Bus Inf Syst Eng
https://doi.org/10.1007/s12599-023-00844-5

STATE OF THE ART

A Survey on Association Rule Mining for Enterprise Architecture Model Discovery

Carlos Pinheiro · Sergio Guerreiro · Henrique S. Mamede

Received: 4 May 2023 / Accepted: 5 October 2023

Abstract Association Rule Mining (ARM) is a field of data mining (DM) that attempts to identify correlations among database items. It has been applied in various domains to discover patterns, provide insight into different topics, and build understandable, descriptive, and predictive models. On the one hand, Enterprise Architecture (EA) is a coherent set of principles, methods, and models suitable for designing organizational structures. It uses viewpoints derived from EA models to express different concerns about a company and its IT landscape, such as organizational hierarchies, processes, services, applications, and data. EA mining is the use of DM techniques to obtain EA models. This paper presents a literature review to identify the newest and most cited ARM algorithms and techniques suitable for EA mining that focus on automating the creation of EA models from existent data in application systems and services. It systematically identifies and maps fourteen candidate algorithms into four categories useful for EA mining: (i) General Frequent Pattern Mining, (ii) High Utility Pattern Mining, (iii) Parallel Pattern Mining, and (iv) Distribute Pattern Mining. Based on that, it discusses some possibilities and presents an exemplification with a prototype hypothesizing an ARM application for EA mining.

Keywords Association rule mining · Data mining · Enterprise architecture mining · Enterprise architecture modelling · Artificial intelligence

1 Introduction

Association Rule Mining (ARM) is defined by Liu et al. (2021) as a data mining (DM) field that attempts to identify correlations among the items in a database. It is an essential DM technique applied in numerous domains to discover association patterns among data in huge databases (Sinaei and Fatemi 2018). The notion of ARM was first established by Agrawal et al. (1993), who proposed the Apriori algorithm to identify frequent patterns (FP) and the corresponding association rules. Since then, a wide variety of applications have been applying ARM to discover trends and patterns from vast sets of data in order to determine the associations, correlations, and frequent patterns among the items present in a database and link the presence of an item depending on the other items in a transaction (Menaga and Saravanan 2021).

In a different context, Enterprise Architecture (EA) includes the concerns, architecture principles, models, views and frameworks of an organization (Greefhorst and Proper 2011). EA models represent different viewpoints for managing different concerns of the company and its IT landscape, such as processes, services, and information

Accepted after two revisions by Hans-Georg Fill.

C. Pinheiro (✉)
Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal
e-mail: carlos.guti@gmail.com

C. Pinheiro · S. Guerreiro
INESC-ID, Rua Alves Redol 9, 1000-029 Lisbon, Portugal
e-mail: sergio.guerreiro@tecnico.ulisboa.pt

S. Guerreiro
Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal

H. S. Mamede
Department of Science and Technology, INESC TEC, Universidade Aberta, Lisbon, Portugal
e-mail: hsmamede@gmail.com

Published online: 21 December 2023

systems (The Open Group 2018). Perez-Castillo et al. (2019) describe EA mining as the usage of data mining techniques to obtain an up-to-date view of EA models. According to them, EA modelling is still done with a low degree of automation, reinforcing the importance and opportunities to develop EA mining techniques. Thus, this paper considers the plausibility of applying DM principles and techniques to the EA context.

Based on these contexts, the main objectives of this paper are to gather the applications of ARM techniques in the field of EA and gather the available methods to correlate data related to EA concerns using ARM to support enterprise architectural modelling. Hence, it performs a literature review to assess whether previous research exists in this domain. The review followed a well-known process described by Kitchenham (2004) to answer two research questions regarding how ARM techniques have been used to build enterprise architectural viewpoints and the latest published ARM techniques applicable to IT-related architecture mining.

This literature review is one of the foundation parts of a research initiative seeking to apply artificial intelligence techniques to automatically build EA models, such as high-level enterprise business processes and information system models (Pinheiro et al. 2021). Usually, events are logged in industrial applications without an explicit correlation, and it is time-consuming to analyze and correlate them manually. Associating different events with different data sets without an explicit link is related to the problem-solving space of ARM. Some successful examples of correlating different data sets apply neural networks to fuse and transform diverse data from different data sources into uniformized data (Cai et al. 2022; Noori et al. 2020), and the usage of user-defined rules to create a single semantically integrated event log into a data mining process (Modaresnezhad et al. 2019; Onan 2019). Hence, considering its viability, this paper seeks to identify the latest published and most cited ARM techniques and algorithms applicable to EA mining, spanning from 2018 to 2023.

Despite the great diversity of algorithms, this review mapped 14 candidate algorithms for four categories that solve different groups of problems, require different equipment, and represent different processing needs regarding the volume of data and processing time that new experiments in EA mining must take into account: (i) General Frequent Pattern Mining, (ii) High Utility Pattern Mining, (iii) Parallel Pattern Mining, and (iv) Distribute Pattern Mining. It, therefore, aims to help choose which ARM solution to apply.

The rest of this paper is presented in the following structure: Section 2 identifies the relevant background about ARM, DM, and EA to provide foundational knowledge and a justification for this review. Section 3 describes the research methodology. Section 4 presents the results report with a data synthesis and the result for each research question. It is followed by a discussion in Sect. 5 about the findings, implications, and insights for EA modelling. Finally, and to conclude, a summary of the results and future research are referred to in Sect. 6.

2 Background

This section introduces the general foundations and main concepts of DM, ARM and EA Mining and justifies the need for this review.

2.1 Motivation

EA management literature indicates that maintaining the EA model is still done through manual activities with a low degree of automation, and it represents one of the most significant challenges for EA management (Farwick et al. 2016; Perez-Castillo et al. 2019). Hence, to contribute to a solution on how to make the modelling of current EA more agile and automatized in order to better support architectural decisions, it is essential to automate the process of at least the AS-IS architecture model discovery. Thus, ARM techniques can play a crucial role in quickly providing new insights and bringing light to hidden relevant knowledge about the current state of EA. Therefore, the necessity to automate the discovery of EA models by collecting and aggregating data from various data sources to support the analysis of current EA models and enabling enterprise architects to make increasingly faster architectural insights and decisions motivated this literature review to explore the possibilities of applying ARM to the EA mining context.

