig. 3.1(e)). Common way of visualizing rules with graphs is using nodes for rule items and edges for connecting items in rules, other visual features such as the thickness of the edges or size of the nodes can be used to add support and confidence to the view [42, 31, 50, 38, 60]. Graphs are visually intuitive; Kakar el at. [42] argue that graphs are a better option compared to tables and matrices for showing the association relationships. However, they can quickly become cluttered.

There is currently a handful of commercial tools for association rule visualization such as aruleVis [31], Weka [33] and Orange [16] that provide some of the techniques discussed above. By the time of writing this thesis, these products do not support sequential rule visualization.

By definition [3], association rules are mined from transaction databases where each transaction is an unordered set of items. Therefore, we do not see a sequential order within antecedent and consequences in any of the discussed visualizations, as expected. Graph-based methods could be modified to incorporate a sequential attribute for the rules; however, they are prone to visual clutter. We proposed to use matrix-based methods with modifications to address both temporal order of items and scalability issues.



SPMF - Pattern vizualization tool 2.05

Patterns:

Pattern	#SUP:	#CONF:
2 ==> 1	4	0.667
1 ==> 2	4	1
4 ==> 1	3	0.75
1 ==> 4	3	0.75
5 ==> 1	4	0.8
1 ==> 5	4	1
3 ==> 2	4	1
2 ==> 3	4	0.667
4 ==> 2	4	1
2 ==> 4	4	0.667
5 ==> 2	5	1
2 ==> 5	5	0.833
5 ==> 3	3	0.6
3 ==> 5	3	0.75
5 ==> 4	3	0.6
4 ==> 5	3	0.75
2 4 ==> 1	3	0.75
1 4 ==> 2	3	1
1 2 ==> 4	3	0.75
4 ==> 1 2	3	0.75
1 ==> 2 4	3	0.75
2 5 ==> 1	4	0.8
1 5 ==> 2	4	1
1 2 ==> 5	4	1
5 ==> 1 2	4	0.8
2 ==> 1 5	4	0.667
1 ==> 2 5	4	1

Search:

Apply filter(s):

Export current view to: SPMF format

Number of patterns: 56

File name: output.txt File size (MB): 0.0017 Last modified: 2021-04-11, 12:41

a: SPMF *Pattern Viewer* [18]

b: Scatter plot in AssocExplorer [49] c: Parallel coordinates by Zhang et al [79]

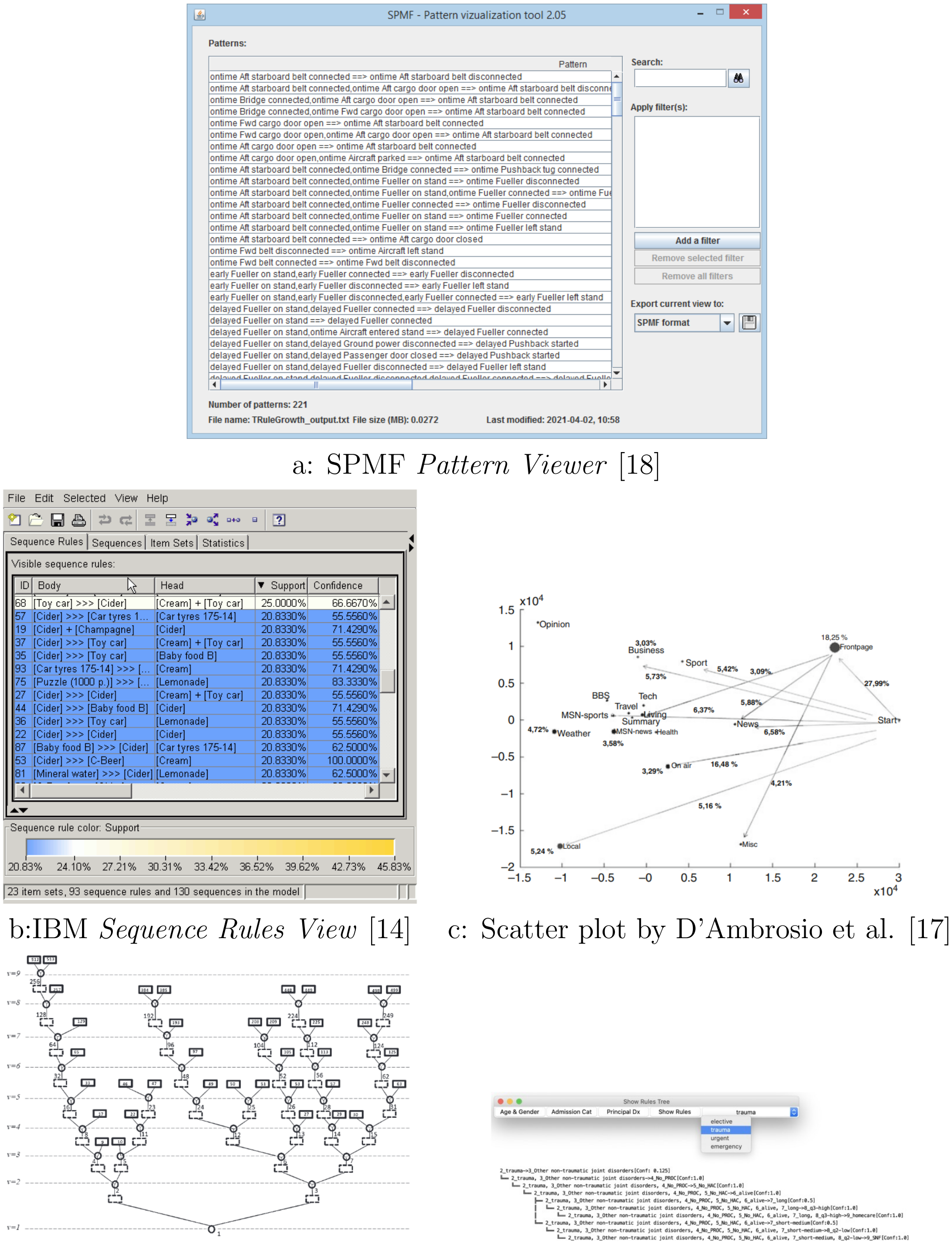
d: Matrix in aRuleVis package [32] e: Graph in aRuleVis package [31]

Figure 3.1: Association rule visualizations techniques in literature.

3.2 Sequential Rules Visualization

Visualization of sequential rules has not received much attention. IBM offers a sequence rule visualizer in the form of a table in its DB2 Intelligent Miner tool [13, 14](Fig. 3.2(b)). Another example of the use of table is the SPMF [18] that provides Graphical User Interface (GUI) for all the implemented pattern mining algorithms (Fig. 3.2(a)). This technique suffers from the same readability issues discussed for association rules.

D'Ambrosio et al. [17] investigated web usage and structure mining by measuring



SPMF - Pattern vizualization tool 2.05

Patterns:

Pattern
ontime Aft starboard belt connected ==> ontime Aft starboard belt disconnected
ontime Aft starboard belt connected,ontime Aft cargo door open ==> ontime Aft starboard belt disconne
ontime Bridge connected,ontime Aft cargo door open ==> ontime Aft starboard belt connected
ontime Bridge connected,ontime Fwd cargo door open ==> ontime Aft starboard belt connected
ontime Fwd cargo door open ==> ontime Aft starboard belt connected
ontime Fwd cargo door open,ontime Aft cargo door open ==> ontime Aft starboard belt connected
ontime Aft cargo door open ==> ontime Aft starboard belt connected
ontime Aft cargo door open,ontime Aircraft parked ==> ontime Aft starboard belt connected
ontime Aft starboard belt connected,ontime Bridge connected ==> ontime Pushback tug connected
ontime Aft starboard belt connected,ontime Fueller on stand ==> ontime Fueller disconnected
ontime Aft starboard belt connected,ontime Fueller on stand,ontime Fueller connected ==> ontime Fue
ontime Aft starboard belt connected,ontime Fueller connected ==> ontime Fueller disconnected
ontime Aft starboard belt connected,ontime Fueller on stand ==> ontime Fueller connected
ontime Aft starboard belt connected,ontime Fueller on stand ==> ontime Fueller left stand
ontime Aft starboard belt connected ==> ontime Aft cargo door closed
ontime Fwd belt disconnected ==> ontime Aircraft left stand
ontime Fwd belt connected ==> ontime Fwd belt disconnected
early Fueller on stand,early Fueller connected ==> early Fueller disconnected
early Fueller on stand,early Fueller disconnected ==> early Fueller left stand
early Fueller on stand,early Fueller disconnected,early Fueller connected ==> early Fueller left stand
delayed Fueller on stand,delayed Fueller connected ==> delayed Fueller disconnected
delayed Fueller on stand ==> delayed Fueller connected
delayed Fueller on stand,ontime Aircraft entered stand ==> delayed Fueller connected
delayed Fueller on stand,delayed Ground power disconnected ==> delayed Pushback started
delayed Fueller on stand,delayed Passenger door closed ==> delayed Pushback started
delayed Fueller on stand,delayed Fueller disconnected ==> delayed Fueller left stand

Search:

Apply filter(s):

Export current view to: SPMF format

Number of patterns: 221

File name: TRuleGrowth_output.txt File size (MB): 0.0272 Last modified: 2021-04-02, 10:58

a: SPMF *Pattern Viewer* [18]

Visible sequence rules:

ID	Body	Head	▼ Support	Confidence
68	[Toy car] >>> [Cider]	[Cream] + [Toy car]	25.0000%	66.6670%
57	[Cider] >>> [Car tyres 1...	[Car tyres 175-14]	20.8330%	55.5560%
19	[Cider] + [Champagne]	[Cider]	20.8330%	71.4290%
37	[Cider] >>> [Toy car]	[Cream] + [Toy car]	20.8330%	55.5560%
35	[Cider] >>> [Toy car]	[Baby food B]	20.8330%	55.5560%
93	[Car tyres 175-14] >>> [...	[Cream]	20.8330%	71.4290%
75	[Puzzle (1000 p.)] >>> [...	[Lemonade]	20.8330%	83.3330%
27	[Cider] >>> [Cider]	[Cream] + [Toy car]	20.8330%	55.5560%
44	[Cider] >>> [Baby food B]	[Cider]	20.8330%	71.4290%
36	[Cider] >>> [Toy car]	[Lemonade]	20.8330%	55.5560%
22	[Cider] >>> [Cider]	[Cider]	20.8330%	55.5560%
87	[Baby food B] >>> [Cider]	[Car tyres 175-14]	20.8330%	62.5000%
53	[Cider] >>> [C-Beer]	[Cream]	20.8330%	100.0000%
81	[Mineral water] >>> [Cider]	[Lemonade]	20.8330%	62.5000%

Sequence rule color: Support

20.83%	24.10%	27.21%	30.31%	33.42%	36.52%	39.62%	42.73%	45.83%

23 item sets, 93 sequence rules and 130 sequences in the model

b:IBM *Sequence Rules View* [14]

c: Scatter plot by D’Ambrosio et al. [17]

d: Tree by Siciliano et al. [66]

Show Rules Tree

```
2_trauma->3_Other non-traumatic joint disorders[Conf: 0.125]
└── 2_trauma, 3_Other non-traumatic joint disorders->4_No_PROC[Conf:1.0]
    └── 2_trauma, 3_Other non-traumatic joint disorders, 4_No_PROC->5_No_HAC[Conf:1.0]
        └── 2_trauma, 3_Other non-traumatic joint disorders, 4_No_PROC, 5_No_HAC->6_alive[Conf:1.0]
            ├── 2_trauma, 3_Other non-traumatic joint disorders, 4_No_PROC, 5_No_HAC, 6_alive->7_long[Conf:0.5]
            │   └── 2_trauma, 3_Other non-traumatic joint disorders, 4_No_PROC, 5_No_HAC, 6_alive, 7_long->8_q3-high[Conf:1.0]
            │       └── 2_trauma, 3_Other non-traumatic joint disorders, 4_No_PROC, 5_No_HAC, 6_alive, 7_long, 8_q3-high->9_homecare[Conf:1.0]
            └── 2_trauma, 3_Other non-traumatic joint disorders, 4_No_PROC, 5_No_HAC, 6_alive->7_short-medium[Conf:0.5]
                └── 2_trauma, 3_Other non-traumatic joint disorders, 4_No_PROC, 5_No_HAC, 6_alive, 7_short-medium->8_q2-low[Conf:1.0]
                    └── 2_trauma, 3_Other non-traumatic joint disorders, 4_No_PROC, 5_No_HAC, 6_alive, 7_short-medium, 8_q2-low->9_SNF[Conf:1.0]
```

e: Dendrogram by Shrestha et al. [65]

Figure 3.2: Sequential rule visualizations in literature. Table in (a) displays the 221 sequential rules mined from our dataset. The same patterns (identical pattern mining parameters) are visualized with our proposed approach in Fig. 5.1

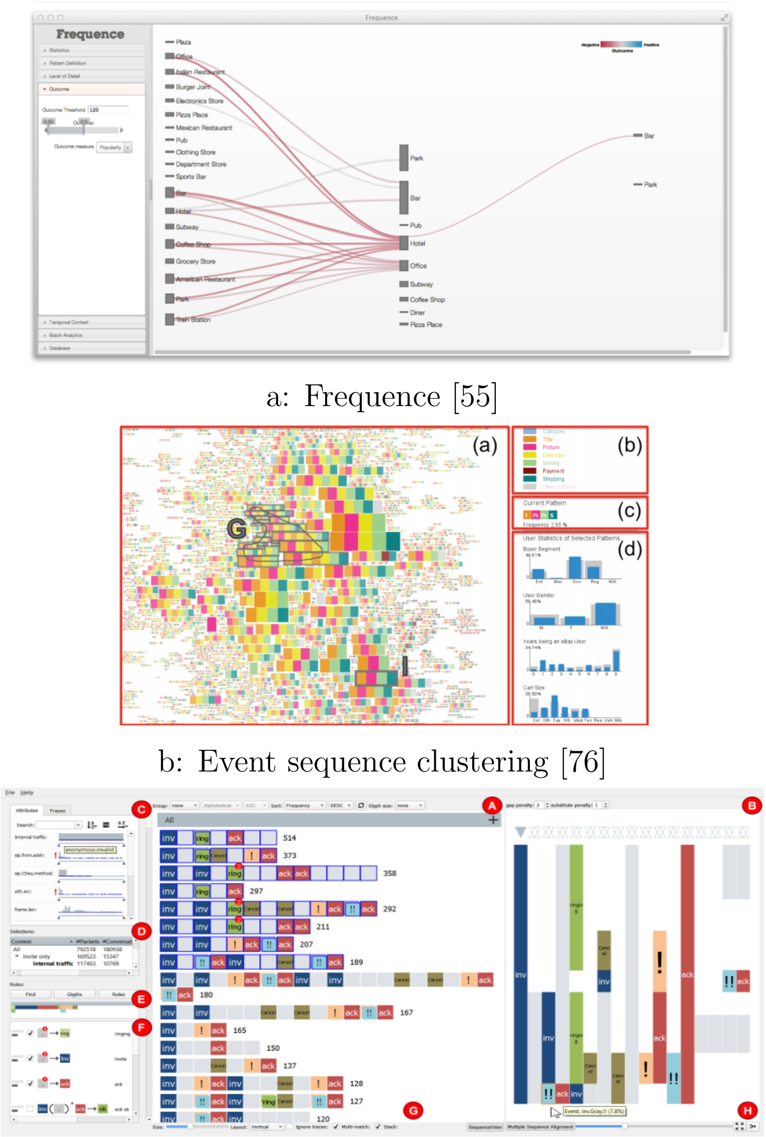
similarity between web sections with multidimensional scaling and visualizing the sequence rules of website visits by linking section nodes in the scatter plot (Fig. 3.2(e)). The rules discussed in this paper have only one antecedent and consequent; even with such simplification, the visualization is prone to clutter and readability problems. Siciliano et al. [66] used a regression tree framework to show sequence rules and predict user navigation behaviour in websites (Fig. 3.2(d)). If the user is interested in the events occurring near the end of the sequences, they have to drill down and explore all branching subpatterns, which could be time-consuming. Shrestha et al. [65] proposed a method for generating sequential rules w.r.t. a user-specified temporal order in clinical scenarios. One of the constraints imposed by the algorithm is the consecutiveness of the items in the rules. This constraint is forced to generate consequential scenarios that do not skip any diagnosis steps. The authors used dendrograms to organized the rules into hierarchies (Fig. 3.2(e)). This approach is restrictive for scenarios where we do not require all the items in the rule to be consecutive.

3.3 Event Sequence Visualization

There exists a considerable body of literature on the visualization of sequence data for different analytical tasks [29]. We studied the literature to learn about important considerations for the visualization of sequential patterns to incorporate in our visual design of sequential rules. We summarize our findings in this section.

When the number of patterns is low, Sankey-based visualization [55, 77] or simply listing each pattern in one line [45] can be used (Fig. 3.3(a)). However, when the number of patterns is high, which is often the case with SPM (and SRM), these methods can suffer from scalability issues.

Apart from the number of patterns, the length of the sequences is important to consider as well. Several studies [47, 12, 76] investigated different clustering techniques to deal with pattern visualization scalability issues and provide a summarized visualization. But they all state that their approach works best with short-lengthened sequences. The problems they faced with long sequences were either algorithmic or visualization-based. In the former, the issues included assigning the patterns to wrong clusters [47], or only one cluster when a long sequence can belong to several [12]. The



a: Frequence [55]

b: Event sequence clustering [76]

c: Multi-alignment of sequences in [9]

Figure 3.3: Event sequence visualizations in literature.

latter [76] involved cardinality issues while colour-coding the events. As the number of events grows, it can adversely affect readability (Fig. 3.3(b)).