2.2 Enterprise Architecture Mining

Enterprise Architecture models represent different viewpoints for managing different concerns of a company and its IT landscape, such as processes, services, and information systems (The Open Group 2018). These models cover concepts that reflect the business and IT perspective and require constant updates in response to the company's continuous transformations. As these models expand, it becomes harder to keep the relevant information in the architecture up-to-date (Farwick et al. 2016).

The term EA Mining is related to applying DM techniques to get an up-to-date view of EA models (Perez-Castillo et al. 2019). It is essential for optimizing the components of EA, planning changes, and ensuring alignment with strategic and business objectives that also help to reduce risks and costs for organizations. At the same time, the speed and volatility we have witnessed in business

changes have demanded an even faster response from the architecture. Hence, companies are continuously redefining their business goals, and it requires them to constantly review and adapt their processes as well as build new views that allow them to predict the impact of changes quickly.

2.3 DM and ARM

DM is an emerging field that has received increasing attention over the last two decades, with many studies seeking to apply it to a wide variety of applications with the aim of discovering trends and patterns from vast sets of data (Menaga and Saravanan 2021). To do that, it uses a variety of specialized algorithms that have been evolving to improve processing time, memory and storage space usage, accuracy improvement, and the utility of mined data (Datta and Mali 2021; Liu et al. 2021; Luna et al. 2019). DM facilitates the discovery of information related to associations, sequences, classification, clustering, and predictions (Laudon and Laudon 2021, p. 266; Liu et al. 2021). These systems perform high-level analysis for patterns or trends but can also break down the data to reveal more detail if necessary (Laudon and Laudon 2021, p. 267).

ARM is an unsupervised DM task that extracts interesting associations and frequent patterns from item sets in a database transaction or other data stores (Liu et al. 2021). ARM determines the associations, correlations, and frequent patterns from frequent items presented in a database, and it identifies when an item's presence depends on the other items in a transaction (Menaga and Saravanan 2021). Usually, traditional ARM approaches are based on support-confidence frameworks. The support (sup) measures the item's frequency and expresses the itemset's popularity. The confidence (conf) measures the probability of an item's occurrence and expresses the strength of the rules (Datta and Mali 2021).

ARM technique identifies an antecedent (or condition) and a consequent (or result) that have a conditional connection present in a transaction, such as defined below (Liu et al. 2021, p. 3):

- A = Antecedent, C = Consequent, T = Transaction
- $A \rightarrow C$, "If an event occurs, then an outcome event will happen."
- A and C are different sets, thus $A \cap C = \varnothing$
- $A \subset T$, then, it is expected that $C \subset T$, "Despite the condition A and the consequent C being different sets, both are contained in the same uniquely identified transaction T".

ARM techniques represent one of the central parts of Knowledge Discovery from Databases (KDD), described by Gullo (2015) as the process of identifying new, valid, potentially useful and understandable patterns in data. In the KDD context, the term "pattern" is a concept that refers to how a subset of the data is expressed in some language or model to represent the relation of its items. The KDD process is a sequence of steps summarized in Fig. 1.

2.3.1 Select Data

The first step in the KDD process is to select data. It is related to defining goals that drive the KDD, extracting data from different data sources to feed the ARM process, and selecting the data that will be worked with, usually creating a transactional event log. Thus, different data are integrated and merged into a unified transaction data log with data that supports the mining objectives.

2.3.2 Preparing Data and Transformation

Preparing Data and Transformation perform essential steps related to cleaning, reducing data dimensions, preparing data to be mined and ensuring the achievement of the goals defined in the data selection. It can be divided into two groups of tasks: prepare data to extract and remove noise, which removes incompatible details from the raw data and defines proper strategies for handling missing data fields. Data transformation converts data into suitable mining types, pursuing the reduction and projection of the data to derive a representation suitable for the specific goal. It is typically accomplished by involving transformation techniques or methods to find invariant data representations, such as discretization and fuzzification.

2.3.3 Frequent Pattern Mining

Frequent Pattern Mining (FPM) is the step that applies algorithms and techniques to discover patterns and hidden rules. In this stage, the trends of the data mining values are established by choosing a proper data-mining method and algorithms, including FIM (Frequent Item Mining), classification, and clustering, among others. FIM extracts sets of items that frequently appear in transactional data. Classification takes a collection of records as input, where each record is composed of a set of attributes, and one of the attributes denotes the class of the record. The goal is to find a model for the class attribute as a function of the values of the other attributes. This involves clustering, which aims to identify a finite set of groups of objects so that the objects within the same cluster are similar, whereas the objects belonging to different clusters are dissimilar.

2.3.4 Visualize and Interpret Patterns

In this step, users get insights and new knowledge from patterns discovered by analyzing their results. It is,