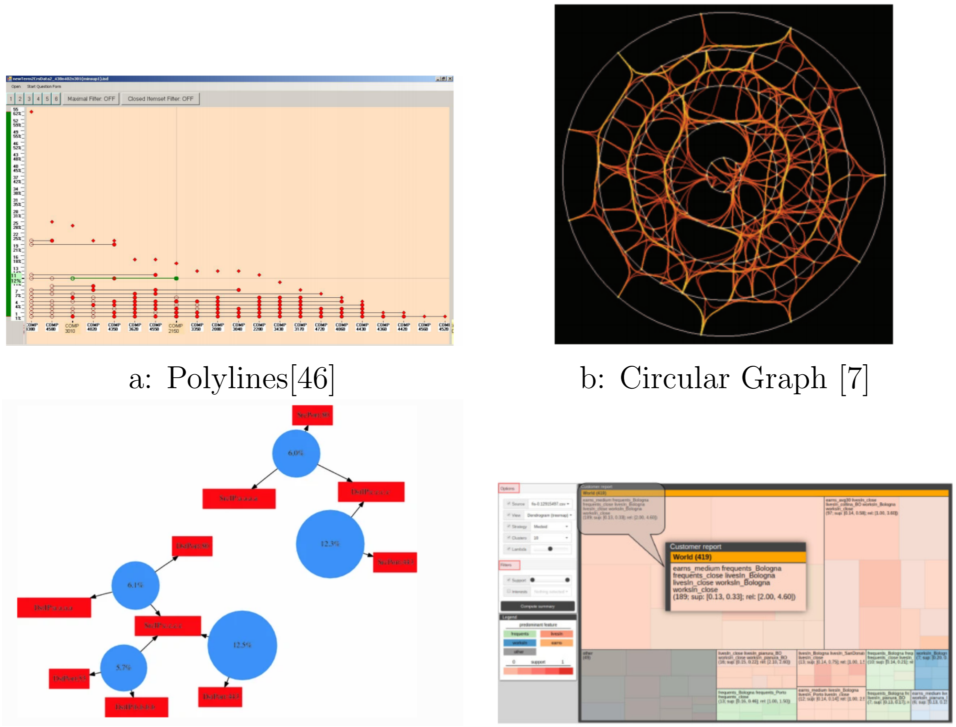
Alignment of the patterns is a useful feature utilized in literature for studying preceding and succeeding events of a selected event or point of time [58, 9, 12]. The sequential patterns may be aligned by one [12], or multiple events [58, 9], and the data analysis can compare before and after the selected event(s) across several patterns (Fig. 3.3(c)).

3.4 Frequent Itemsets Visualization

In this study, we are primarily interested in visualizing (partially-ordered) sequential rules. However, based on our analysis requirements described in section 5.1, we generated a transaction from each sequence by finding intervals between specific pairs of events and aim to visualize the patterns discovered from the generated transaction dataset. To this end, we reviewed the visualization methods for frequent itemsets (and not sets in general).

There have been numerous studies researching ways of providing a visual summary of frequent itemsets by means of polylines [46], circular graph [7], hypergraphs [27], and tree-based structures [24, 71] (Fig. 3.4).

A recent study [24] argues that previous work did not comply with Shneiderman's "overview first, zoom and filter, then details-on-demand" mantra [64] by not providing an overview of similar itemsets. The authors proposed a graph-based overview of hierarchical clusters in frequent itemsets, followed by a tree-map detail view of each cluster. Considering the need for an overview of similar patterns, we incorporated the notion of merging itemsets by a similarity measure proposed in FpVAT [46] in our matrix-based design. We utilize a similar design for both sequential rules and frequent itemsets but with minor differences. This let us design a system for visualizing both patterns without making it too complex for the end-user with different techniques.



Customer report
World (419)
earns_medium frequents_Bologna
frequents_close livesIn_Bologna
livesIn_close worksIn_Bologna
worksIn_close
(189; sup: [0.13, 0.33]; rel: [2.00, 4.60])

a: Polylines[46]

b: Circular Graph [7]

c: Hypergraphs [27]

d: Treemap [24]

Figure 3.4: Frequent itemset visualizations techniques in literature.

Chapter 4

Domain Problem Characteristics

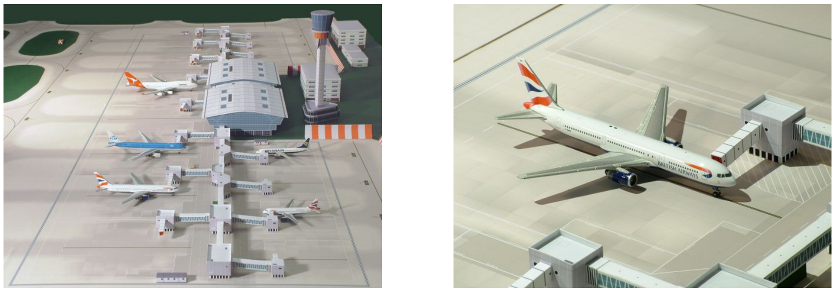
In this chapter, we first provide an explanation for a few basic terms and domain knowledge related to our target domain, followed by a description of the motivation behind this study. Then, we discuss the data characteristics since they influenced many of the design decisions of the framework.

4.1 Ground Handling Terminology

Each airport has an apron with multiple stands intended for parking of aircrafts (Fig. 4.1) [1].The time between an aircraft arrival at a stand until it is ready to depart, is called a turnaround [67]. During this time, it receives a set of services, called ground handling operations, such as fuelling, maintenance, exchange of cargo, baggage and passengers to prepare for the return journey. These services need to be completed within a scheduled time to ensure the aircraft will meet its scheduled departure time, making this stage a fundamental part of the airport operations.

4.2 Motivation

This work was motivated by exploratory research conducted on a log of ground handling operations at an airport. The stakeholders were interested in knowing if there are any patterns within the logs that would help them increase the airport's performance. Ground handling operations follow a strict chronological order and are usually expected to occur within a time slot defined in the Service Level Agreements (SLA) [61]. Inefficient turnaround operations could result in significant delays, and in turn, increase the costs for aviation stakeholders[61]. Turnaround punctuality is influenced by factors such as the airport capacity, timeliness of the ground operations, technical issues, and weather [25]. Domain experts are interested to know the patterns that adversely affect the performance to develop an appropriate solution, optimize their



a: An apron with multiple stands b: One stand

Figure 4.1: Airport apron and stand
source by:`http://airports.londoncontrol.com`

operations, and prevent revenue leakage.

We propose a visual prototype for the detection and exploration of the patterns via data mining. The advantage of such a prototype is twofold. First, the use of pattern mining algorithms decreases the time required to find such patterns manually. Second, interactive visualizations increase the efficiency of the analysis [29, 41].

4.3 Data Description

In this section, we describe the data that the airport provided for analysis. Overall, we integrated three datasets for the research described as follows:

Ground handling log: This dataset mainly includes the timestamps of start and end of the ground handling operations for each turnaround at three airport stands. The log is collected over three months (January-March 2020), using cameras installed at the designated stands. The timestamps were automatically generated by an Artificial intelligence (AI) system that processes the footage in real-time. Each turnaround has a unique identifier. Each record in the log is defined by the turnaround identifier, type of event, timestamp, and the AI system's confidence level that detected the event. The records in this dataset were anonymized and did not have flight meta-data information. Note that the AI system only logged the operations visible on the stand happening outside the aircraft. Therefore, information such as cleaning, catering, cabin inspection, and the like is not included in the logs.

Flights: This dataset includes key information about the flights during the log collection period, including flight type (arrival or departure), airline, source and destination, scheduled and the actual time of arrival/departure, and the designated stand.

Weather: Detailed weather information recorded every half an hour with a Road Weather Information System (RWIS).

4.3.1 Preprocessing

In this section, we will describe the main preprocessing steps we took (Fig.1.1 (1)). First, we derived the following from the ground handling log:

- *Turnaround duration*: Duration between first and last timestamp recorded for the turnaround.
- *Time gaps between events*: Duration between two events in a turnaround. The domain experts were interested in analyzing the duration between specific event pairs (and not all possible pairwise combinations). For example, the amount of time it takes for a fueller to start fuelling after it has entered the stand.
- *Turnaround performance*: High/low performance label for each turnaround. According to the domain experts we interviewed, a turnaround is perceived as high performance if it is serviced under an hour, with shorter intervals between its associated operations. This metric is one of the key measures used by the experts for analysis and decision-making in our target domain.
- *Event sequences*: Sequence of ground handling events in each turnaround, created from sorting the events per turnaround by their timestamp.
- *Itemsets*: An unordered set of time gaps derived for each turnaround.

We removed non-passenger and overnight flights (turnarounds with duration over 12 hours) from the data at the domain experts' instruction. We also merged the turnaround logs with flight information by matching stand and the actual time of arrival/departures. The arrival flights' actual time was compared against the *Aircraft entered stand* event timestamp. In contrast, the departures were compared with the *Aircraft left stand*.

After the merging step, we had 388 turnarounds with 52 unique events in total. The average length of the turnarounds is 19 events.

Labelling the events and time gaps

As mentioned before, we are interested in the punctuality patterns within the turnarounds. Arrival and departure events were easily labelled by comparing their actual time against the scheduled time by the airlines (table 4.1a). However, for the other events, we did not have access to the SLA contracts due to confidentiality. Therefore, we could not compare each flight's ground handling operations with the expected schedule and identify delayed events or broken protocols. Consequently, we followed a statistical approach instead. During the interviews with the domain experts, we learned that the turnarounds' performance is influenced by the airline, aircraft type, season, weekday, and day time. Therefore, we first grouped the turnaround by these factors. Then, for each group, we get the time difference between each event and an origin, then compare the interval against the 0.33 and 0.66 quantiles, and label the event as *early*, *on-time* or *delayed* (table 4.1b). Based on the recommendation of one of the domain experts that we consulted during this study, the origin is *Aircraft on stand* for high-performance flights and the start of the main operation for low-performance flights. For example, for low-performance flights, the fuelling-related operations must be timed from the entrance of the fueller on the stand. The fueller entrance itself is timed from *Aircraft on stand*. We follow a similar approach for labelling the time gaps (table 4.1c).

4.3.2 Data Characteristics

The characteristics of the data that we need to consider for the framework design are the following:

C1 **Varied turnaround length**: The turnarounds do not receive an identical set of services and depends on the flight type (for example, passenger or cargo flight) and the contract between the airline and the ground handlers, and so on.

C2 **Multivariate**: Each turnaround has a number of attributes such as the flight information, and a sequence of events (ground handling services).

C3 **High Cardinality**: The number of unique events in data is high.

Arrival and departure	**Label**
-15 min <Scheduled – actual <15 min	on-time
Scheduled – actual ≤ -15 min	early
Scheduled – actual ≥ 15 min	delayed

(a) Arrival and departure

Other events	**Label**
0.33 quantile ≤ time from origin (in seconds) ≤ 0.66 quantile	on-time
time from origin (in seconds) <0.33 quantile	early
0.66 quantile <time from origin (in seconds)	delayed

(b) Other events

Time gaps between pairs of events	**Label**
0.33 quantile ≤ interval (in seconds)≤ 0.66 quantile	on-time
interval (in seconds) <0.33 quantile	short
0.66 quantile <interval (in seconds)	long

(c) Time gaps between pairs of events

Table 4.1: Labelling the events and time gaps

4.4 Conclusion

Motivated by the research purposes described in section 4.2, we preprocessed our raw datasets to prepare them for pattern mining. We drove a sequence of ground handling operations and an unordered itemset of intervals between specific events for each turnaround. We labelled the items in sequences and itemsets based on flight information and performance for a fruitful analysis. Based on the research objectives, data characteristics of the final dataset after preprocessing, and design lessons we learned from literature, we formulated the requirements and design goals for our visual analytics framework as described in the following chapters.

Chapter 5

Methodology

In this chapter, we describe the system requirements, architecture and visual design.

5.1 Requirements

Two domain experts helped us understand the data and the turnaround performance analysis process. One of the consulted domain experts has years of experience as business aviation supervisor and as an airline duty manager. The other one is the senior technology manager at the aforementioned airport. We based the requirements on (1) the input of the domain experts over meetings and inquiries, and (2) the literature review. The derived design requirements are summarized as follows:

R1 **Provide an overview of patterns.** Even though the domain experts did not explicitly require an overview, we followed Shneiderman's visualization mantra [64] on the need for an overview that provides a high-level view of the patterns. When the number of mined patterns is high, a summary becomes helpful.

R2 **Provide exploration of sequential rules.** The domain experts stated that they are interested in the performance and punctuality analysis of the turnarounds. The sequential ordering of the operations is important, and we need a white-box analysis (as opposed to a black-box prediction model) of the relationship between the sequence events. To this end, we employ sequential rules to discover such patterns [20].

R3 **Explore relationship between intervals.** The domain experts are concerned about the time interval between different operations and often compare these gaps against the expected duration in their analysis. Our prototype should also support providing an understanding of the frequent patterns regarding intervals of interest.

R4 **Support exploration of multivariate data.** Each turnaround has several attributes, such as flight information and date. According to the domain experts, these attributes influence the turnarounds' punctuality; thus, it is essential to include these details in the analysis to derive valuable insights.

R5 **Support statistical breakdown of the patterns.** During our discussions with the domain experts, we learned that the status quo for analyzing the turnaround performance is comparing the operations against a baseline. In the absence of an accurate baseline (SLA in our case), the prototype must provide statistical details per turnaround on demand. Moreover, the experts should see the complete sequences/itemsets for each pattern to check their assumptions during analysis.

5.2 System Architecture

We developed the SeRViz, an interactive visualization framework for the analysis of sequential rules and frequent itemsets, based on the requirements in section 5.1. For the implementation, we mainly used Python[70] and Flask [28] for the backend, Vue.js [1] and D3 [6]. We used the SPMF open-source library [18] for pattern mining. The backend mines the pattern from preprocessed data and post-processes the results (Fig.1.1 (1),(2) and (3)). Users can run the frontend in modern web browsers and configure pattern mining parameters and explore the results with interactive visualizations (Fig.1.1 (4)). The backend responds the fronted requests in JSON [57] format.

For mining the sequential rules, we employed the TRuleGrowth algorithm [23]. One of the key features of this algorithm is the window size constraint. We designed the SeRViz prototype in a way that the user can control the window size. They can use their domain expertise to mine patterns within a specific time span. This choice is based on the literature on visual analytics for frequent patterns, which suggest segmenting long sequences since they can potentially include different patterns in each segment [72, 12]. Furthermore, TRuleGrowth is one of the fastest [2] and most

[1] `https://vuejs.org`

[2] depending on how the window size constraint is set

memory efficient algorithms for SRM [18, 23]. After rule mining, we post-process the rules to remove any redundancies and minimize the number of rules we need to display for the user. For FIM, we used FP-Growth [34], for the speedy execution and memory efficiency [18].

5.3 Visual Design

The design goals of the system based on the requirements are the following:

5.3.1 Design Goals

G1 Multi-level exploration of patterns. Following the visual information-seeking mantra [64], we organize the pattern analysis in the following order: the users can see an overview of the partial order of events found in rules (**R1**), explore sequential rules (**R2**) or frequent itemsets (**R3**), analyze particular pattern(s) (**R4**), and be able to view details of the associated sequences (**R5**). The users must be able to filter (1) the sequences before pattern mining, (2) the patterns after mining.

G2 Summaries of patterns based on a similarity criterion. Similar patterns should be grouped to address scalability issues. This also provides an overview (**R1**). We used identical consequent and overlapping sets as the similarity measures for sequential rules and frequent itemsets, respectively. The choice of these measures is based on our analytical approach in our use cases.

G3 Visualization of sequential rules w.r.t. to the partial order of antecedents and consequents. The patterns must be visualized in a way to convey the partial order between the items with the least cognitive load (**R2**). Graphs and Sankey diagrams are intuitive and suitable for this purpose. However, visual clutter and readability are serious issues once the cardinality (**C3**) and the number of patterns grow. Furthermore, we want the patterns to be vertically aligned based on the items. This helps the user understand; for example, an item appearing as consequent in one rule also appears in another rule as an antecedent. The alignment can help with understanding a chain-like flow of rules in the sequences. For this purpose, we will utilize a matrix-based visualization.

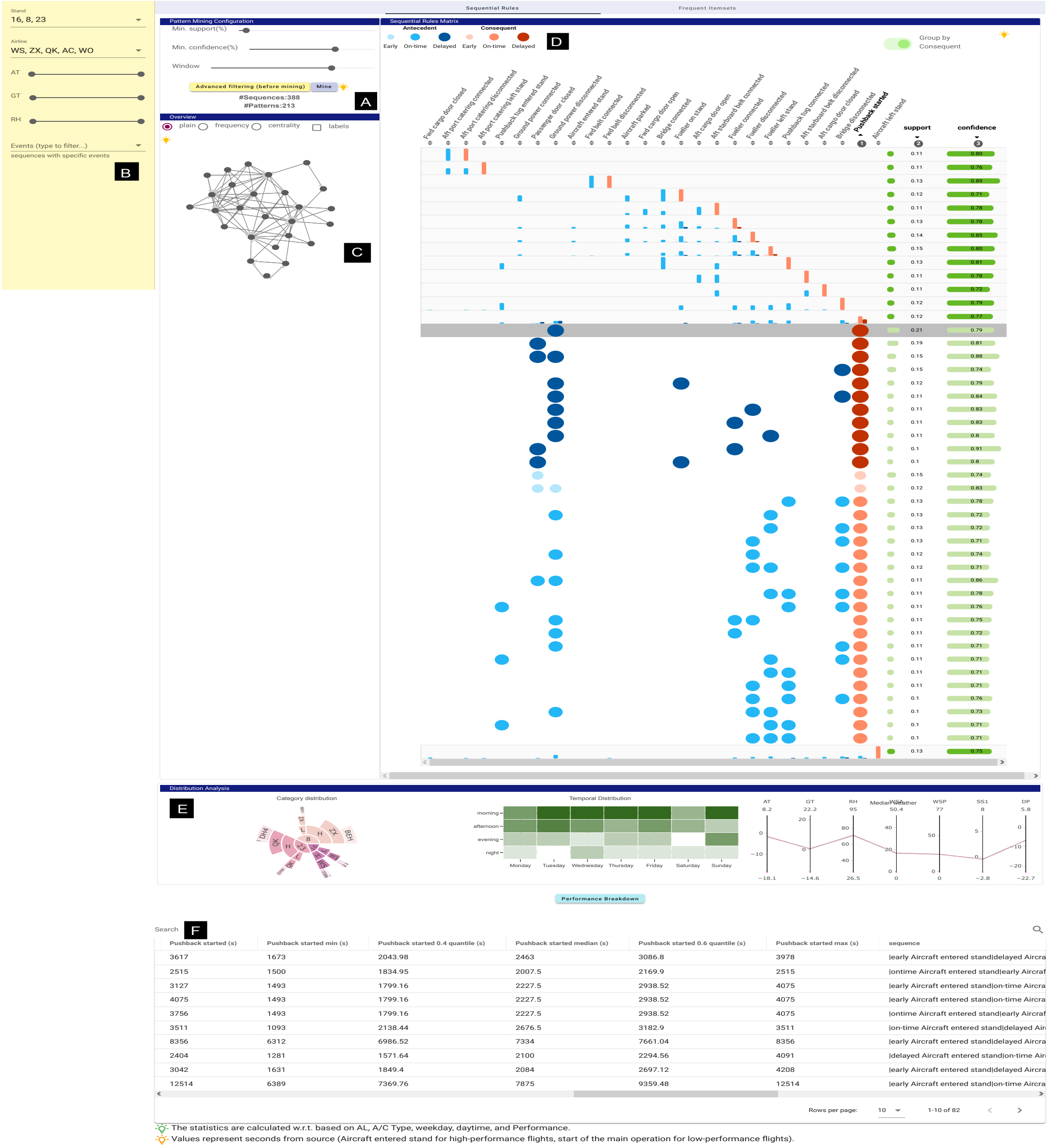
G4 **Visualization of frequently co-occurring (labelled) intervals.** The users should be able to see the FIM results (**R3**). For this purpose, we used a matrix-based visualization, similar to **G3**, so we do not overwhelm the user with too many varying views.

G5 **Support in-depth analysis of patterns.** The user should be able to explore additional data (attributes other than the events) associated with the sequences of a pattern (**R4**, **C1**, and **C2**). Furthermore, the actual sequences, along with their statistical information, must be provided on demand (**R5**).

We developed SeRViz based on the aforementioned requirements and design goals (Fig.5.1). The analytical process provided by SeRViz involves the following steps: (1) The user configures minimum support, confidence and window size for SRM (Fig.5.1 (A) (or only minimum support for FIM as in Fig. 5.2). Optionally, they can filter sequences prior to SRM (or FIM) (Fig.5.1 (B)) (2) The user will see an overview of the unique items found in the rules and their partial order (Fig.5.1 (C)) (3) The rules(or frequent itemsets) are displayed in a matrix (Fig.5.1 (D), Fig. 5.2) (4) The user can select a set of rules from the matrix and analyze them further using the distribution analysis view (Fig.5.1 (E)) (5) A more detailed information about the sequences that the selected patterns are derived from is available in the *Performance Breakdown* section (Fig.5.1 (F)).

5.3.2 Overview

Overview graph (Fig.5.1 (C)), which is a DAG, shows the partial orders found between items in the sequential rules (**R1**, **G1**). The nodes and links represent unique rule items and their partial order, respectively. A directed edge $x \rightarrow y$ means there is at least one rule $X \rightarrow Y$, where $x \in X$ and $y \in Y$ (5.3 (a)). This design is different from the common method of visualizing association rules with graphs (for example, as in the arulevis package in R [31]). There are two types of nodes in that method: one for items and one for rules. The edges demonstrate relationships in rules (5.3 (b)).



Stand
16, 8, 23

Airline
WS, ZX, QK, AC, WO

Events (type to filter...)
sequences with specific events

#Sequences:388
#Patterns:213

◉ plain ○ frequency ○ centrality □ labels

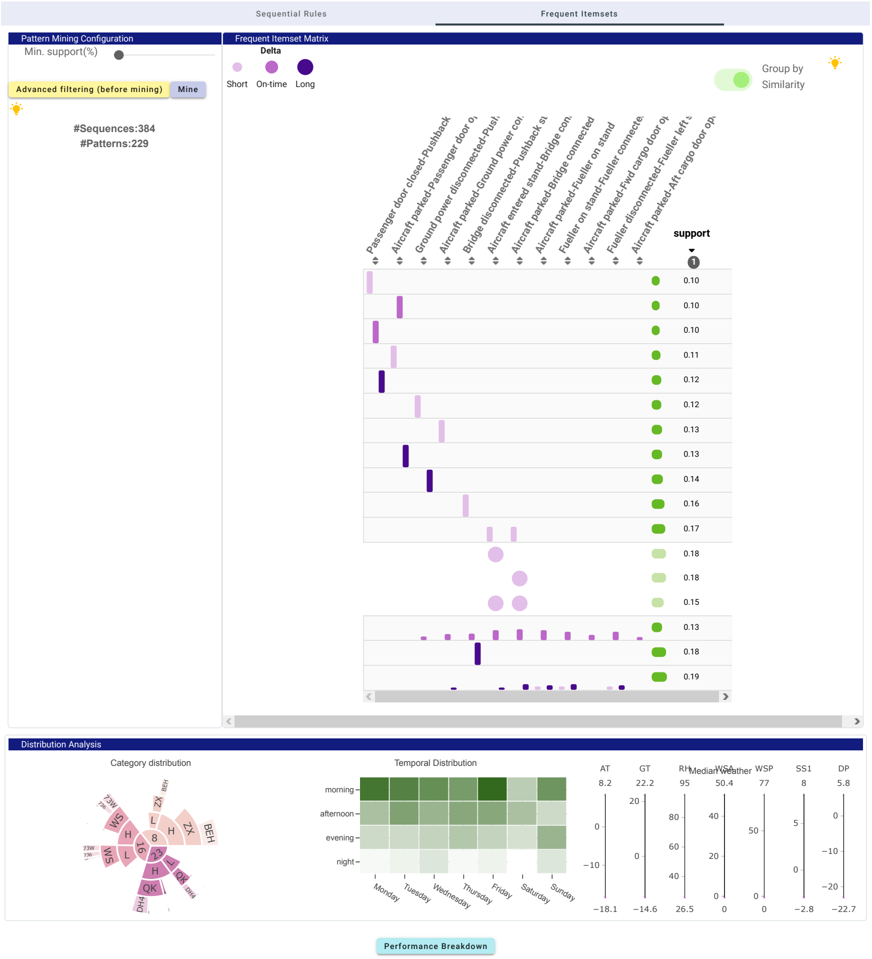
Pushback started (s)	Pushback started min (s)	Pushback started 0.4 quantile (s)	Pushback started median (s)	Pushback started 0.6 quantile (s)	Pushback started max (s)	sequence
3617	1673	2043.98	2463	3086.8	3978	|early Aircraft entered stand|delayed Aircra
2515	1500	1834.95	2007.5	2169.9	2515	|ontime Aircraft entered stand|early Aircraf
3127	1493	1799.16	2227.5	2938.52	4075	|early Aircraft entered stand|on-time Aircra
4075	1493	1799.16	2227.5	2938.52	4075	|early Aircraft entered stand|on-time Aircra
3756	1493	1799.16	2227.5	2938.52	4075	|ontime Aircraft entered stand|early Aircraf
3511	1093	2138.44	2676.5	3182.9	3511	|on-time Aircraft entered stand|delayed Air
8356	6312	6986.52	7334	7661.04	8356	|early Aircraft entered stand|delayed Aircra
2404	1281	1571.64	2100	2294.56	4091	|delayed Aircraft entered stand|on-time Air
3042	1631	1849.4	2084	2697.12	4208	|early Aircraft entered stand|delayed Aircra
12514	6389	7369.76	7875	9359.48	12514	|early Aircraft entered stand|on-time Aircra

Rows per page: 10 1-10 of 82

The statistics are calculated w.r.t. based on AL, A/C Type, weekday, daytime, and Performance.

Values represent seconds from source (Aircraft entered stand for high-performance flights, start of the main operation for low-performance flights).

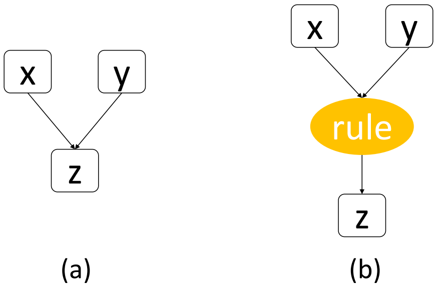
Figure 5.1: SeRViz:(A) Sequential rule mining configuration for support, confidence and window size. (B) Sequence filtering based on attributes and events. User will adjust the options prior to pattern mining. (C) Overview graph. The nodes are unique events found in rules, and the edges represent partial order between the events. (D) Pattern matrix. The columns are events and rows are rule. The rows are grouped by consequent. Table legend displays event label guide. One of the groups is collapsed and rows are sorted by consequent label, support and confidence. (E) Distribution analysis views showing distribution of categorical and numerical attributes of the sequences associated with (selected) patterns. (F) Performance breakdown table showing sequence details.



#Sequences:384
#Patterns:229

Figure 5.2: SeRViz: Frequent itemsets view

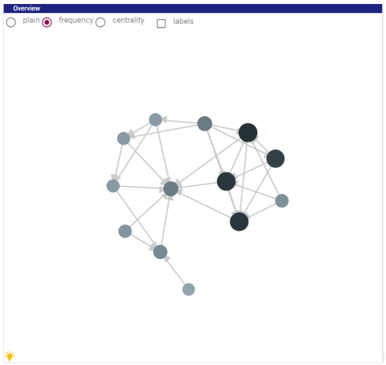
The main idea behind this graph is finding an order for the columns of the rule matrix using topological search. This graph's rationale and generation are discussed in more detail in section 5.3.3. The colour and size of nodes can optionally show either frequency of the item in the mined patterns (Fig.5.4), or the betweenness centrality of the node. The betweenness centrality measures how much a node falls on the shortest path between other pairs of nodes in a graph [35]. A node with high centrality has a strong influence over the information passing in the graph. An example of analyzing betweenness centrality in the context of pattern mining is finding nodes with high centrality degrees in a network of association rules from prescription records to find a symptom that can be cured with many types of herbal teas [78]. Optionally, the user can select a node from the graph to explore the patterns with that particular item in



(a)

(b)

Figure 5.3: Two graphs for sequential rule $\{x, y\} \to z$: (a) DAG of the partial order (b) rule graph.


Overview

○ plain ● frequency ○ centrality □ labels

Figure 5.4: Overview graph with node size and colour adjusted by item frequency in the sequential rules.

the antecedent or consequent.

5.3.3 Sequential Rule View

For visualizing the patterns, we utilized a matrix in which the rows and columns represent patterns and unique events, respectively (Fig.5.1 (D)) (**R2,G1-G3**). A matrix is a common visualization choice for association rules, as mentioned in chapter 3. Considering the characteristics of our data (**C1,C3**), we chose matrix since it is more readable and less prone to clutter and overlapping links than graphs or Sankey diagrams.

Unlike association rules, the antecedent and consequent itemsets in sequential rules are partially ordered. Using a random order for the matrix columns can be confusing in terms of conveying the partial order. For example, in Fig. 5.5 (b), for patterns in rows r_2 and r_3, the user can identify the antecedents and consequents easily with colors, but it may not be evident that the blue events take place *before*

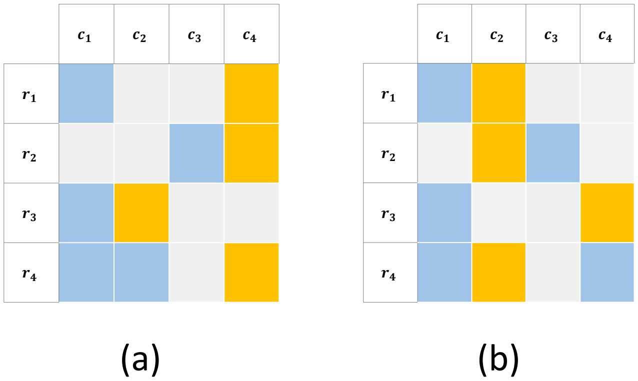


Figure 5.5: Matrix with a. partially b. randomly ordered columns. Blue and orange cells represent antecedent and consequent, respectively.

the orange ones. However, the order of columns in Fig. 5.5(a) is more aligned with the common left to right flow of sequences. Another benefit of the latter design is that the patterns are inherently aligned by events. Sequence (or pattern) alignment is one the most common techniques used in event sequence analysis [29]. This is useful for comparing the analyzing of events before and after an alignment point over multiple patterns.

We propose algorithm 1 for finding a column order for the pattern matrix that incorporates the partial order of events. For the implementation, we used the Networkx package in Python [30].

We create a DAG from sequential rules where a directed edge $x \rightarrow y$ means there is at least one rule $X \rightarrow Y$, where $x \in X$ and $y \in Y$ (algorithm 2). Then, by sorting the vertices topologically, we derive an order for the matrix columns. This way, we make sure that the antecedents do not appear after consequents in the matrix rows. For avoiding cycles in the graph, we might need to create multiple DAGs, and in turn, multiple matrices (algorithm 1). Throughout our explorations, we rarely encountered such situations (for example, in Fig. 5.6, which is for patterns mined specifically from sequences with *delayed Aft startboard belt connected*). In such situations, the user can view the matrices by selecting nodes from each DAG in the overview.

Algorithm 1: Generating sequential rule matrices w.r.t. to the partial order of antecedents and consequents.

```
Input: Set of sequential rules $R = \{r_1, ..., r_i\}$
Output: Rule Matrices with partially ordered columns $M = \{m_1, ..., m_k\}$
$D \leftarrow \emptyset$ $\triangleright$ where $d_i$ is a DAG generated from items in rules

$M \leftarrow \emptyset$ $\triangleright$ where $m_i$ is a matrix with columns generated from topological sort
 of $d_i$ vertices

foreach $r_i \in R$ do
    $inserted \leftarrow false$
    foreach $d_j \in D$ do
        $\acute{d_j} \leftarrow$ AddRuleToDAG($r_i$,$d_j$)
        if $\acute{d_j} \neq \emptyset$ then
            $inserted \leftarrow true$
            break
        end
    end
    if $inserted \neq true$ then
        $D \leftarrow D \cup$ AddRuleToDAG($r_i$,$K_0$)
    end
end
foreach $d_i \in D$ do
    $m_i \leftarrow d_i$ rules $\times$ topological sort of $d_i$ vertices
    $M \leftarrow M \cup m_i$
end
```

Item Glyphs

The rule items are visualized with circles in each row. We appended the relative support and confidence to the matrix as columns (Fig.5.7). To improve the readability, we used bars with width representing support and confidence values. Inspired by [5], we encoded antecedent and consequents with colours. The sequence events are

Algorithm 2: Adding a sequential rule to a DAG

```
Function AddRuleToDAG($r$, $d$):
    $g \leftarrow d$
    foreach $a \in r.LHS$ do
        $g \leftarrow g \cup (a, r.RHS)$
    end
    if $g$ is acyclic then
        return $g$
    else
        return $\emptyset$
    end
```

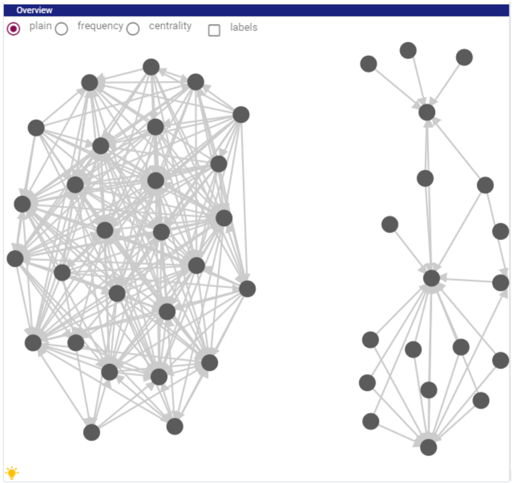
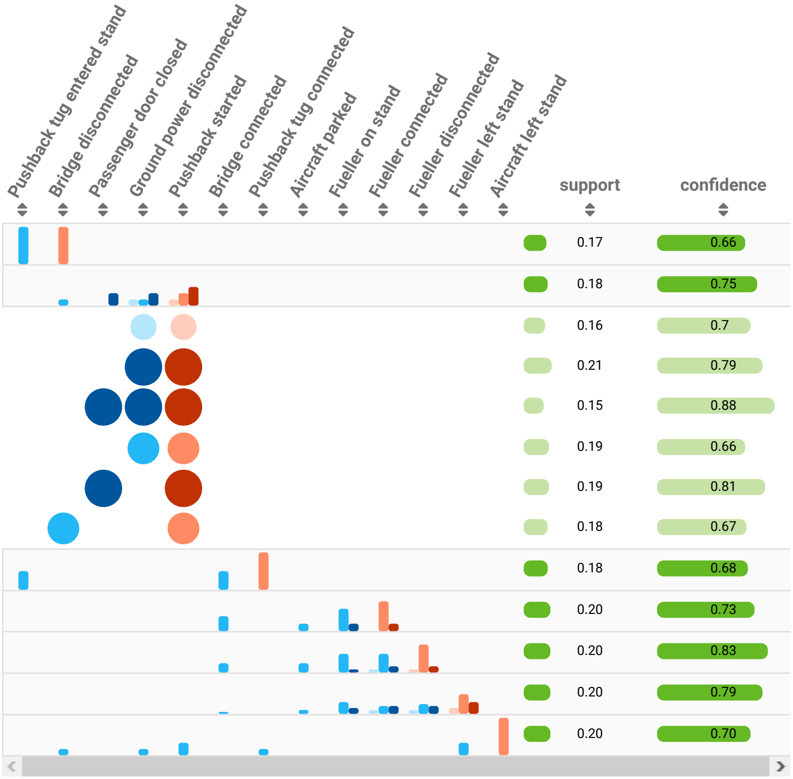


Figure 5.6: Two DAGs found for a set of sequential rules.

labelled as *delayed*, *on-time* or *early* (which are naturally ordinal), and we encoded these labels with color intensity and glyph size (Fig.5.8). If we had more classes, we would avoid this design as it can become difficult to differentiate between colour intensities.

Design alternatives: Instead of glyph colour and size, we could use one column per label. This design would lead to $\#labels \times \#events$ number of columns, resulting in a sparse matrix. Another option was not to use a matrix-based visualization and use a combination of colour-coding the events and (multi)alignment, such as in [12, 9]. However, since we have many different event types (**C3**), it would be visually difficult



support	confidence
0.17	0.66
0.18	0.75
0.16	0.7
0.21	0.79
0.15	0.88
0.19	0.66
0.19	0.81
0.18	0.67
0.18	0.68
0.20	0.73
0.20	0.83
0.20	0.79
0.20	0.70

Figure 5.7: Sequential rule matrix. Columns and rows represent items and rules, respectively. The last two columns display relative support and confidence. The rows are grouped by consequent. The second row is expanded to view all the patterns in one group.

to distinguish between the events [69].

Consequent Groups

One problem we faced using a matrix is that it does not scale well for a high number of patterns. Furthermore, the patterns do not always include all the unique events in the dataset, especially if we use a small window size in the TRuleGrowth algorithm. This leads to a sparse matrix with many rows. We grouped the rows by consequent (**G2**). The choice of this criterion is based on our analysis approach, which is mostly target-based. That is, we ask *"what happens before x?"* or *"what if x happens?"*. We relaxed the problem by minimizing the scope to rules with only one consequence. To show the distribution of each item in the group, we used a histogram per item. For support and confidence, we report the average of the group.

When the matrix is in grouped mode, the group rows are sorted by the same

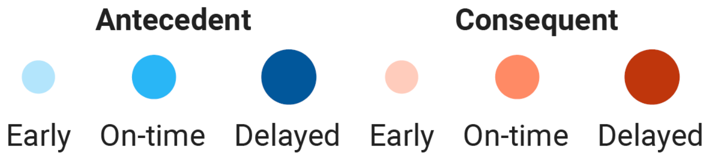


Figure 5.8: Encoding antecedent, consequent and labels with glyphs.

topological order used for ordering the columns (Fig.5.7). The purpose is to embed a chain-like flow of patterns in the matrix instead of an arbitrary order of rows.