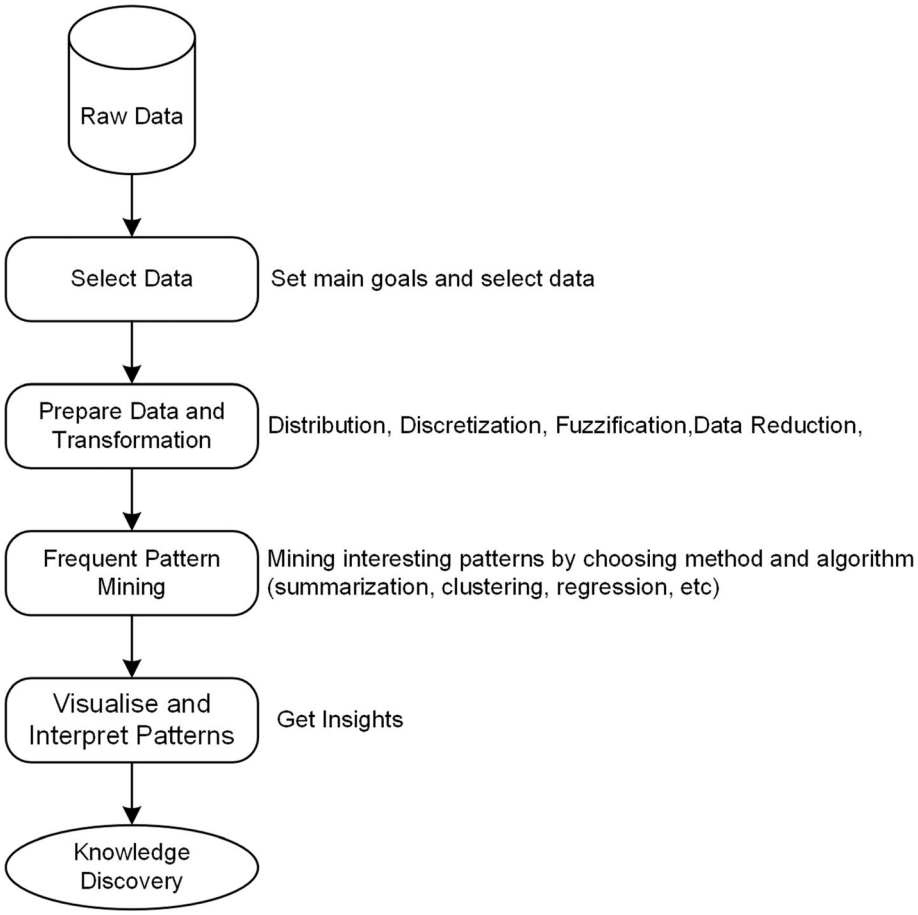


Fig. 1 The basic flow of the KDD process (Adapted from Gullo 2015)

therefore, essential to provide tools to aid users in the task of interpreting and evaluating the discovered patterns and consolidating knowledge.

This topic presented an outlook on the processes involved in KDD to clarify the position of this research. For that reason, the process was described. However, this paper focuses on the FPM step and identifies some possible applications for the identified methods in EA modelling.

3 Research Methodology

According to Kiteley and Stogdon (2014), a literature review may be conducted as a research methodology focusing on gathering what is currently known about a specific subject or problem. Therefore, a literature review can be used to consolidate understanding, gather findings, or highlight the most convincing proposals in the published literature thus far. In this context, this review followed the process proposed by Kitchenham (2004), composed of three phases, as illustrated in Fig. 2.

3.1 Planning

In the planning phase, this paper sets two main objectives based on the context, problem, and motivations previously presented as the drivers for this research. Firstly, gather the applications of ARM techniques in the field of EA, and secondly, gather the available methods in the literature to correlate or associate data related to enterprise architecture concerns using ARM to support enterprise architectural or process modelling. Therefore, it seeks to answer the questions below to review the state of the art for these objectives.

RQ 1—How are ARM techniques used for building enterprise architectural viewpoints?

RQ 2—What are the latest ARM techniques applicable to IT-related architecture mining?

RQ 1 focuses on gathering the usage of ARM, specifically in EA. RQ 2 seeks a broader scope to identify techniques applied to other IT-related architecture that may not have been applied to EA yet. Thus, the review was split into two search lines for each RQ.

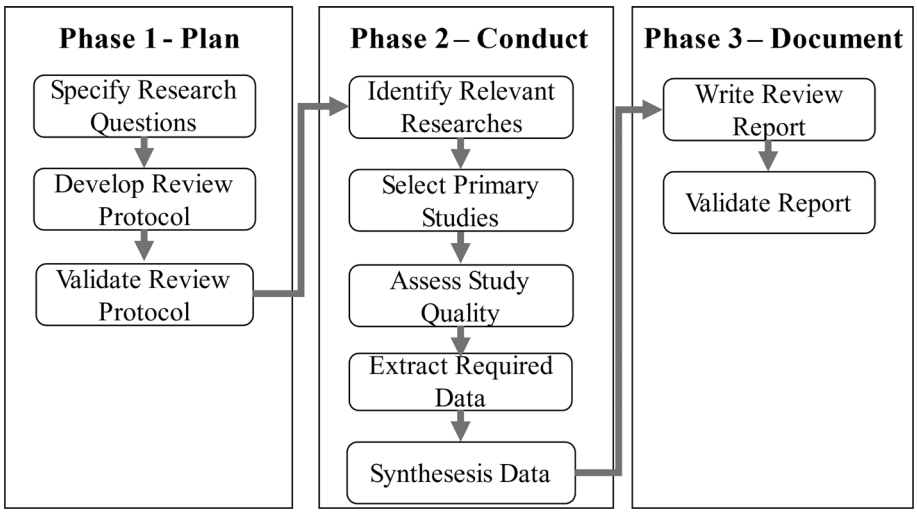


Fig. 2 Kitchenham's systematic review process

3.1.1 Plan for RQ 1 (How are ARM Techniques Used for Building Enterprise Architectural Viewpoints?)

3.1.1.1 Search Strategy for RQ 1 This first search focuses on the enterprise architect's population to identify the usage of ARM techniques to build architectural models. For that reason, the following search string was elaborated.

> ("Association Rule* Mining" OR ARM OR "Knowledge Discovery from Database") AND ("Enterprise Architecture")

3.1.1.2 Study Selection Plan for RQ 1 Table 1 describes the inclusion and exclusion criteria for selecting papers regarding RQ 1.

For this selection, the criteria for not complying with the objective sought to exclude works unrelated to the application of ARM to support architectural modelling. For instance, works that explain an ARM architecture in a medical science application.

3.1.2 Plan for RQ 2 (What are the Latest ARM Techniques Applicable to IT-Related Architecture Mining?)

3.1.2.1 Search Strategy for RQ 2 This second search focuses on the enterprise architect's population. However, unlike the first search, this one aims to find applications of ARM to build any level of IT-related architecture, correlating data structures, such as software architecture, infrastructure IT architecture, application architecture and others.