5.3.4 Frequent Itemset View

We utilize FIM for finding frequent co-occurring labels within specific intervals between events (Fig.5.2) (**R3**, **G1**, **G2** and **G4**). For the domain experts, these intervals were the most important ones to analyze among all the pairwise combinations of events. We used a matrix with design choices similar to the sequential rules matrix, including choice of using colour and shape size for representing ordinal labels (Fig. 5.9). The most notable differences are the following:

- The order of columns are arbitrary. For each pattern we only have one set of items, unlike sequential rules were we had two partially ordered itemsets.

- Row grouping criterion is itemset overlap instead of similar consequents.

Similarity Groups

We grouped similar rows in the matrix for dealing with scalability issues caused by a high number of patterns (**G2**). We used set overlap as similarity metric, and define two frequent itemsets F_i and F_j similar if $F_i \cap F_j \neq \emptyset$.

Grouping the rows was inspired by FpVAT [46], which compressed the frequent patterns with the same frequency range in one line. As Alsallakh et al. discuss in their set visualization survey [4], the choice of similarity measure for sets depends on the data and end goals. Thus, the grouping can be altered to use other metrics such as Jaccard distance based on requirements.

Alternative: We initially built a graph based on itemsets similarity, with frequent itemsets and non-empty overlaps as nodes and edges, respectively. The graph edges

Delta
Short
On-time
Long

Figure 5.9: Encoding item labels with color and size for frequent itemsets matrix. The term *Delta* refers to time difference between two events $\delta(e_a, e_b)$.

quickly become cluttered and difficult to analyze. We decided to discard this graph both as the main visualization and the overview.

5.3.5 Distribution Views

Considering the multivariate nature of our data (**C2**), we utilized three charts to show the distribution of sequence attributes (Fig. 5.1 (E)) (**R4**, **G1** and **G5**). According to domain experts, the categorical and numeric attributes associated with the ground handling event sequences impact the operations' performance. Thus, they are not only interested in the frequent patterns, but also the sequences' associated information. The sunburst and calendar heatmap shows the distribution of categorical attributes of the sequences used for pattern mining. The median of weather attributes per pattern is visualized with parallel coordinates. These charts answer questions like "*this (these) pattern(s) happened at which stands? for which airlines? what time of day? what was the weather like when this happened?*". If the user is interested in a pattern, they can select it from the matrix by clicking on its row and see its distribution charts. For example, in Fig. 5.1, a row is selected from the matrix (D), and the charts show the distribution of attributes of the sequences with this pattern (E). If the user does not select any particular patterns, the charts display distribution over the entire (filtered) sequences.

5.3.6 Statistical Breakdown View

The statistical breakdown table (Fig. 5.1.F) provides details about the pattern sequences for the domain expert to validate their hypothesis (**R5**, **G1** and **G5**). The table is hidden to save space and becomes visible by selecting patterns and clicking on the performance breakdown button. The domain expert can inspect the raw event sequences and their categorical attributes (**C2**). Since we have labelled the events based on certain attributes and statistics, we provided this information along with the actual sequences. For example, if an event is labelled as *delayed* w.r.t. its occurrence time since the beginning of turnaround ($\delta(start, event)$), the domain expert can see the actual δ along with *minimum, 0.33 quantile, median, 0.66 quantile* and *maximum* of that δ in sequences(4.3.1).

5.3.7 Interactions

SeRViz supports the following interactions to enhance the pattern exploration:

Interactive mining: Users can tune the SPM and FIM parameters (Fig. 5.1 (A), 5.2). Initially, the user can start working with patterns mined with preset parameters and then tune them if required.

Sequence Filtering: Sequences can be filtered by categorical values and labels (for example, mine patterns from sequences having *delayed fueller connected*) before pattern mining. Furthermore, the performance breakdown table support text and number search.

Pattern Filtering: If the user is only interested in patterns with a particular event in the antecedent/consequent of the sequential rules, they can select the event node from the overview graph. The rest of the views will be updated to show related patterns/details for the selected node.

Pattern Selection: The rows in the pattern matrix are selectable, and once clicked, the distribution charts and performance breakdown table are updated accordingly.

Sorting: The pattern matrix support soring rows by multiple columns. The support and confidence will be sorted numerically, and event columns by the label.

5.4 Conclusion

In this chapter, we described our design requirements and how we tried to address each of them with the design goal. For each visual component's design, we also tried to consider our data characteristics and the questions the component will answer. We demonstrate the usability of SeRVis with a few use cases in the next chapter.

Chapter 6

Use Case

In this chapter, we use two use cases to demonstrate how the SeRVis system can be used to find patterns in a log of ground handling operations. In the first use case, we look for interesting patterns in a set of turnarounds where the fuelling was delayed. The second use case is about finding undesirable patterns (for example, any longer than usual intervals between operations) in freezing weather.

6.1 Sequential Rules from Sequences with Delayed Fuelling

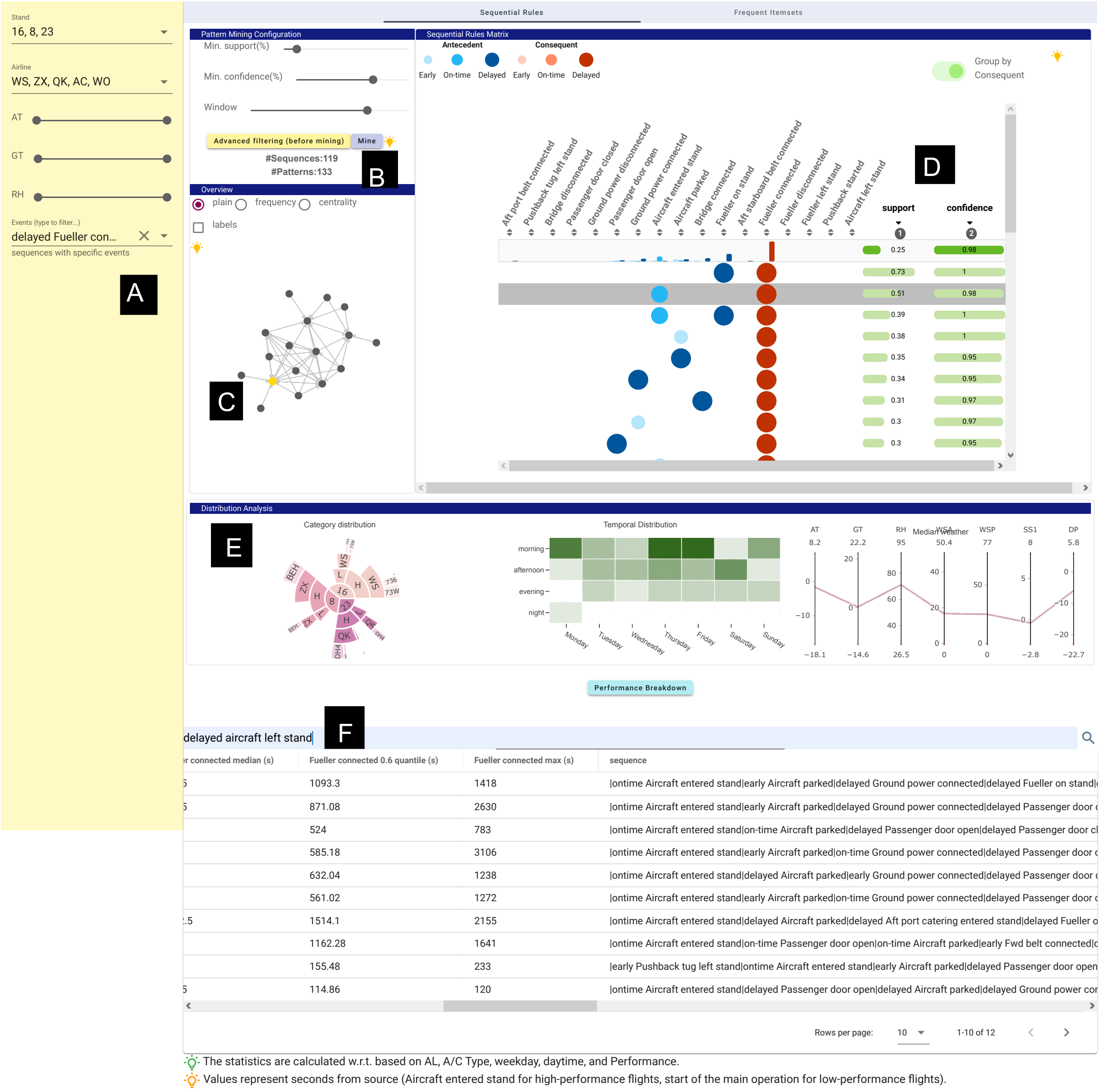
This scenario presents Alice, a manager at an airport who wants to analyze the turnarounds with delayed fuelling, or more precisely, turnarounds with labelled event *delayed fueller connected*. Fuelling is one of the ground handling operations the domain experts, such as Alice, are most interested in. Inefficient fuelling is a waste of energy and a loss of money for all stakeholders. Alice does not have the time to go through every single turnaround individually, so she decides to use SeRVis to find and analyze the most frequent patterns. She is interested in knowing what events, including *Fueller connected*, are delayed and what were the steps that led to them and in what conditions.

She first filters the sequences with *delayed fueller connected* (Fig.6.1 (A)). With only 119 flights with delayed fuelling, she defines a pattern interesting if it has appeared in at least near 20 flights (support = 15%), and the probability of seeing an event given a certain subsequence before it must be high (confidence = 70%) (Fig.6.1 (B)). She sets the window size to 15, half of the maximum sequence length in this dataset. She is interested in the patterns from ground handling operation events within close proximity. After setting the mining configuration, she mines the patterns.

Initially, the matrix is in grouped mode (Fig. 6.2). Alice notices aggregated antecedents found for *Fueller connected* as a consequent (red rectangle in Fig. 6.2). She learns that *Aircraft entered stand* (mostly laballed as *on-time*) and *Fueller on stand* (mostly lablled as *delayed*) were top most frequent antecedents of her target event (red rectangle in Fig. 6.2). *delayed Fueller connected* affected *Fueller disconnected*, *Fueller left stand* and *Pushback started* and partially contributed to their delay (blue rectangle in Fig. 6.2).

Alice selects her target event, *Fueller connected*, from the overview graph to filter the related patterns. Then, she looks into the group with her target event as the consequent and sorts the rows by support. One unexpected rule she finds is {*Ontime aircraft on stand*} → {*delayed fueller connected*}, which has occurred in 51% of the filtered sequences, with almost perfect confidence. She was not expecting to see a delay in fuelling for turnarounds that were arrived on time and ideally would get serviced as scheduled. She selects the row from the table and analyzes the associated attributes. The pattern existed in all three airport stands, mostly in mornings (between 4:00 A.M. to 11:59 A.M.), and in cold (air and ground temperature around 0 $^{\circ}C$) and humid (relative humidity around 70%). At this point, Alice could look into these flights' turnaround history to investigate if something is blocking the fueller from getting connected to the aircraft in the situations mentioned above. She gets curious to know how many of the flights with this pattern also had a delayed departure. Thus, she queries the performance breakdown table with *delayed Aircraft left stand*, which giver her 12 (out of 60) flights. This could mean the ground handling operations were often swift enough to prepare the aircraft to depart on time despite the disruptions.

Next, Alice explores the patterns related to *Aircraft left stand* to see in she finds any patterns regarding delayed departures, which have a lot of costs associated with them [56]. These costs include negative passenger experience, compensation of delayed flights for airlines, and wasted labour productivity [56]. Alice selects her new target from the overview (Fig.6.3 (A)). In the group of *Aircraft left stand* as the consequent, she notices a dark orange bar in label distribution, which represents a delay (Fig.6.3(B)). We collapse the group and find the frequent pattern {*delayed passenger door open, delayed pushback started*} → {*delayed aircraft left stand*} with the support of 17% and confidence of 87%. From the distribution analysis views, Alice learns that this trend has occurred mostly in stands 8 and 23 (Fig.6.3(C)). It seems like this



Stand: 16, 8, 23

Airline: WS, ZX, QK, AC, WO

AT

GT

RH

Events (type to filter...): delayed Fueller con... ✕

sequences with specific events

Min. support(%)

Min. confidence(%)

Window

Advanced filtering (before mining) | Mine

#Sequences:119
#Patterns:133

● plain ○ frequency ○ centrality

□ labels

support	confidence
0.25	0.98
0.73	1
0.51	0.98
0.39	1
0.38	1
0.35	0.95
0.34	0.95
0.31	0.97
0.3	0.97
0.3	0.95

delayed aircraft left stand

r connected median (s)	Fueller connected 0.6 quantile (s)	Fueller connected max (s)	sequence
5	1093.3	1418	|ontime Aircraft entered stand|early Aircraft parked|delayed Ground power connected|delayed Fueller on stand|
5	871.08	2630	|ontime Aircraft entered stand|early Aircraft parked|delayed Ground power connected|delayed Passenger door
	524	783	|ontime Aircraft entered stand|on-time Aircraft parked|delayed Passenger door open|delayed Passenger door cl
	585.18	3106	|ontime Aircraft entered stand|early Aircraft parked|on-time Ground power connected|delayed Passenger door
	632.04	1238	|ontime Aircraft entered stand|delayed Aircraft parked|early Ground power connected|delayed Passenger door
	561.02	1272	|ontime Aircraft entered stand|early Aircraft parked|on-time Ground power connected|delayed Passenger door
.5	1514.1	2155	|ontime Aircraft entered stand|delayed Aircraft parked|delayed Aft port catering entered stand|delayed Fueller o
	1162.28	1641	|ontime Aircraft entered stand|on-time Passenger door open|on-time Aircraft parked|early Fwd belt connected|
	155.48	233	|early Pushback tug left stand|ontime Aircraft entered stand|early Aircraft parked|delayed Passenger door open
5	114.86	120	|ontime Aircraft entered stand|delayed Passenger door open|delayed Aircraft parked|delayed Ground power cor

Rows per page: 10 1-10 of 12

The statistics are calculated w.r.t. based on AL, A/C Type, weekday, daytime, and Performance.

Values represent seconds from source (Aircraft entered stand for high-performance flights, start of the main operation for low-performance flights).

Figure 6.1: Target based analysis for delayed aircraft fuelling. (A) The sequences are filtered by target event. (B) SRM parameters are set. (C) Target event is selected from the overview graph (yellow node). Tooltips or label option help the user find the event. (D) Group with target event as consequent is collapsed and sorted by support column. A pattern is selected from the matrix. (E) Attribute distribution analysis for the selected pattern. (F) Sequence details for the selected pattern. User can search for a specific event in the sequences.

pattern frequently happens throughout the week. At this point, Alice could form a hypothesis about a passenger boarding issue at these two stands and investigate if it is true using full turnaround history (if available) of the corresponding flights.

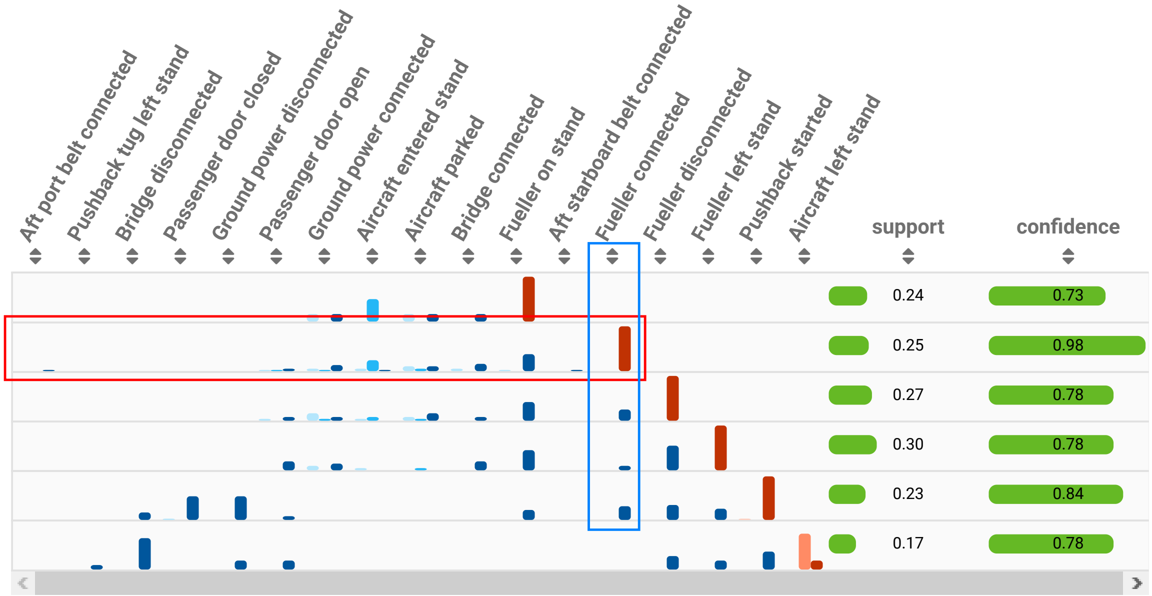
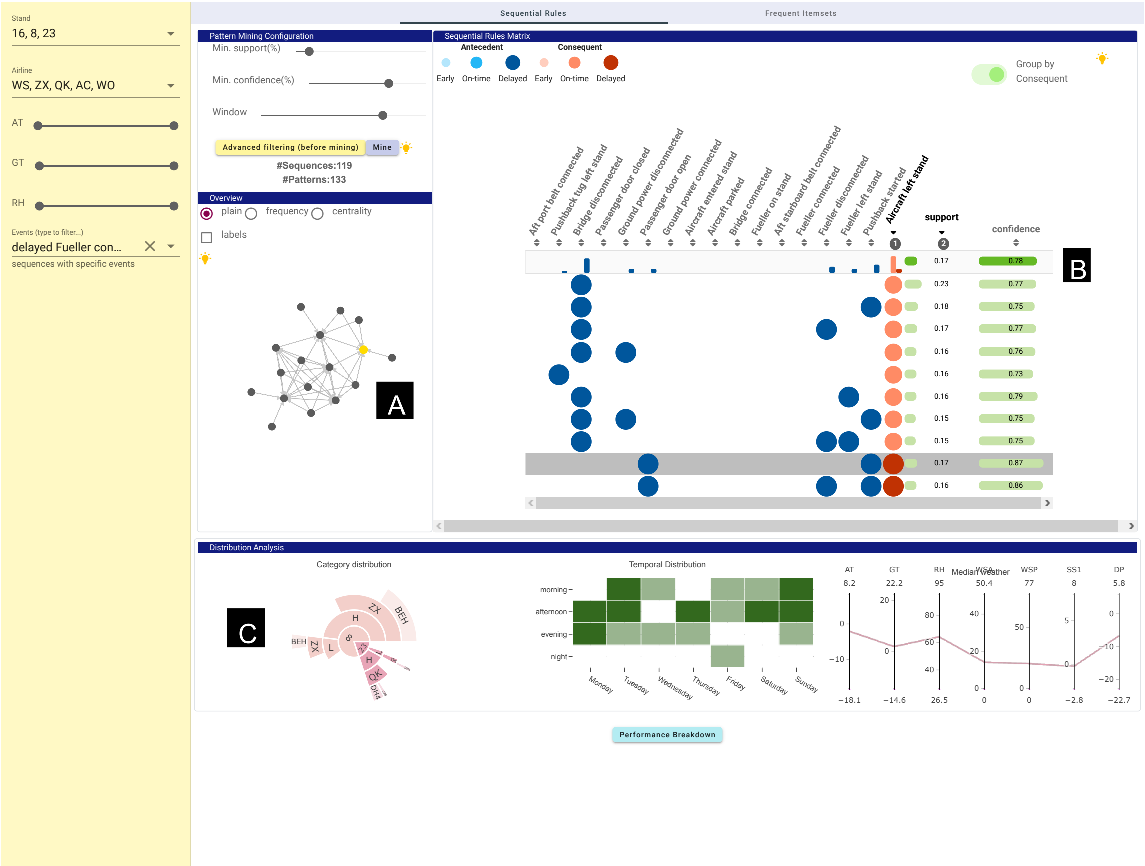


Figure 6.2: Sequential rules grouped by consequent. The big red rectangle shows the aggregated antecedents of *Fueller connected*, while the blue one shows all the groups in which this event appears an antecedent.

6.2 Frequent Itemsets from Sequences in Freezing Weather

This use case presents Bob, a manager working at an airport situated in a climate with long snowy winters and frequent cold-season storms. Bob wants to know how freezing weather is affecting the intervals between ground handling operations. He first filters the turnarounds having both ground and air temperature below 0 $°C$ (Fig.6.4(A)), then he mines itemsets with minimum support of 15% (fig.6.4(B)). He only wants to explore the patterns that took place in at least 25 flights since anything less than that could be due to chance.

In the pattern matrix, he sees a group with only *long* labels, indicating a group of inefficient turnarounds (Fig.6.4(C)). Once he collapses the group rows, he can see an itemset of intervals {*Aircraft parked-Bridge connected, Fueller on stand-Fueller connected*} all labelled as *long*, with average support of 15%. Bob interprets this pattern as follows: A group of flights took a long time for the bridge to get connected to the aircraft after arrival. Also, the fueller did not get connected to the aircraft for a long time after it had entered the stand. Bob investigated one of the patterns and found out that it happened at stands 16 and 23, mostly on Tuesday mornings (from



Stand: 16, 8, 23

Airline: WS, ZX, QK, AC, WO

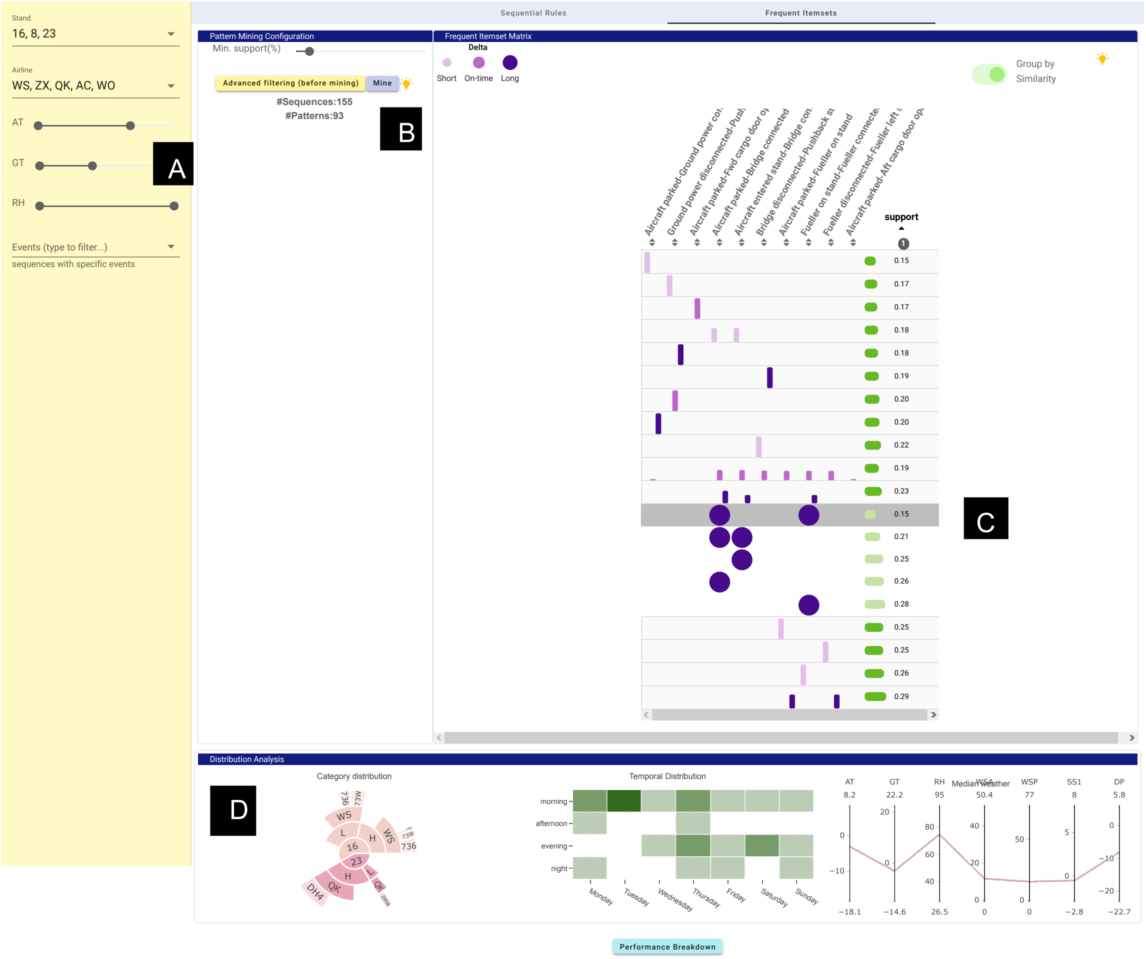
Events (type to filter...): delayed Fueller con...

#Sequences:119

#Patterns:133

Figure 6.3: Target based analysis for delayed aircraft fuelling. Investigating *Aircraft left stand* patterns. (A) Target event is selected from the overview graph (yellow node). (B) The group with target event as the cosequent shows a dark orange bar in the corresponding cell, hinting a delay-related pattern. Group is collapsed and rows are sorted by support and confidence. (C) Attribute distribution for the selected pattern.

4:00 A.M. to 11:59 A.M.) (Fig.6.4(D)). Having access to more turnaround details, Bob could look for causes of disruption in these flights and inspect if there is a common technical issue affecting them.



Stand: 16, 8, 23

Airline: WS, ZX, QK, AC, WO

#Sequences:155
#Patterns:93

Sequential Rules
Frequent Itemsets
AT
GT
RH
Events (type to filter...)
sequences with specific events
Pattern Mining Configuration
Min. support(%)
Advanced filtering (before mining)
Mine
Frequent Itemset Matrix
Delta
Short
On-time
Long
Group by Similarity
Aircraft parked-Ground power co...
Ground power disconnected-Push...
Aircraft parked-Fwd cargo door o...
Aircraft parked-Bridge connected
Aircraft entered stand-Bridge con...
Bridge disconnected-Pushback st...
Aircraft parked-Fueller on stand
Fueller on stand-Fueller connecte...
Fueller disconnected-Fueller left s...
Aircraft parked-Aft cargo door op...
support
0.15
0.17
0.17
0.18
0.18
0.19
0.20
0.20
0.22
0.19
0.23
0.15
0.21
0.25
0.26
0.28
0.25
0.25
0.26
0.29
Distribution Analysis
Category distribution
Temporal Distribution
morning
afternoon
evening
night
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Median Weather
AT 8.2 −18.1
GT 22.2 −14.6
RH 95 26.5
WSA 50.4 0
WSP 77 0
SS1 8 −2.8
DP 5.8 −22.7
Performance Breakdown
A
B
C
D

Figure 6.4: Exploring frequent itemsets derived from sequences in freezing weather. (A) The itemsets are filtered by weather conditions. (B) FIM parameter is set (C) Group with multiple *long* time gaps is collapsed. A pattern is selected from the matrix. (D) Attribute distribution analysis for the selected pattern.

Chapter 7

Evaluation

To evaluate the system usability, we conducted individual interviews with two aviation domain experts, along with a user study on a group of non-expert volunteers. Through the rest of this chapter, we will describe the evaluation steps.

7.1 Domain Expert Interviews

To get the domain experts' feedback on the SeRVis framework and its usability in aviation, we interviewed two domain experts from the airport that provided us with the ground handling logs. Their field of expertise is management and supervision of air service operations. Note that the interviewees were not the same experts whom we consulted during the study. The interview sessions included around 10-15 minutes presentation on an overview of the study and a brief background on FIM and SRM. Then, we demonstrated the use cases in chapter 6 in about 10 minutes. After the presentations, we asked for their feedback on the system. We asked if they found the patterns in use cases insightful, whether they would be interested in using SeRVis for day-to-day operations and if they had any suggestions to improve the system. Each of the interview sessions took about 30 minutes in total. Ideally, we would have had a longer evaluation session to train the domain experts on how to use the system. Then, ask them to perform exploratory analysis and find patterns that they find insightful. However, due to time constraints of the experts, we changed the domain expert evaluation strategy. We summarize the interview results in the two following sections:

7.1.1 Usability

The experts stated that the framework could potentially provide valuable information for the airport and the airlines. According to one of the experts, the airport collects a lot of data but does not analyze it to the level of detail presented in this study.

Currently, they only look at how often a particular flight number or a route number is delayed, but the airlines would be interested in knowing the causes. Whenever there are delays, the airport is concerned with understanding the responsible parties (for example, the ground handlers, airlines, or airport), or potential facility issues. Thus, a system that could help them extract such information is beneficial. Furthermore, the experts were interested in the attribute distribution analysis about the gates and aircraft type. For assigning each aircraft to a specific gate, knowing which type of aircrafts performs worse at which gates can help them with a more efficient assignment. One of the experts mentioned that they would rather have the over-night flights not removed in the preprocessing step since that kind of flight can influence delays in morning turnarounds, and it would be valuable to analyze their influence.

We discussed the lack of full operation history per flight in the data and not having access to SLA information. The experts agreed that if the dataset could be expanded with boarding and resource information, the system may help find insights to improve performance time. For example, different airlines have contracts with different companies for ground handling operations. There is a preparation time before each turnaround for moving the companies' equipment to the designated stand, and analyzing this stage is crucial since it can be time-consuming. The dataset could be expanded to include these details as well for more fruitful analysis. Moreover, SLA standards vary among airlines, which should be incorporated in the labelling. For example, we labelled some arrivals as delayed since they were 15 minutes late, but this threshold can be 8 minutes for some airlines.

7.1.2 User Interface

The experts suggested several changes in the user interface. One of the things both of the experts pointed out was the time unit in the detail table (Fig.5.1 (F)), which is reported in seconds. They suggested changing the unit to minutes or hours, which is more convenient for their use. The support and confidence in the matrix could be reported in percentages instead of ratio, which is more suitable for reporting. The calendar view could be improved to display the delays as well. For example, red could determine delays or interruptions in a day, while green can stand for expected performance. Lastly, the weather information can include labels such as *snowing* and *raining*, or more information is provided to the user on how to interpret the numbers.

7.2 User Test

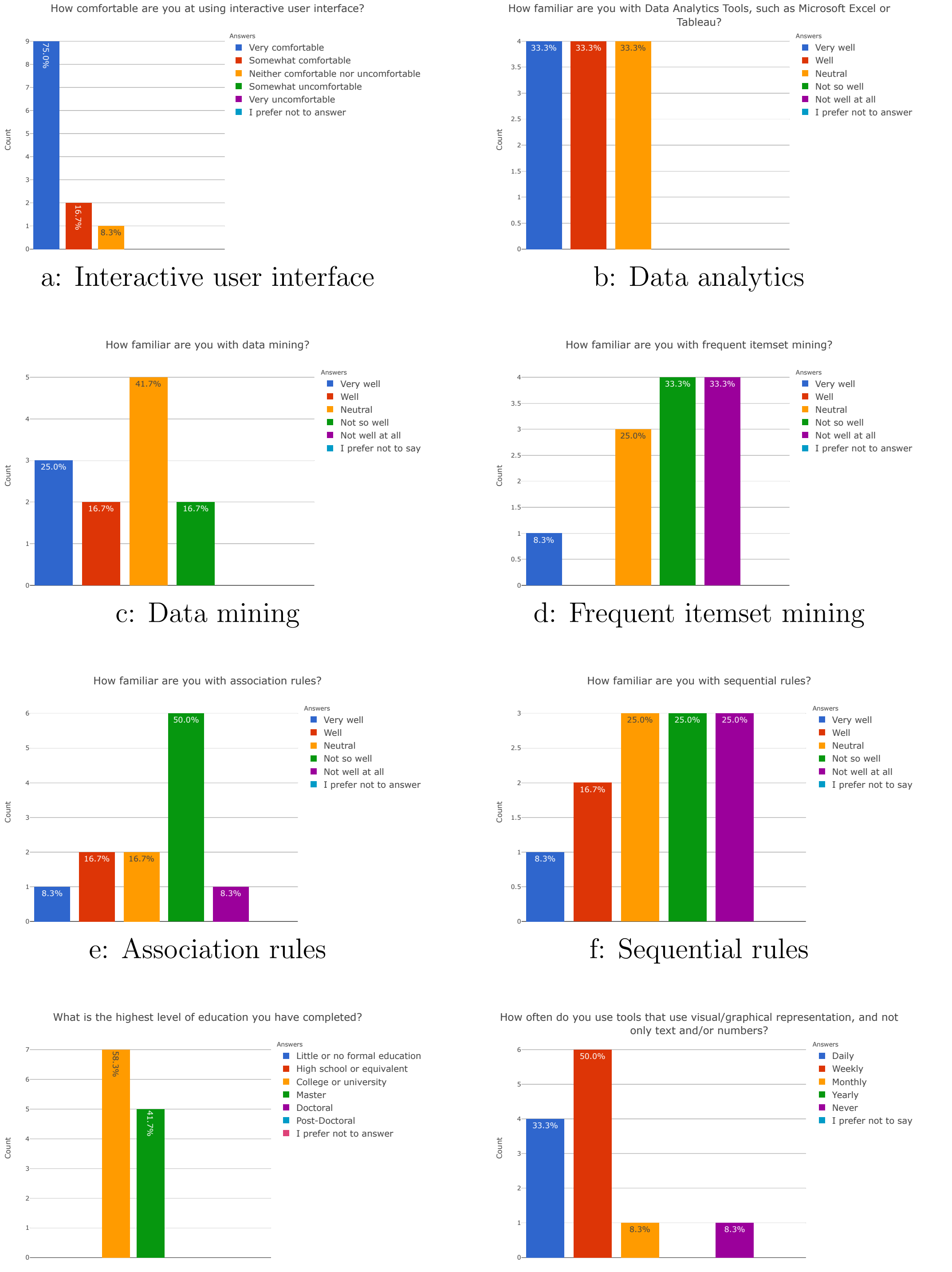
We conducted a user study to assess the system's usability, possible improvements and whether our proposed visual framework reduces the cognitive load of exploring patterns. The study was conducted online and anonymously using Microsoft Forms. The volunteers were given a series of video tutorials on the data mining concepts and how to interact with the system. Then, they had to perform several analytical tasks to answer the scenario-based questions. Finally, they were asked to rate the system from different aspects. The average time for completing the test was 73 minutes.

7.2.1 Participant Selection

We looked for volunteers by sending a recruitment notice to computer science students at Dalhousie University since they are already familiar with interactive software. We were also expecting the computer science students to be familiar with the data mining concepts and require less training for using the prototype. The notice was emailed to students through *csjobs@cs.dal.ca*, which all the computer science students are automatically subscribed to it. Ideally, we were looking for 30 participants, but only 19 students volunteered, and 13 of them completed the study. A possible reason for not completing the user test could be the long average duration of completing the survey.

We discarded one of the inputs where the volunteer had copy-pasted the question in the text answers or had left them blank and had chosen *Quit* for the multi-choice questions and selected the *neutral* option for the majority of the Likert questions.

The demographic questionnaire results reveal that users were familiar and comfortable with using visualizations and interactive user interface (Fig. 7.1). Seven volunteers had completed the bachelors, and the other five had completed masters. Only a few users were familiar with data mining concepts used in the study.



a: Interactive user interface

b: Data analytics

c: Data mining

d: Frequent itemset mining

e: Association rules

f: Sequential rules

g: Education

h: Using visual/graphical representation

Figure 7.1: User study demographics revealing familiarity of the users with different topics used in the study and their education level

7.2.2 Experiment Setup

We invited the participants through the *csjobs@cs.dal.ca* email list and sent the consent form to whoever responded to the notice. The consent form described the study procedure, and the volunteers could read it to decide if they wanted to proceed with the study. The consent form is found in appendix A. A link to the Microsoft form and login credentials for using the SeRVis website was provided to those who consented to participate.

7.2.3 Training

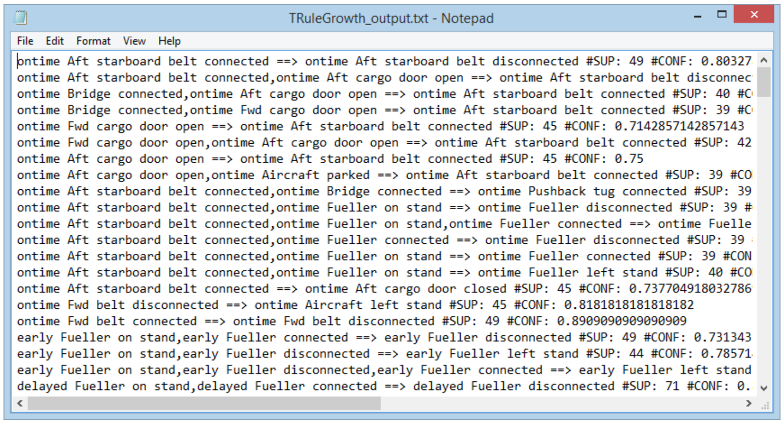
Once the users answer a basic demographic questionnaire (Fig. 7.1), they are provided with a series of short one-minute tutorials on FIM, SRM, filtering sequences and mining patterns, how to use the pattern matrix, and how to select a row and analyze the attribute distributions and pattern details. Furthermore, they are provided with brief definitions of the airport apron, stand and ground handling operations to get familiar with the terms used in the analytical questions.