It intends to identify the best candidate approaches to correlate data mined in system logs to build architectural models. Hence, the search string that follows was elaborated with that in mind.

> ("Association Rule Mining" OR "Knowledge Discovery from Database") AND (Algorithm OR Technique) AND Architecture

3.1.2.2 Study Selection Plan for RQ 2 In RQ 2, the inclusion and exclusion criteria have also been defined, as described in Table 2.

Table 1 RQ 1 inclusion and exclusion criteria

Type	Criteria
Inclusion	Written in English
Exclusion	Not accessible, not a paper, i.e., books, thesis, patents,
	Title and Keyword out of the research objectives,
	Not peer-reviewed,
	Abstract, introduction and conclusion out of the research objectives,
	Predatory suspect by Beall's List or Stop Predatory Journal,
	Content out of the objective

Table 2 RQ 2 inclusion and exclusion criteria

Type	Criteria
Inclusion	Published after 2018 with more than one citation,
	Published after 2021 independently of the citations,
	Written in English
Exclusion	Source incomplete, not found, not accessible, not a paper, i.e., books, thesis, and patents,
	Title and Keyword out of the research objectives,
	Not peer-reviewed,
	Abstract, introduction and conclusion out of the research objectives,
	Predatory suspect by Beall's List or Stop Predatory Journal,
	Content out of the objective

For both research questions, the two search strings were planned to be applied to all fields on the Web of Science, IEEE Explorer, ACM Digital Library, and Scopus.

In this case, the criteria for not complying with the objective excluded works that do not present an ARM technique applicable to any IT-related architecture mining.

3.2 Conduction for RQ 1 (How are ARM Techniques Used for Building Enterprise Architectural Viewpoints?)

The search string was applied to all fields on the Web of Science, IEEE Explorer, ACM Digital Library, and Scopus between 16 February 2023 and 21 February 2023, resulting in a total of 145 papers.

3.2.1 Study Selection

The selection process, illustrated in Fig. 3, resulted in 11 papers selected from a total of 145 papers. No paper was excluded by not being published through a peer-reviewed process, and none was reported as predatory.

As 11 papers are a small number of works to read, no other filter based on quality was applied. It is remarkable that no single paper describes any ARM uses for EA models, except those that only cite the possible application of data mining to some EA contexts.

3.2.2 Data Extraction

Figure 4 depicts the extraction of papers by year. Despite the absence of papers using ARM explicitly and expressly, it demonstrates that the interest in EA mining has risen over the last 5 years. While, Fig. 5 shows that most papers are available on Scopus, followed by ACM.

3.3 Conduction for RQ 2 (What are the Latest ARM Techniques Applicable to IT-Related Architecture Mining?)

3.3.1 Study Selection

For this case, it was also considered out of scope to capture uses of ARM in IT, but with the main indications from disciplines such as healthcare, image processing, individual