7.2.4 Scenario Exercises

The exercises were designed for two objectives. First, assess our hypothesis that a visual analytics framework will reduce the cognitive load of exploring patterns. Second, the usability of framework design. For the first part, we designed two similar sets of three questions. The user is asked to answer the first set with plain-text output of a popular pattern mining library that is merely a list of patterns. The second set of questions should be answered using the visual prototype. After each set of questions, the user is asked to answer the NASA-TLX [36] to measure their workload. Ideally, we expect to see a lower (mental) workload for the visual prototype. The second part of the assessment is a series of tasks targeting design goals **G2-G5** to see if the users can use all the features properly.

Exercise Rationale

The first three questions assess the users' ability to answer basic data mining questions from the plain-text output of a popular off-the-shelf pattern mining library [18] (Fig.



```
ontime Aft starboard belt connected ==> ontime Aft starboard belt disconnected #SUP: 49 #CONF: 0.80327
ontime Aft starboard belt connected,ontime Aft cargo door open ==> ontime Aft starboard belt disconnec
ontime Bridge connected,ontime Aft cargo door open ==> ontime Aft starboard belt connected #SUP: 40 #C
ontime Bridge connected,ontime Fwd cargo door open ==> ontime Aft starboard belt connected #SUP: 39 #C
ontime Fwd cargo door open ==> ontime Aft starboard belt connected #SUP: 45 #CONF: 0.7142857142857143
ontime Fwd cargo door open,ontime Aft cargo door open ==> ontime Aft starboard belt connected #SUP: 42
ontime Aft cargo door open ==> ontime Aft starboard belt connected #SUP: 45 #CONF: 0.75
ontime Aft cargo door open,ontime Aircraft parked ==> ontime Aft starboard belt connected #SUP: 39 #CO
ontime Aft starboard belt connected,ontime Bridge connected ==> ontime Pushback tug connected #SUP: 39
ontime Aft starboard belt connected,ontime Fueller on stand ==> ontime Fueller disconnected #SUP: 39 #
ontime Aft starboard belt connected,ontime Fueller on stand,ontime Fueller connected ==> ontime Fuelle
ontime Aft starboard belt connected,ontime Fueller connected ==> ontime Fueller disconnected #SUP: 39
ontime Aft starboard belt connected,ontime Fueller on stand ==> ontime Fueller connected #SUP: 39 #CON
ontime Aft starboard belt connected,ontime Fueller on stand ==> ontime Fueller left stand #SUP: 40 #CO
ontime Aft starboard belt connected ==> ontime Aft cargo door closed #SUP: 45 #CONF: 0.737704918032786
ontime Fwd belt disconnected ==> ontime Aircraft left stand #SUP: 45 #CONF: 0.8181818181818182
ontime Fwd belt connected ==> ontime Fwd belt disconnected #SUP: 49 #CONF: 0.8909090909090909
early Fueller on stand,early Fueller connected ==> early Fueller disconnected #SUP: 49 #CONF: 0.731343
early Fueller on stand,early Fueller disconnected ==> early Fueller left stand #SUP: 44 #CONF: 0.78571
early Fueller on stand,early Fueller disconnected,early Fueller connected ==> early Fueller left stand
delayed Fueller on stand,delayed Fueller connected ==> delayed Fueller disconnected #SUP: 71 #CONF: 0.
```

Figure 7.2: SPMF[18] plain-text output for mining sequential rules from 388 sequences, with minimum support = 10%, minimum confidence = 70%, and window size = 15.

7.2). The patterns were already mined, and the result was displayed to the user with a browser. The users were asked to copy-paste the entire line with the correct answer for convenience. When we evaluated the responses, we noticed some users wrote only one part of the rule (for example, without the support and confidence) or had added explanations to their answer. We edited the questionnaire key to accept all possible correct answers. The questions about the patterns in pure text format are the following:

Q1 What is the most frequent sequential rule? (highest support)?

Q2 What is the rule showing the most probable causes of “ontime Fueller connected”?

Q3 Can you find an item called “ontime Aircraft parked-Fueller on stand”? What other item(s) most frequently appear with it?

These questions were followed by a NASA-TLX questionnaire. Then, another set of similar questions, listed below, were asked but this time the users were asked to use the visual prototype. Note that these questions inherently assess design goals **G3** and **G4** as well. We want to see if the users can easily identify antecedents and consequents, interest measures, and itemsets having specific items. After answering the second set of questions, the users were asked to fill the NASA-TLX questionnaire once again.

Q4 Which of the following rules has perfect confidence $X \rightarrow Y, P(X|Y) = 1$?

Q5 Which of the following set of events are most frequently lead to "on-time Bridge disconnected"?

Q6 Can you find an item called "ontime Aircraft parked-Fueller on stand"? What other item(s) most frequently appear with it?

For evaluating **G2**, we asked the following two questions to see if the user can use the grouped mode properly and read the aggregated antecedents for a given consequent or find the groups where an item appears as an antecedent.

Q7 In the grouped matrix (Group by consequence switch on), which of the following effect "Fueller connected"?

Q8 In the grouped matrix (Group by consequence switch on), which of the following does "Fueller connected" effect?

Lastly, to evaluate whether the user can use the features related to **G5**, we asked the following questions. Note that we also asked the user to perform a sequence filtering by setting air and ground temperature to a certain range and also configuring the mining parameters.

Q9 How many sequences were filtered?

Q10 In the following rule: "delayed Passenger door closed → delayed Pushback started support: 0.22 confidence: 0.86", what can we understand with distribution analysis?

Q11 How many of the sequences with the rule you found in the previous question have "delayed fueller on stand" in their sequences?

Results

We first report how many of the volunteers answered the analytical questions correctly, and then compare the NASA-TLX results for plain-text and visual prototype.

We compared two sets of questions **Q1-Q3** and **Q4-Q6** together (table 7.1). First, we noticed that **Q4** and **Q6** have a higher number of correct responses than **Q1** and

Q3 (75% and 50% vs. 66.7% and 41.7%), which means users were more successful in finding the correct answer using the visual prototype (table 7.1 (a) and (c)).

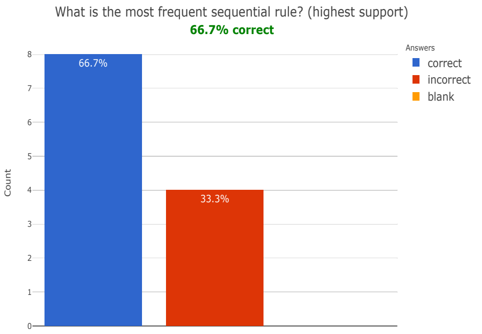
In **Q4**, we expected the users first to ungroup the rows and then find the answer by sorting them by confidence. The wrong answers came from selecting the row with the highest average confidence in grouped mode. One possible reason could be that the users assumed that high average confidence among all groups means the rules in that group must have the highest confidence overall, which was not the case. Or they forgot about what support and confidence values mean in the grouped mode, even though we mentioned that in the Microsoft Forms before tasks. One possible solution is training with more examples and explanations.

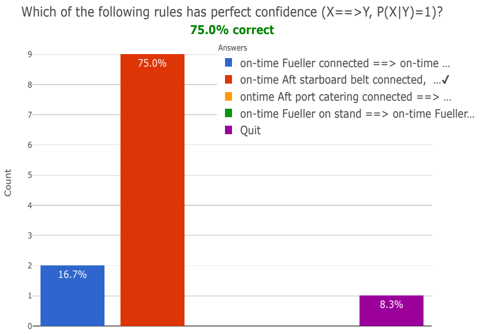
In **Q5** and **Q6**, we were expecting the users to use the multi-sort feature of the matrices and sort the rows based on the event/item in the questions, and then by support. When we analyzed the wrong answers, we noticed that most of the incorrect answers are due to not sorting the rows by the support and selecting the first row with the requested labelled event/item. Perhaps the users assumed that the rows are sorted by support by default.

Fig. 7.3 shows the result of the last set of analytical questions **Q7**-**Q11**. Questions **Q7** and **Q8** had multi-choice answers, and most users were able to find the correct answers. These questions required the users to change the default mining parameters and set the minimum support higher before answering the questions. We guess that the wrong answers for **Q8** might be due to not changing the default parameters and including rules with lower support.

Q9 was a sanity check to see if the users could filter the sequences properly for answering the last three analytical questions. The majority of the wrong answers for **Q10** was due to not selecting the specific pattern row and selecting the answer that corresponds to the aggregated attribute distribution for all the filtered sequences. Ten users answered **Q11** since it was optional, and people who were able to select the pattern from the matrix row successfully were also able to answer this question correctly. Even though we covered the pattern selection in the tutorial videos, perhaps the system required more training before use.

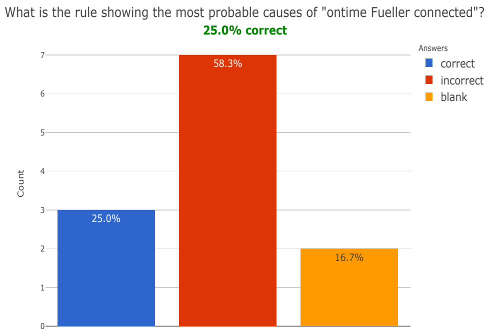
The comparison of NASA-TLX questionnaire shows that overall, the visual prototype did reduce the cognitive load of pattern exploration (Fig. 7.4).

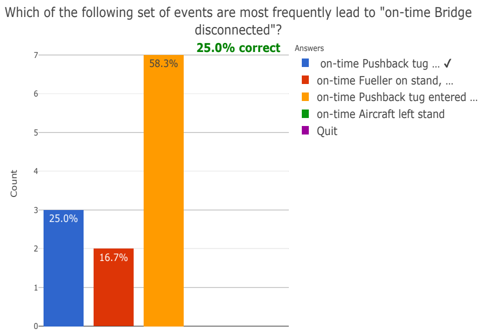
<table>
<tr><th>Text Question</th><th>SeRVis Question</th></tr>
<tr><td>

Q1</td><td>

Q4</td></tr>
</table>

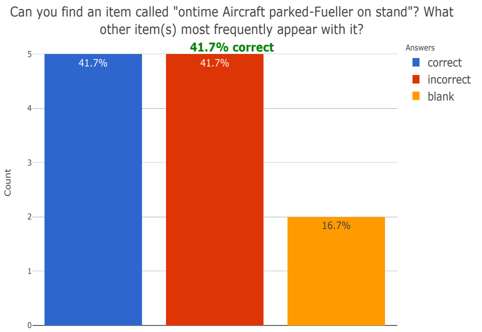
(a) Finding highest support/confidence

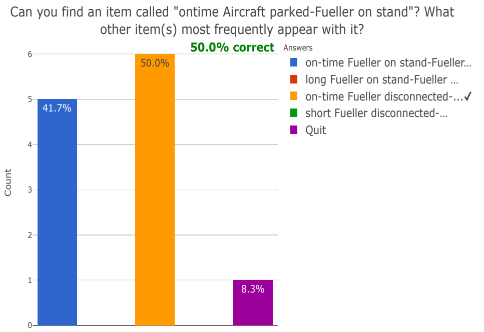
<table>
<tr><th>Text Question</th><th>SeRVis Question</th></tr>
<tr><td>

Q2</td><td>

Q5</td></tr>
</table>

(b) Finding antecedents/consequents

<table>
<tr><th>Text Question</th><th>SeRVis Question</th></tr>
<tr><td>

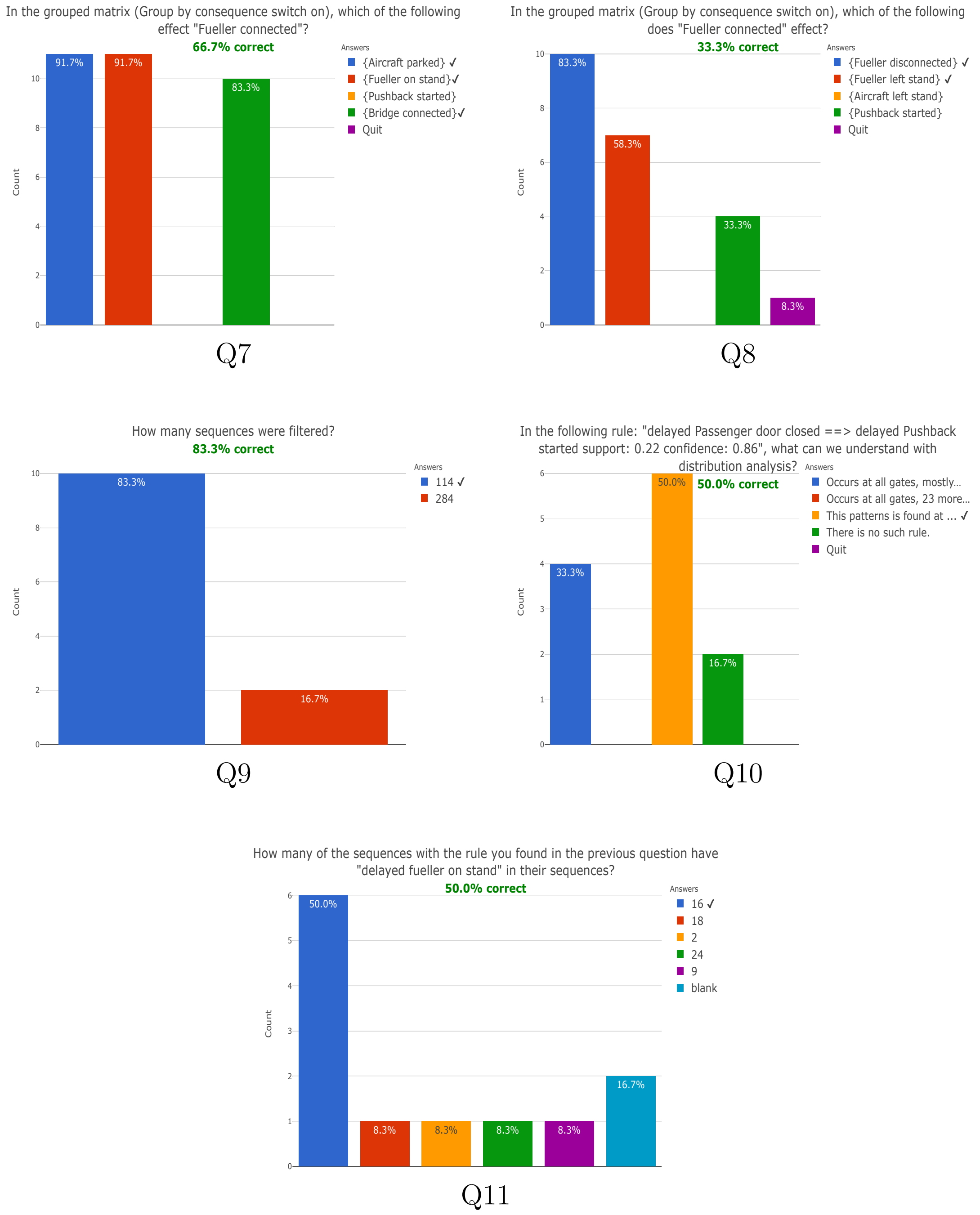
Q3</td><td>

Q6</td></tr>
</table>

(c) Finding certain co-occurrence

Table 7.1: User study results for Q1-Q6. Each pair of questions assess a specific analytical task where the users have to answer using two different tools.

To assess if the difference in the overall average of the NASA-TLX results is statistically significant, we used the paired t-test [44] since we have same group of



Q7

Q8

Q9

Q10

Q11

Figure 7.3: User study results for Q7-Q11. For Q7 and Q8, the correct percentage (reported with green text) represents the percentage of the users that selected all the correct answers; partial selections do not count.

subjects and we want to compare the their input from two different conditions. To this end, we first need to make sure the amount of change in samples (NASA-TLX

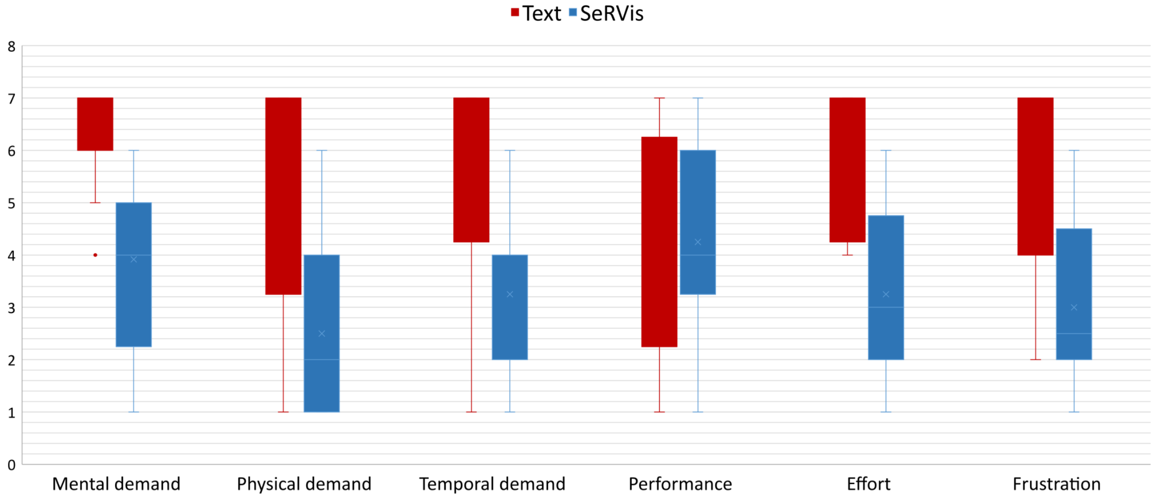


Figure 7.4: NASA-TLX results comparison for plain-text output vs. SeRVis visual prototype. The answers were on scale of 1 (=very low) to 7(=very high).

results from text and visualization) are normally distributed and both samples have the same covariance to satisfy paired t-test assumptions [44]. For the former we used the Shapiro-Wilk normality test w.r.t. to our sample size [59]. For the latter, we used the Levene's test [62]. Levene's test is robust to non-normally-distributed samples. Both of our individual samples were not normally distributed when we checked the Shapiro-Wilk test, however for the paired t-test, the difference in the amount of change between each pair of samples is normally distributed 7.2.

For each pair of the questions, our hypothesis is that using a visual prototype will reduce the workload compared to using a text-based tool. Once we make sure that the assumptions of paired t-test are satisfied, we conducted the t-test. As we see in table 7.2, out hypothesis is accepted in all cases except *Performance* where the users had to rate how well they thought they had executed the tasks. One possible reason for same average could be that most users were not familiar with data mining concepts before the study (Fig. 7.1), thus were not sure if they have fulfilled what was asked from them.