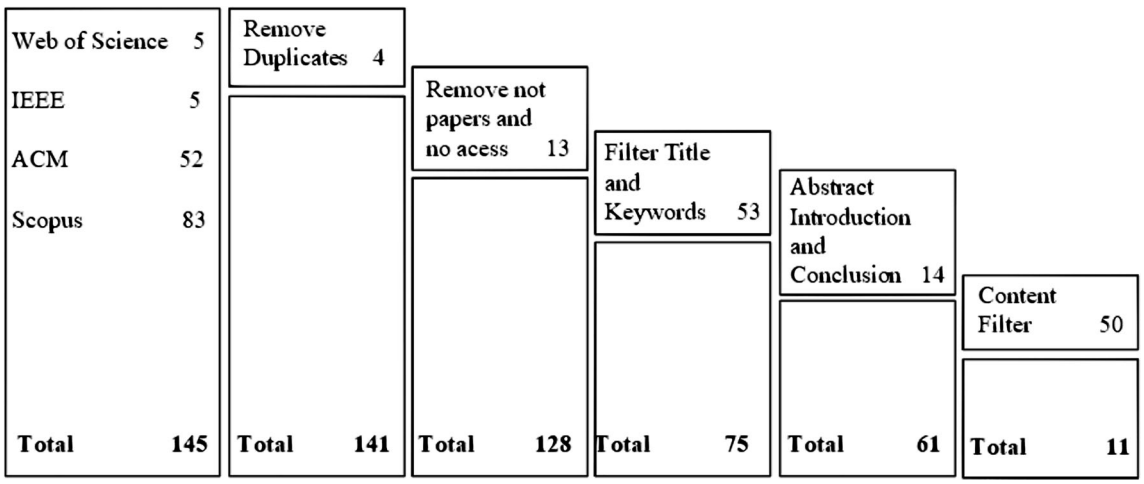


Fig. 3 RQ 1 study selection

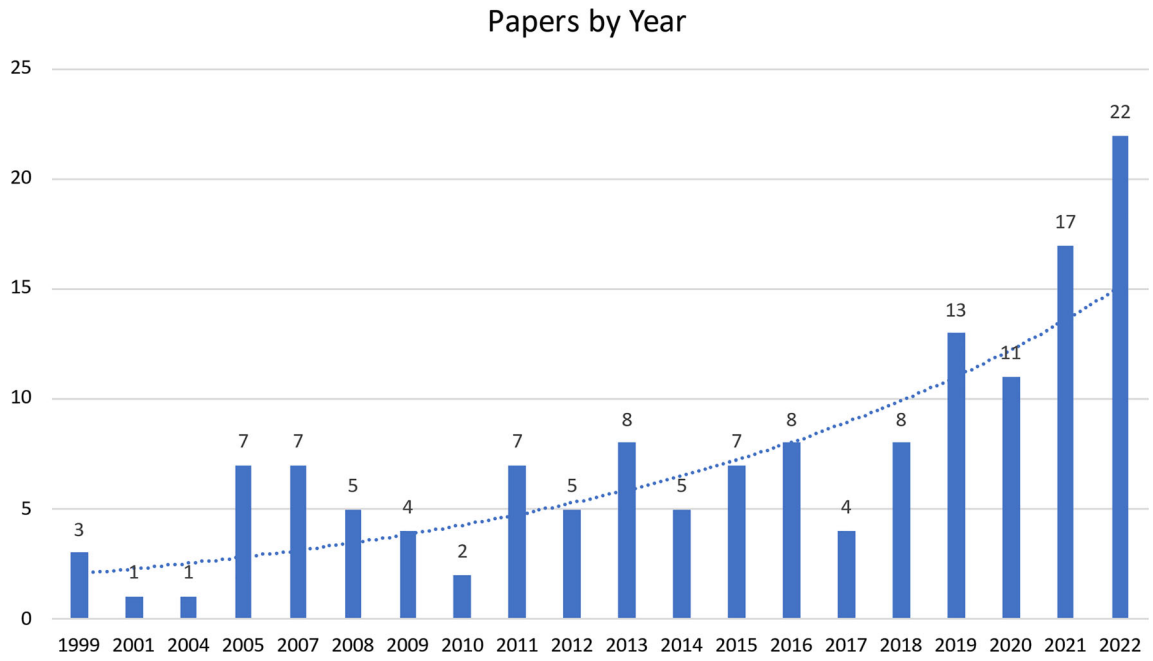


Fig. 4 EA mining by year

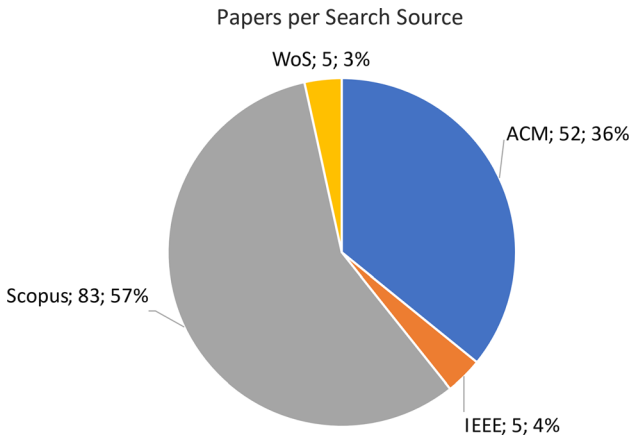


Fig. 5 EA mining per source

recommendation, Geographic Information Systems (GIS) and other domain-specific applications unrelated to EA or IT architecture. This also included many granular topics, such as IoT-related TELECOM Infrastructure and error detection in software source code. The selection process is depicted in Fig. 6, which resulted in 25 papers selected for analysis.

3.3.1.1 Quality Assessment After applying the inclusion and exclusion criteria, a quality analysis was also carried out to assess the quality and adherence of papers selected for the context being investigated, basically answering the seven quality questions listed here, which are related to the objective of this research:

1. Was the designed study aimed at any kind of architecture modelling automation?
2. Was the research method clearly described?
3. Was the method of association rule mining detailed and well explained?
4. Was it validated with a solid and clear evaluation method?
5. Were outcomes assessed using objective criteria?
6. Was there any comparison to other reference algorithms?
7. Are all study questions answered?

Each question had three possible answers: “Yes”, “Partially”, and “No”, with values corresponding to 1.0, 0.5, and 0.0, respectively. For the first question, the analysis considered if the paper designated or pointed out some aspects related to architecture modelling. In the second one, the focus was to verify if the research method was clearly described (“Yes”), if not described but is somehow perceivable (“Partially”), or if it was not identified (“No”). The third question targets to ensure that it described and exemplified the mining method (“Yes”), if it was described at a high level without presenting its implementation (“Partially”), or if it looked for other cases (“No”). The fourth question assessed if the work presented an empirical and repeatable validation (“Yes”), if it presented an experiment but with some identified weaknesses, such as not presenting the data set used by the experiment (”Partially“), or if validation was not present (“No”). The fifth question aimed to analyze if the outcomes were objectively evaluated based on experiment results (“Yes”), if there

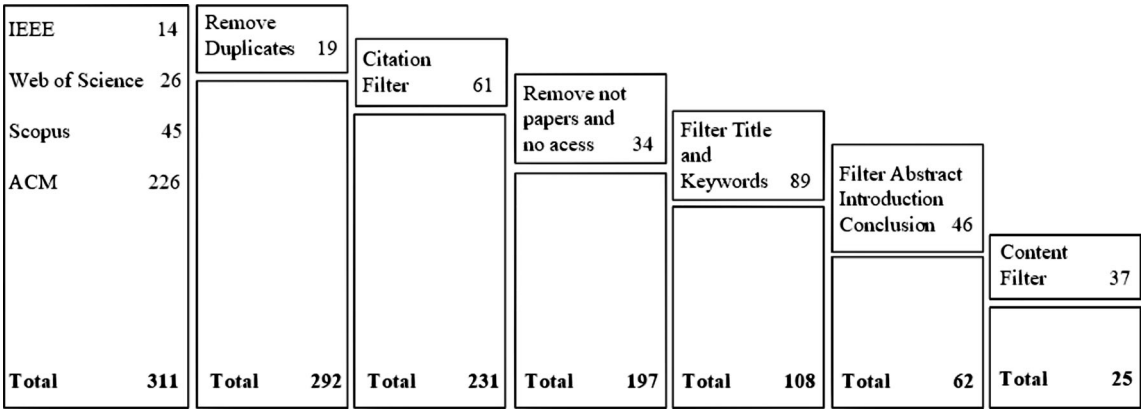


Fig. 6 RQ 2 study selection

was some or partial outcome evaluation ("Partially"), and whether the outcome evaluation was not present ("No"). For the sixth question, we confirmed if the method was compared with others ("Yes") or not ("No"). Finally, the seventh question checked if the research questions were clearly defined and answered ("Yes") if they were not directly and easily identified but it was possible to perceive the problem and whether or not it was answered ("Partially"), or other cases ("No").