7.2.5 Questionnaire

After the analytical questions, the users were asked to answer two Likert questionnaires to rate the interface and the software usability [8]. The users could optionally

Tool	Mean	Standard Deviation		
Text (T)	6.42	0.95		
SeRVis (S)	3.92	1.55		
Hypothesis	**Test**	**Statistics**	**P**	**Hypothesis Accepted?**
T-S is normally distributed	Shapiro-Wilks	0.95	0.65	Yes
T and S have the same covariance	Levene	2.77	0.11	Yes
T and S averages are significantly different	Paired t-test	3.95	0.00	Yes

(a) Mental demand

Tool	Mean	Standard Deviation		
Text (T)	4.75	2.13		
SeRVis (S)	2.50	1.55		
Hypothesis	**Test**	**Statistics**	**P**	**Hypothesis Accepted?**
T-S is normally distributed	Shapiro-Wilks	0.94	0.47	Yes
T and S have the same covariance	Levene	1.32	0.26	Yes
T and S averages are significantly different	Paired t-test	3.08	0.01	Yes

(b) Physical demand

Tool	Mean	Standard Deviation		
Text (T)	5.42	1.71		
SeRVis (S)	3.25	1.53		
Hypothesis	**Test**	**Statistics**	**P**	**Hypothesis Accepted?**
T-S is normally distributed	Shapiro-Wilks	0.92	0.33	Yes
T and S have the same covariance	Levene	0.00	1.00	Yes
T and S averages are significantly different	Paired t-test	3.22	0.01	Yes

(c) Temporal demand

Tool	Mean	Standard Deviation		
Text (T)	3.83	1.99		
SeRVis (S)	4.25	1.83		
Hypothesis	**Test**	**Statistics**	**P**	**Hypothesis Accepted?**
T-S is normally distributed	Shapiro-Wilks	0.94	0.55	Yes
T and S have the same covariance	Levene	0.02	0.89	Yes
T and S averages are significantly different	Paired t-test	-0.62	0.55	No

(d) Performance

Tool	Mean	Standard Deviation		
Text (T)	5.92	1.26		
SeRVis (S)	3.25	1.69		
Hypothesis	**Test**	**Statistics**	**P**	**Hypothesis Accepted?**
T-S is normally distributed	Shapiro-Wilks	0.94	0.47	Yes
T and S have the same covariance	Levene	0.74	0.40	Yes
T and S averages are significantly different	Paired t-test	3.75	0.00	Yes

(e) Effort

Tool	Mean	Standard Deviation		
Text (T)	5.50	1.71		
SeRVis (S)	3.00	1.68		
Hypothesis	**Test**	**Statistics**	**P**	**Hypothesis Accepted?**
T-S is normally distributed	Shapiro-Wilks	0.89	0.13	Yes
T and S have the same covariance	Levene	0.00	1.00	Yes
T and S averages are significantly different	Paired t-test	3.12	0.01	Yes

(f) Frustration

Table 7.2: NASA-TLX t-test to investigate statistical significance. α is considered 0.05 for to compare the p-values.

answer two open-ended questions to comment on the system.

Results

Overview of the results is found in Fig. 7.5. One of the things users have rated and is also reflected in their answers is that they have found the system challenging to learn.

We even received a comment regarding the learning curve in the open-ended questions at the end of the survey: *"This System is new and initially It took me a bit of time to get a hang of it, but it was fairly easy to understand it"*. Or another comment stated that working with groups was confusing: *"The grouping was confusing and it was difficult to locate a certain item in the matrix"*. This can explain why the users did not perform well in the tasks that required them to work with the grouping feature. We conclude that the prototype requires more training for the users to grasp all features fully.

The users responded positively to the design choices such as glyphs and groupings. They also agreed that the visual prototype is easier to use compared to a text-based list of patterns. More than half of the users did not find the use of a variety of visual components overwhelming (Fig. 7.5 (a)).

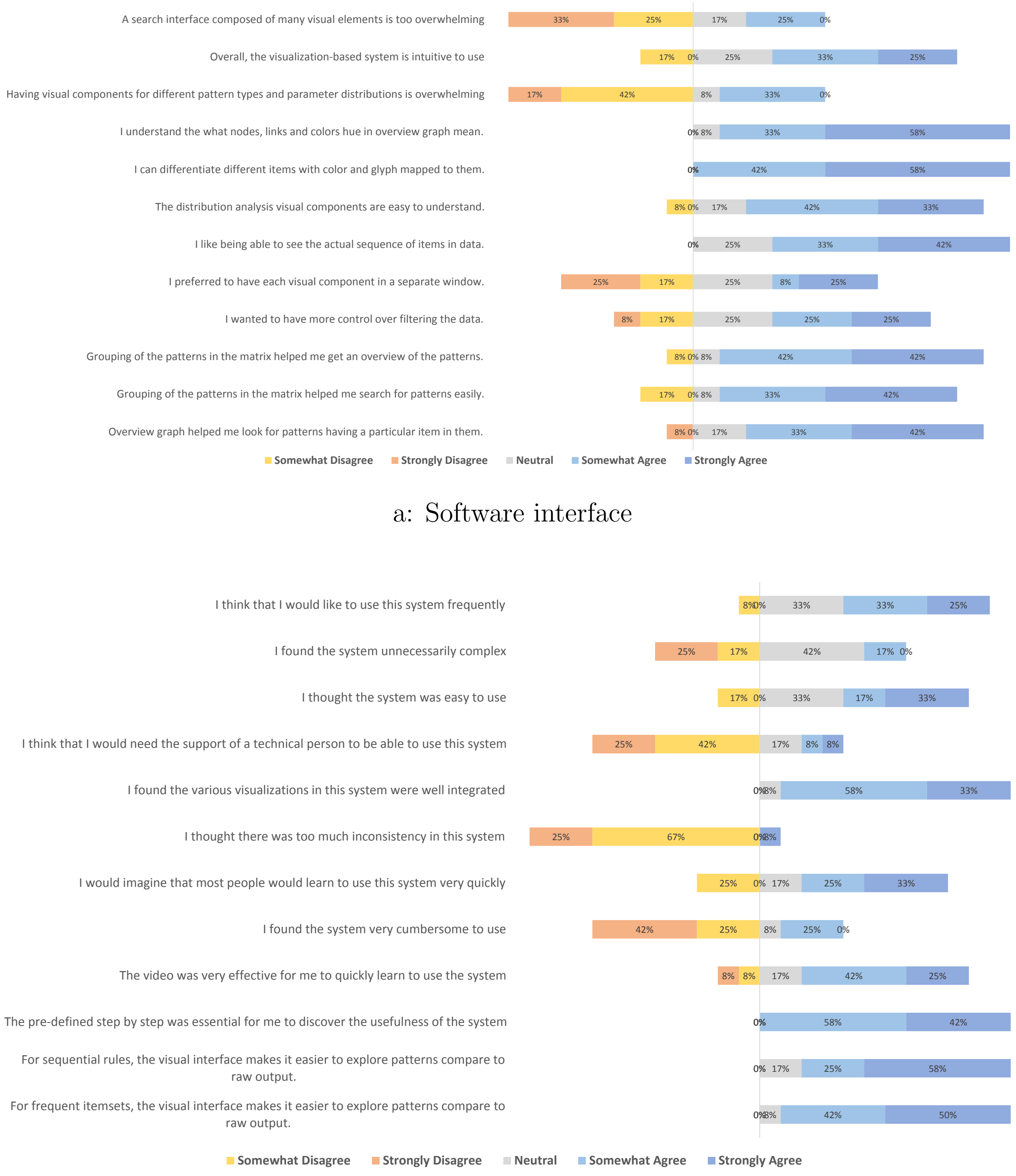
We also received several suggestions through open-ended questions. One request was to add a reset button for the default configurations. One of the participants mentioned that they did not use the overview graph and its labels get cluttered and hard to read. Another user demanded the interface to be more interactive and seamless. One use suggested highlighting the entire column rather than having tooltips to show them the glyphs' name. They also believed tooltips could provide more information other than the event/item names.

We received some compliments on the visual appeal of the framework including *"A very neat system with good UX*[1]*."* and *"The system was not only useful but also visually attractive."*

7.2.6 Discussion

We were able to show that users experienced a lower workload when they worked with a visual prototype compared to using plain text output. Most of the users were able to use the features in SeRVis. As we mentioned, the study was conducted online and there was no interaction with the users and they completed the study on their own using video tutorials. Perhaps having in-person training sessions, or longer online session with practice sessions and more examples for the users could improve the results.

[1] User Experience



a: Software interface

b: Software usability [8]

Figure 7.5: Software interface and usability ratings

Chapter 8

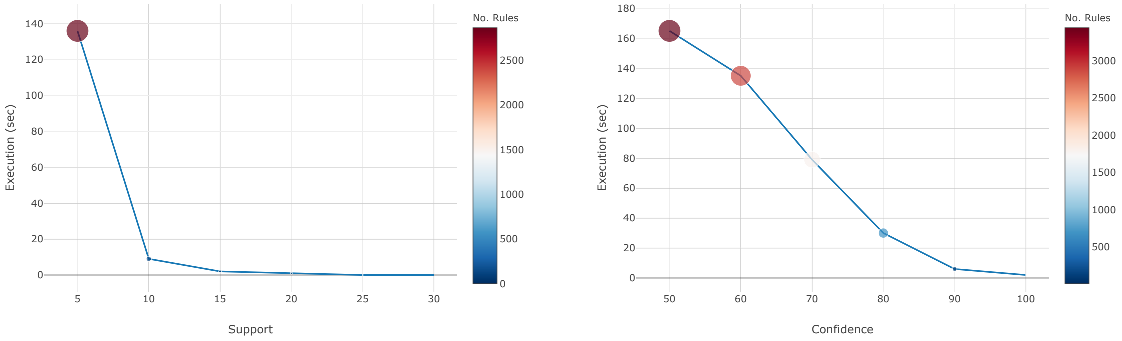
Conclusion

In this study, we developed a visual analytics framework to explore sequential rules and frequent itemsets in a sequence dataset of ground handling operations at an airport. We formulated our analytical requirements based on discussions with domain experts to design the framework accordingly. Following the design goals, we created the SeRViz that allows the user to mine frequent patterns with desired configurations and displays the output in an interactive matrix-based visualization. We evaluated the efficacy of our framework by conducting two use cases on the dataset. Further study is required to examine the effectiveness of this tool in aviation, using fine-grained data and larger datasets. In the following sections, we describe the limitations and ideas for system improvement in future work.

8.1 Limitations and Future Work

8.1.1 Performance

The SPMF library used for pattern mining does not provide a feature for the user to know each pattern is mined from which sequences/itemsets. Therefore, we needed to add a post-processing level on top of pattern mining to find the associated sequences/itemsets for each pattern. For the sequential rules, we have two other layers of post-processing for removing redundancies, and building the DAGs described in section 5.3.3. During our experiments, we noticed that these layers of post-processing increase the time complexity of pattern mining and adversely affect the user experience. As we can see in Fig. 8.1, as long as the number of rules is not in order of a few hundred, the pattern mining user experience is not seamless. To overcome this problem and improve the performance, we plan to combine pattern mining and post-processing into one algorithm.



a: Execution time vs. support

b: Execution time vs. confidence

Figure 8.1: System performance. We timed the backend processing time for mining the sequential patterns, finding sequences of each pattern, redundancy removal and DAG generation. (a) confidence = 60% and window = 12 (b) support = 10% and window = 12. The experiments were conducted on a 64-bit operating system with 8 GB of RAM and CPU with 2.50 GHz clock speed.

8.1.2 Scalability

We discuss the visual scalability for each of the sequential rule and frequent itemset matrices.

In the sequential rules matrix, the number of rows and columns in grouped mode is equal to the number of unique events in the worst-case scenario. In ungrouped mode, number of rows are equal to the number of patterns. Features such as filtering, sorting, and increasing support and confidence can help, but it still remains a challenge as the number of patterns grows. For datasets with a higher number of unique events, we can try grouping columns, for example, by category, for the matrix's horizontal scalability. For the vertical scalability, we can try grouping rows by more constraints compared to the current design.

In the frequent itemsets matrix, our major challenge is the vertical scalability in ungrouped mode, as the number of unique items (in our target domain, it is a predefined set of deltas requested by the domain experts) is low. Using the grouped mode and only collapsing rows of a group of interest, and the multi-sort feature can curb this challenge.

8.1.3 Accuracy of the Analysis

One major limitation we had for this study is that we did not have access to the airport-airline SLAs due to confidentiality constraints. Each airline has its specific business rules/procedures for ground handling operations. For example, some airlines expect the fueller to connect and start fuelling the aircraft in one minute after the fueller's arrival at the stand. According to one of the domain experts, the most valuable information can be derived from comparing airline operations schedule against the actual turnaround and looking for potential SLA violations by handlers such as fueler/caterer, and so on. In the absence of such information, we did not know how accurately SLA was performed for each turnaround. Thus, we looked at the statistical distribution of each event's occurrence w.r.t. to a starting point (e.g. arrival). Consequently, the labelling might be inaccurate from an aviation perspective. We can improve this by incorporating the airline countdowns and constraints in data.

In this paper, we proposed a method for visualization of sequential rules which has the potential of revealing cause-effect relationships in data [37]. However, one must be careful that association rules and Causal Ruless (CRs) are not interchangeable; although CRs imply association, the reverse may not always be true [48]. Thus this system is not applicable for cases where the user is solely interested in casual relationships. Furthermore, particularly for our main target domain, the turnarounds' punctuality is influenced by many external factors such as delayed flight cycles, technical malfunctions, and so on. [25]. Hence, when we look at a pattern that, for example, demonstrates that specific antecedents lead to a delayed event, there might be reasons beyond those antecedents contributing to that delay. More accurate analysis might be possible with more details about the turnaround history in data.

8.1.4 Rare Patterns

Another limitation is that rare but interesting patterns might be filtered out by support/confidence threshold. A workaround could be using algorithms specifically designed for finding rare patterns [15].

8.1.5 Sequential Rule Simplification

For simplicity, we mined the rules with only one consequent. We explicitly set the maximum number of consequents to one in the TRuleGrowth algorithms and avoided discarding any pattern in the post-processing. We plan to extend the use of prototype to rules with more than one consequent.

Bibliography

[1] EUROCONTROL ATM Lexicon. `https://ext.eurocontrol.int/lexicon/index.php/Main_Page`,note = Accessed: 2021-04-11.

[2] Charu C. Aggarwal and Jiawei Han, editors. *Frequent Pattern Mining*. Springer, 2014.

[3] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

[4] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter Rodgers. The state-of-the-art of set visualization. In *Computer Graphics Forum*, volume 35, pages 234–260. Wiley Online Library, 2016.

[5] Fatima Zohra Benhacine, Baghdad Atmani, and Fawzia Zohra Abdelouhab. Contribution to the association rules visualization for decision support: a combined use between boolean modeling and the colored 2d matrix. *IJIMAI*, 5(5):38–47, 2019.

[6] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.

[7] Gwenael Bothorel, Mathieu Serrurier, and Christophe Hurter. Visualization of frequent itemsets with nested circular layout and bundling algorithm. In *International Symposium on Visual Computing*, pages 396–405. Springer, 2013.

[8] John Brooke. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189, 1996.

[9] Bram CM Cappers and Jarke J van Wijk. Exploring multivariate event sequences using rules, aggregations, and selections. *IEEE transactions on visualization and computer graphics*, 24(1):532–541, 2017.

[10] Sharma Chakravarthy and Hongen Zhang. Visualization of association rules over relational dbmss. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 922–926, 2003.

[11] Wei Chen, Cong Xie, Pingping Shang, and Qunsheng Peng. Visual analysis of user-driven association rule mining. *Journal of Visual Languages & Computing*, 42:76–85, 2017.

[12] Yuanzhe Chen, Panpan Xu, and Liu Ren. Sequence synopsis: Optimize visual summary of temporal event data. *IEEE transactions on visualization and computer graphics*, 24(1):45–55, 2017.

[13] International Business Machines Corporation. IBM Java-based Mining. `https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.5.0/com.ibm.im.visual.doc/c_visualminingjava.html`,note = Accessed: 2021-04-11.

[14] International Business Machines Corporation. IBM sequence rules view. `https://www.ibm.com/support/knowledgecenter/bg/SSEPGG_9.7.0/com.ibm.im.visual.doc/c_the_sequences_rules_view.html?view=embed`,note = Accessed: 2021-04-11.

[15] Sadeq Darrab, David Broneske, and Gunter Saake. RPP Algorithm: A Method for Discovering Interesting Rare Itemsets. In *International Conference on Data Mining and Big Data*, pages 14–25. Springer, 2020.

[16] Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, Miha Štajdohar, Lan Umek, Lan Žagar, Jure Žbontar, Marinka Žitnik, and Blaž Zupan. Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, 14:2349–2353, 2013.

[17] Antonio D'Ambrosio and Marcello Pecoraro. Multidimensional scaling as visualization tool of web sequence rules. In *Classification and Multivariate Analysis for Complex Data Structures*, pages 309–316. Springer, 2011.

[18] Philippe Fournier-Viger, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Cheng-Wei Wu, and Vincent S Tseng. SPMF: a Java open-source pattern mining library. *The Journal of Machine Learning Research*, 15(1):3389–3393, 2014.

[19] Philippe Fournier-Viger, Ted Gueniche, and Vincent S Tseng. Using partially-ordered sequential rules to generate more accurate sequence prediction. In *International Conference on Advanced Data Mining and Applications*, pages 431–442. Springer, 2012.

[20] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017.

[21] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Bay Vo, Tin Truong Chi, Ji Zhang, and Hoai Bac Le. A survey of itemset mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(4):e1207, 2017.

[22] Philippe Fournier-Viger and Vincent S Tseng. TNS: mining top-k non-redundant sequential rules. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 164–166, 2013.

[23] Philippe Fournier-Viger, Cheng-Wei Wu, Vincent S Tseng, and Roger Nkambou. Mining sequential rules common to several sequences with the window size constraint. In *Canadian Conference on Artificial Intelligence*, pages 299–304. Springer, 2012.

[24] Matteo Francia, Matteo Golfarelli, and Stefano Rizzi. Summarization and visualization of multi-level and multi-dimensional itemsets. *Information Sciences*, 520:63–85, 2020.

[25] Hartmut Fricke and Michael Schultz. Delay impacts onto turnaround performance. In *ATM Seminar*, 2009.

[26] Magaly Lika Fujimoto, Veronica Oliveira de Carvalho, and Solange Oliveira Rezende. Evaluating generalized association rules combining objective and subjective measures and visualization. In *ICEIS (2)*, pages 285–288, 2009.

[27] Eduard Glatz, Stelios Mavromatidis, Bernhard Ager, and Xenofontas Dimitropoulos. Visualizing big network traffic data using frequent pattern mining and hypergraphs. *Computing*, 96(1):27–38, 2014.