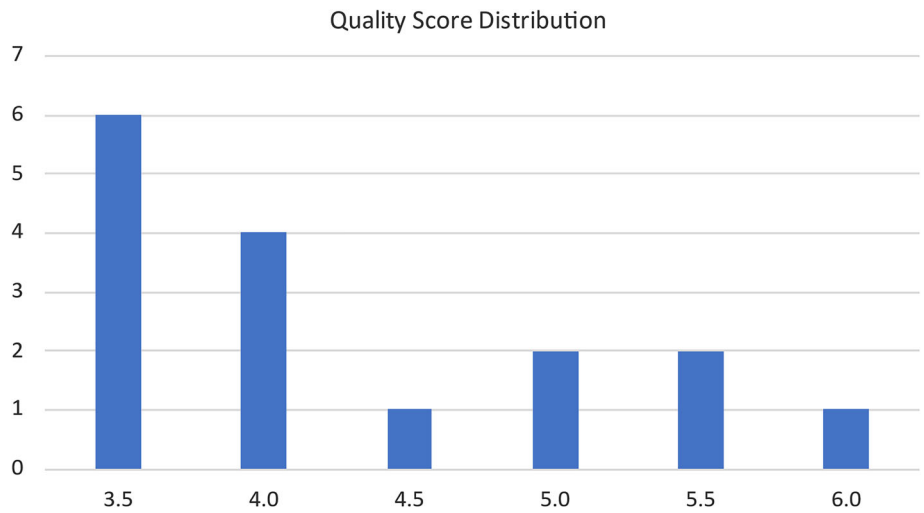
The quality analysis had a maximum of seven points, and it was decided not to incorporate works that had less than three points mainly because they probably had a minor contribution. The graph plotted in Fig. 7 shows the distribution of the quality score of the selected papers.

3.3.2 Data Extraction

Due to the large number of available algorithms, this research did not intend to cover all algorithms, both known and older ones. Presumably, previous advances had already been incorporated into new versions of the core algorithms. Therefore, this research focused on the latest algorithms reported in the literature since 2018 or referenced by works during this period. Despite this, algorithms with the best performance in the experiments are referenced, even when published before 2017. Figure 8 shows the distribution of selected papers by year.

Figure 9 depicts the quantity and percentage of papers selected by the search database. Notably, the ACM Digital Library was responsible for half of the papers in this search. At the end of the process, the 15 papers listed in Table 3 served as primary references for the detailed analysis of the methods and algorithms. Some other algorithms were also added to these references and used in other sections ahead for algorithm comparative analysis. Regarding the location where the works have been published, they are quite dispersed in this list of papers. However, two journals stand out with two publications each: the *ACM Transactions on Knowledge Discovery from Data* and the *Proceedings of the ACM on Measurement and Analysis of Computing Systems*.

Fig. 7 Quality score distribution of selected papers



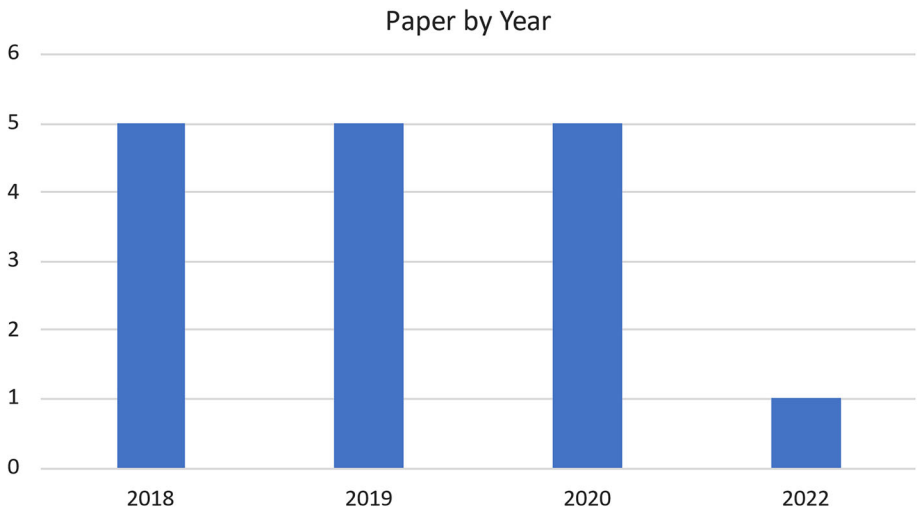


Fig. 8 Distribution of selected papers by year

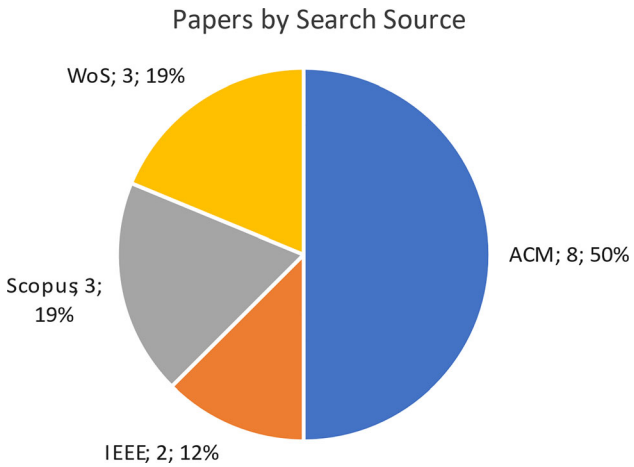


Fig. 9 RQ 2 paper selected by source

4 Results Report

This section summarizes the main works that contributed to this review. It represents the knowledge base available in the analyzed literature on ARM and architecture modelling automation that may help in EA AS-IS modelling automation.

4.1 Data Synthesis

Before going into the results of each research question, this subsection presents a synthesis of the selected works.

Many algorithms have been proposed for ARM. This research identifies the latest techniques applied to four distinct categories: general local sequential mining, high utility mining, parallel mining, and distributed mining.

Firstly, for RQ 1, the objective was to identify the use of ARM specifically in EA Mining. However, no paper demonstrated a direct application of ARM to EA. Despite that, we present and discuss some works highlighting insights into EA mining.

Secondly, for RQ 2, the goal was to identify the use of ARM in any IT-related architecture. In this way, Table 3 provides an overlook of the contribution found in relation to the perspective of RQ 2, indicating the location where these contributions were published.

4.2 Results for RQ 1 (How are ARM Techniques Used for Building Enterprise Architectural Viewpoints?)

This topic highlights some studies aligned with data mining for enterprise architectural model automation, as reported in the literature.

Neaga and Harding (2005) suggest that extending the existing enterprise modelling and integrating architectures by incorporating KDD and DM systems could significantly improve the decision-making process and business performance. They developed a conceptual design and development of an enterprise modelling and integration framework using knowledge discovery and data mining techniques. They suggested an approach to utilizing existing references for EA and modelling frameworks by introducing new enterprise views, such as mining and knowledge views. However, they did not develop the ARM process for EA. They only recommended the use of knowledge discovery and data mining systems libraries, such as PolyAnalyst^{TM}, Clementine, Weka, and ArMiner.

Gustavsson and Planstedt (2005) developed the fractal information fusion model (FIF), which is a model for the simulation of multiple hypotheses and intentions of different agents for military purposes, such as different behaviors of opponents and other agents. It aimed to provide an agent architecture that aligns with the global initiatives in an enterprise architecture initiative for the Swedish armed forces. They collect data and fuse them to

Table 3 Root contribution from the literature of RQ 2

Title	Author(s)	Journal/Conference
Spatio-Temporal Frequent Itemset Mining on Web Data	Aggarwal and Toshniwal (2018)	IEEE International Conference on Data Mining Workshops (ICDM Workshops)
Large-Scale Frequent Episode Mining from Complex Event Sequences with Hierarchies	Ao et al. (2019)	Proceedings of the ACM on Measurement and Analysis of Computing Systems
An effective Map-Reduce-based association rule mining method	Barkhordari and Niamanesh (2018)	Journal of Big Data
Splicing Community Patterns and Smells: A Preliminary Study	De Stefano et al. (2020)	International Conference on Software Engineering (ICSE)
Run-time mapping algorithm for dynamic workloads using association rule mining	Sinaei and Fatemi (2018)	Journal of Systems Architecture
Bacterial Colony Algorithms for Association Rule Mining in Static and Stream Data	da Cunha et al. (2018)	Mathematical Problems in Engineering
A Survey of Parallel Sequential Pattern Mining	Gan et al. (2019)	ACM Transactions on Knowledge Discovery from Data
Automatic Detection of Latent Software Component Relationships from Online Q&A Sites	Karthik and Medvidovic (2019)	International Workshop on Realising Artificial Intelligence Synergies in Software Engineering
Fast Dimensional Analysis for Root Cause Investigation in a Large-Scale Service Environment	Lin et al. (2020)	Proceedings of the ACM on Measurement and Analysis of Computing Systems
Apriori Versions Based on MapReduce for Mining Frequent Patterns on Big Data	Luna et al. (2019)	Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery
NOV-CFI: A Novel Algorithm for Closed Frequent Itemsets Mining in Transactional Databases	Phan (2018)	International Conference Proceeding Series (ICPS)
Mining Local Process Models with Constraints Efficiently: Applications to the Analysis of Smart Home Data	Tax et al. (2018)	International Conference on Intelligent Environments (IE)
High-Utility Itemset Mining with Effective Pruning Strategies	Wu et al. (2019)	ACM Transactions on Knowledge Discovery from Data
An Evolutive Frequent Pattern Tree-based Incremental Knowledge Discovery Algorithm	Liu et al. (2022)	ACM Transactions on Management Information Systems
Multitask-based association rule mining	Yildirim Taşer et al. (2020)	Turkish Journal of Electrical Engineering and Computer Sciences

predict results based on the integration of information from different sources about the behavior of a particular system to support decisions relating to this system. In their approach, a data sensor is constantly fed to the database using data-mining techniques, among other methods. However, the use of data mining techniques is not detailed.

Perez-Castillo et al. (2019) observed that enterprise application logs might be a powerful source of knowledge and, associated with data mining techniques, may raise the level of automation in Enterprise Architecture Modelling. Thus, they developed a reverse engineering approach based on a set of predefined architectural relationships to draw a dynamic model in ArchiMate (The Open Group 2019). Despite indicating the use of data mining techniques to extract and associate data based on textual terms and grammar rules, the authors focused more on explaining the ArchiMate model generation and did not detail the ARM algorithm used in any of the two subsequent works (Pérez-Castillo et al. 2020; 2021). However, their work solved part of the job in transforming the data resulting from the ARM process to a standardized EA model graphic visualization in ArchiMate. At the same time, it opens up opportunities to complement their approach through a deeper application of ARM techniques.

Despite these studies that form the background of this literature review, the point is that the usage of ARM in the field of EA modelling is still open, and few researchers have explored it. The literature process has not captured works that apply ARM to EA. Although some of them cite DM techniques, these works have not explored the full potential of ARM yet.

4.3 Results for RQ 2 (What are the Latest ARM Techniques Applicable to IT-Related Architecture Mining?)

Due to the large number of algorithms, it is important to define a criterion to help select algorithms related to

different challenges that impact EA mining efforts. The abstraction level of EA models must be filled by combining different ARM solutions. For instance, mining knowledge from other architectural models will not demand a high volume, but accuracy may be essential. However, mining data from application system logs will probably demand a more performant approach or even oblige the use of a distributed and more expensive solution due to the volume of data.

Thus, to obtain the latest advances in ARM techniques that can be applied to EA mining, the algorithms and methods identified were initially grouped into some ARM approaches that address different application opportunities as described and analyzed ahead.

In this context, Fig. 10 summarizes the timeline and comparisons observed in the literature gathered in this paper, which emphasizes the most cited and the latest advanced algorithm, and considers the experiments reported by comparative studies. It helps to quickly identify and select the candidate algorithms for future research developments, or at least the newer and most cited candidates applicable to EA mining purposes.