[28] Miguel Grinberg. *Flask web development: developing web applications with python*. O'Reilly Media, Inc., 2018.

[29] Yi Guo, Shunan Guo, Zhuochen Jin, Smiti Kaul, David Gotz, and Nan Cao. Survey on visual analysis of event sequence data. *arXiv preprint arXiv:2006.14291*, 2020.

[30] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[31] Michael Hahsler. arulesviz: Interactive visualization of association rules with r. *R Journal*, 9(2), 2017.

[32] Michael Hahsler and Radoslaw Karpienko. Visualizing association rules in hierarchical groups. *Journal of Business Economics*, 87(3):317–335, 2017.

[33] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[34] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1):53–87, 2004.

[35] Derek Hansen, Ben Shneiderman, and Marc A Smith. *Analyzing social media networks with NodeXL: Insights from a connected world*. Morgan Kaufmann, 2010.

[36] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[37] Hossein Hassani, Xu Huang, and Mansi Ghodsi. Big data and causality. *Annals of Data Science*, 5(2):133–156, 2018.

[38] Martin Hlosta, Michal Šebek, and Jaroslav Zendulka. Approach to visualisation of evolving association rule models. In *2013 Second International Conference on Informatics & Applications (ICIA)*, pages 47–52. IEEE, 2013.

[39] Patrick Hoffman, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley. Dna visual and analytic data mining. In *Proceedings. Visualization'97 (Cat. No. 97CB36155)*, pages 437–441. IEEE, 1997.

[40] Alfred Inselberg. The plane with parallel coordinates. *The visual computer*, 1(2):69–91, 1985.

[41] Wolfgang Jentner and Daniel A Keim. Visualization and visual analytic techniques for patterns. In *High-Utility Pattern Mining*, pages 303–337. Springer, 2019.

[42] Tabassum Kakar, Xiao Qin, Elke A Rundensteiner, Lane Harrison, Sanjay K Sahoo, and Suranjan De. Diva: Exploration and validation of hypothesized drug-drug interactions. In *Computer Graphics Forum*, volume 38, pages 95–106. Wiley Online Library, 2019.

[43] Mehmed Kantardzic. *Data mining: concepts, models, methods, and algorithms.* John Wiley & Sons, 2011.

[44] Tae Kyun Kim. T test as a parametric statistic. *Korean journal of anesthesiology*, 68(6):540, 2015.

[45] Bum Chul Kwon, Janu Verma, and Adam Perer. Peekquence: Visual analytics for event sequence data. In *ACM SIGKDD 2016 Workshop on Interactive Data Exploration and Analytics*, volume 1, 2016.

[46] Carson Kai-Sang Leung and Christopher L Carmichael. FpVAT: a visual analytic tool for supporting frequent pattern mining. *ACM SIGKDD Explorations Newsletter*, 11(2):39–48, 2010.

[47] Chenlu Li, Xiaoju Dong, Wei Liu, Shiying Sheng, and Aijuan Qian. SSRDVis: Interactive visualization for event sequences summarization and rare detection. *Journal of Visualization*, 23(1):171–184, 2020.

[48] Jiuyong Li, Thuc Duy Le, Lin Liu, Jixue Liu, Zhou Jin, Bingyu Sun, and Saisai Ma. From observational studies to causal rule mining. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(2):1–27, 2015.

[49] Guimei Liu, Andre Suchitra, Haojun Zhang, Mengling Feng, See-Kiong Ng, and Limsoon Wong. Assocexplorer: an association rule visualization system for exploratory data analysis. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1536–1539, 2012.

[50] Xiaotong Liu and Han-Wei Shen. Association analysis for visual exploration of multivariate scientific data sets. *IEEE transactions on visualization and computer graphics*, 22(1):955–964, 2015.

[51] Zhicheng Liu, Himel Dev, Mira Dontcheva, and Matthew Hoffman. Mining, pruning and visualizing frequent patterns for temporal event sequence analysis. In *Proceedings of the IEEE VIS 2016 Workshop on Temporal & Sequential Event Analysis*, pages 2–4, 2016.

[52] Zhicheng Liu, Yang Wang, Mira Dontcheva, Matthew Hoffman, Seth Walker, and Alan Wilson. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):321–330, 2016.

[53] José María Luna, Philippe Fournier-Viger, and Sebastián Ventura. Frequent itemset mining: A 25 years review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(6):e1329, 2019.

[54] Abhishek Mukherji, Xika Lin, Ermal Toto, Christopher R Botaish, Jason Whitehouse, Elke A Rundensteiner, and Matthew O Ward. Fire: a two-level interactive visualization for deep exploration of association rules. *International Journal of Data Science and Analytics*, 7(3):201–226, 2019.

[55] Adam Perer and Fei Wang. Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 153–162, 2014.

[56] Everett B Peterson, Kevin Neels, Nathan Barczi, and Thea Graham. The economic cost of airline flight delay. *Journal of Transport Economics and Policy (JTEP)*, 47(1):107–121, 2013.

[57] Felipe Pezoa, Juan L Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*, pages 263–273. International World Wide Web Conferences Steering Committee, 2016.

[58] Peter J Polack Jr, Shang-Tse Chen, Minsuk Kahng, Kaya De Barbaro, Rahul Basole, Moushumi Sharmin, and Duen Horng Chau. Chronodes: Interactive multifocus exploration of event sequences. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(1):1–21, 2018.

[59] Nornadiah Mohd Razali, Yap Bee Wah, et al. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*, 2(1):21–33, 2011.

[60] Cristóbal Romero, José María Luna, José Raúl Romero, and Sebastián Ventura. Rm-tool: A framework for discovering and evaluating association rules. *Advances in Engineering Software*, 42(8):566–576, 2011.

[61] Michael Schmidt. A review of aircraft turnaround operations and simulations. *Progress in Aerospace Sciences*, 92:25–38, 2017.

[62] Brian B Schultz. Levene's test for relative variation. *Systematic Zoology*, 34(4):449–456, 1985.

[63] Yoones A Sekhavat and Orland Hoeber. Visualizing association rules using linked matrix, graph, and detail views. *International Journal of Intelligence Science*, 3(1A):34–49, 2013.

[64] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pages 364–371. Elsevier, 2003.

[65] Aashara Shrestha, Dimitrios Zikos, and Leonidas Fegaras. An annotated association mining approach for extracting and visualizing interesting clinical events. *International Journal of Medical Informatics*, page 104366, 2020.

[66] Roberta Siciliano, Antonio D'Ambrosio, Massimo Aria, and Sonia Amodio. Analysis of web visit histories, part ii: Predicting navigation by nested stump regression trees. *Journal of Classification*, 34(3):473–493, 2017.

[67] Zheng Yang Sng and R John Hansman. A petri net framework for the representation and analysis of aircraft turnaround operations. Master's thesis, Massachusetts Inst. Technol., Cambridge, MA, USA, 2019.

[68] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.

[69] James J Thomas and Kristin A Cook. A visual analytics agenda. *IEEE computer graphics and applications*, 26(1):10–13, 2006.

[70] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[71] Mihaela Vranić, Damir Pintar, and Marko Banek. Towards better understanding of frequent itemset relationships through tree-like data structures. *Expert systems with applications*, 42(3):1717–1729, 2015.

[72] Katerina Vrotsou and Aida Nordman. A window-based approach for mining long duration event-sequences. 2020.

[73] Huy Quan Vu, Gang Li, Rob Law, and Yanchun Zhang. Travel diaries analysis by sequential rule mining. *Journal of travel research*, 57(3):399–413, 2018.

[74] Biying Wang, Tingting Zhang, Zheng Chang, Tapani Ristaniemi, and Guohua Liu. 3D matrix-based visualization system of association rules. In *2017 IEEE International Conference on Computer and Information Technology (CIT)*, pages 357–362. IEEE, 2017.

[75] Feng Wang, Wenwen Li, Sizhe Wang, and Chris R Johnson. Association rules-based multivariate analysis and visualization of spatiotemporal climate data. *ISPRS International Journal of Geo-Information*, 7(7):266, 2018.

[76] Jishang Wei, Zeqian Shen, Neel Sundaresan, and Kwan-Liu Ma. Visual cluster exploration of web clickstream data. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 3–12. IEEE, 2012.

[77] Krist Wongsuphasawat and David Gotz. Outflow: Visualizing patient flow by symptoms and outcome. In *IEEE VisWeek Workshop on Visual Analytics in Healthcare, Providence, Rhode Island, USA*, pages 25–28. American Medical Informatics Association, 2011.

[78] Dong Hoon Yang, Ji Hoon Kang, Young Bae Park, Young Jae Park, Hwan Sup Oh, and Seoung Bum Kim. Association rule mining and network analysis in oriental medicine. *PLOS one*, 8(3):e59241, 2013.

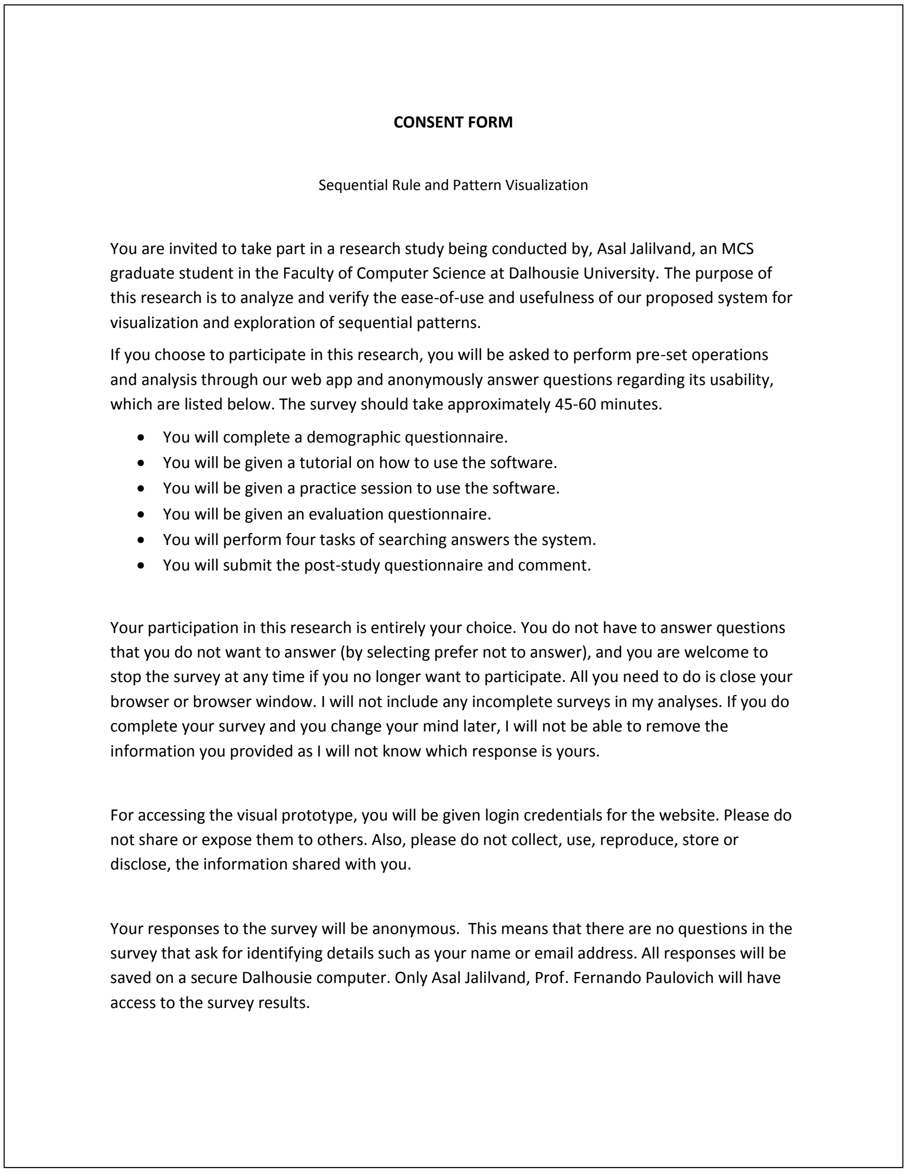
[79] Chong Zhang, Yang Chen, Jing Yang, and Zhengcong Yin. An association rule based approach to reducing visual clutter in parallel sets. *Visual Informatics*, 3(1):48–57, 2019.

[80] Huijun Zhang, Junjie Chen, Yan Qiang, Juanjuan Zhao, Jiangyang Xu, Xiaobo Fan, Yemin Yang, and Xiaolong Zhang. DART: a visual analytics system for understanding dynamic association rule mining. *The Visual Computer*, pages 1–17, 2020.

[81] Hanqing Zhao, Huijun Zhang, Yan Liu, Yongzhen Zhang, and Xiaolong Luke Zhang. Pattern discovery: a progressive visual analytic design to support categorical data analysis. *Journal of Visual Languages & Computing*, 43:42–49, 2017.

Appendix A

Consent Form



CONSENT FORM

Sequential Rule and Pattern Visualization

You are invited to take part in a research study being conducted by, Asal Jalilvand, an MCS graduate student in the Faculty of Computer Science at Dalhousie University. The purpose of this research is to analyze and verify the ease-of-use and usefulness of our proposed system for visualization and exploration of sequential patterns.

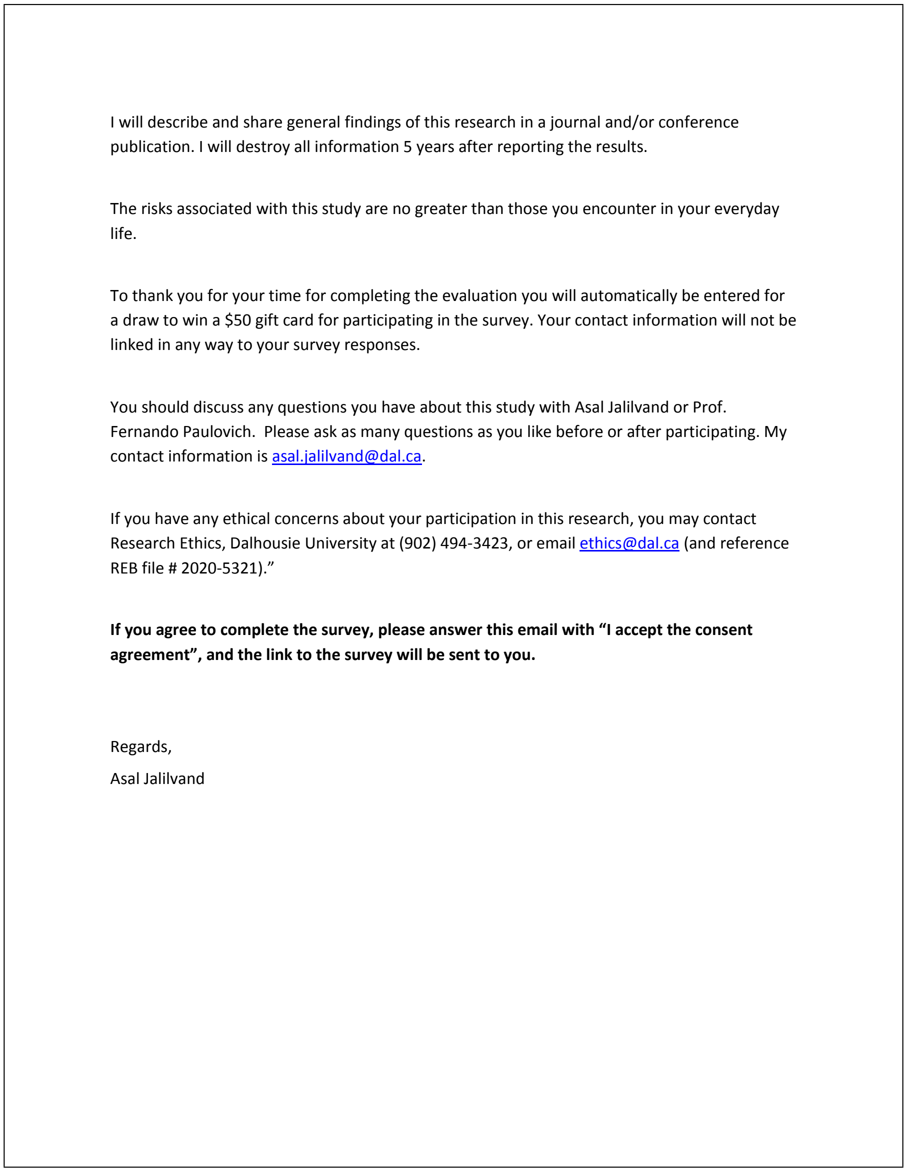
If you choose to participate in this research, you will be asked to perform pre-set operations and analysis through our web app and anonymously answer questions regarding its usability, which are listed below. The survey should take approximately 45-60 minutes.

- You will complete a demographic questionnaire.
- You will be given a tutorial on how to use the software.
- You will be given a practice session to use the software.
- You will be given an evaluation questionnaire.
- You will perform four tasks of searching answers the system.
- You will submit the post-study questionnaire and comment.

Your participation in this research is entirely your choice. You do not have to answer questions that you do not want to answer (by selecting prefer not to answer), and you are welcome to stop the survey at any time if you no longer want to participate. All you need to do is close your browser or browser window. I will not include any incomplete surveys in my analyses. If you do complete your survey and you change your mind later, I will not be able to remove the information you provided as I will not know which response is yours.

For accessing the visual prototype, you will be given login credentials for the website. Please do not share or expose them to others. Also, please do not collect, use, reproduce, store or disclose, the information shared with you.

Your responses to the survey will be anonymous. This means that there are no questions in the survey that ask for identifying details such as your name or email address. All responses will be saved on a secure Dalhousie computer. Only Asal Jalilvand, Prof. Fernando Paulovich will have access to the survey results.



I will describe and share general findings of this research in a journal and/or conference publication. I will destroy all information 5 years after reporting the results.

The risks associated with this study are no greater than those you encounter in your everyday life.

To thank you for your time for completing the evaluation you will automatically be entered for a draw to win a $50 gift card for participating in the survey. Your contact information will not be linked in any way to your survey responses.

You should discuss any questions you have about this study with Asal Jalilvand or Prof. Fernando Paulovich. Please ask as many questions as you like before or after participating. My contact information is asal.jalilvand@dal.ca.

If you have any ethical concerns about your participation in this research, you may contact Research Ethics, Dalhousie University at (902) 494-3423, or email ethics@dal.ca (and reference REB file # 2020-5321)."

If you agree to complete the survey, please answer this email with "I accept the consent agreement", and the link to the survey will be sent to you.

Regards,

Asal Jalilvand

Appendix B

SeRVis Video Demo

A video demo of the first use case in chapter 6 is uploaded as an electronic supplement on Dalspace. This video may also be found at Youtube `https://youtu.be/snxpXPj1Vmg`.