Most algorithms are presented as an evolution of the Apriori algorithm, developed by Agrawal et al. (1993), which continues to be one of the most cited algorithms (Gan et al. 2019; Luna et al. 2019). On the other hand, researchers were aware of the difficulty and limitations of local sequential mining and the importance of proposing not only efficient algorithmic solutions but also novel approaches (parallel and distributed computing) to handle such a problem (Luna et al. 2019).

Based on the seminal work by Agrawal et al. (1993), the contributions in terms of algorithms were classified into four different categories. These require different kinds of requirements and logic, and represent different needs of processing regarding the volume of data and processing

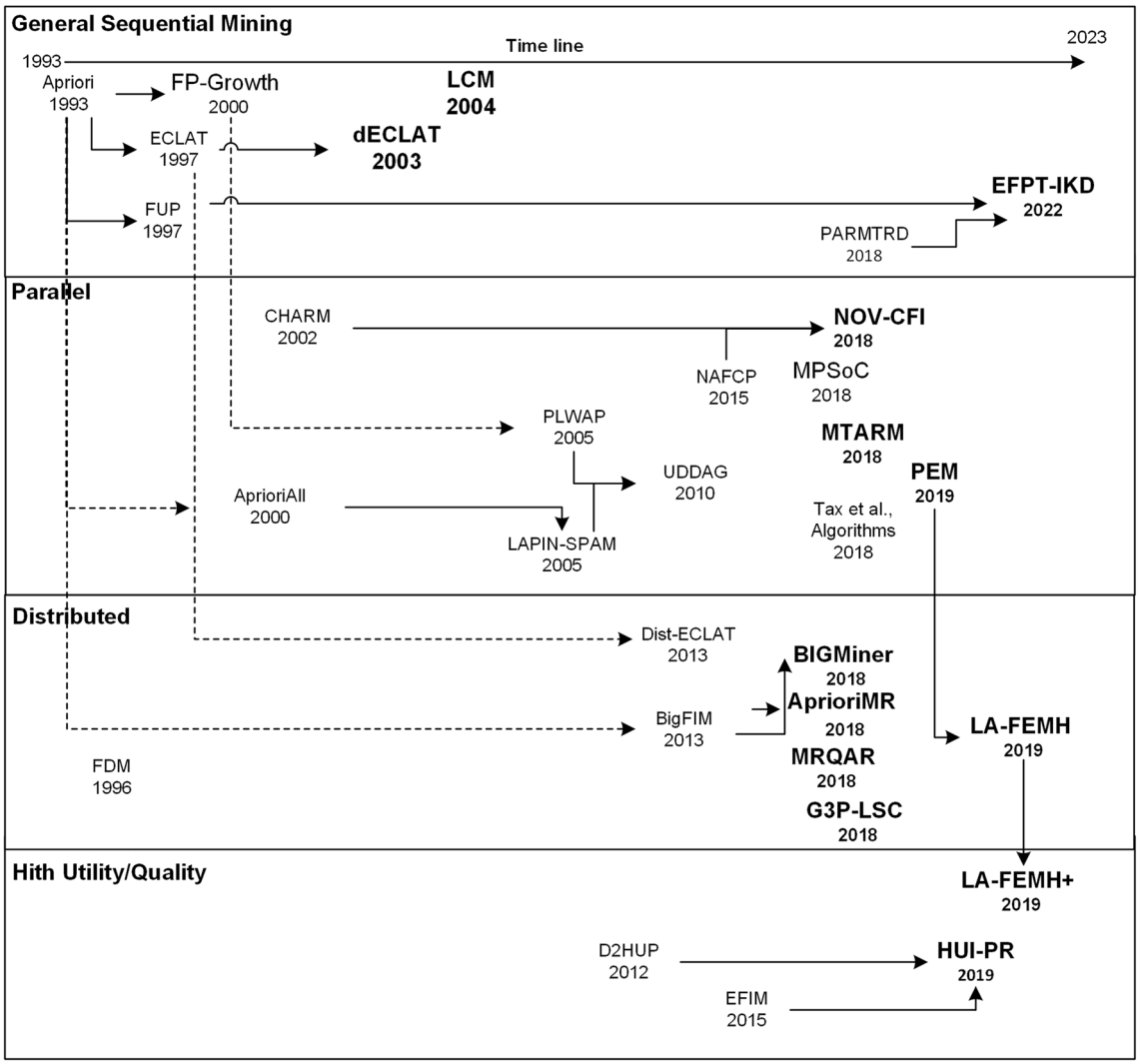


Fig. 10 ARM timeline evolution

time, which must be taken into account for new experiments in EA mining.

- General Frequent Pattern Mining
- High Utility Pattern Mining
- Parallel and Multi-thread Mining
- Distributed Mining

FIM is the root of the pattern-mining field, which encompasses multiple tasks that aim to extract sets of items that frequently (or infrequently) appear in data on multiple forms and for various purposes. Sometimes, it has been used interchangeably with the term ARM due to the objective of obtaining a frequent pattern (Luna et al. 2019). General Frequent Pattern Mining congregates ARM algorithms based on serial or sequential mining. Sequential, in this case, means algorithms that mine based on a linear sequence logic, in contrast with algorithms that aim to mine temporal or hierarchy sequences. These execute ARM in its logic, segmenting data and parallelly processing it, and in a more advanced approach, are found in a distributed environment. On the other hand, high-utility mining tries to obtain rules that maximize utility and generate fewer rules for all cases.

Indeed, at first glance, it is essential to consider the novelty of the algorithm and that the newer ones would bring improvements over the older ones. However, it is also essential to observe the relevance of the proposals, considering their adoption in the research field. In this sense, those that are most used are identified by the number of references (citations). Although the older ones tend to be cited more often, they still reinforce the usefulness and importance of the algorithm.

4.3.1 General Frequent Pattern

FIM is an essential task within data analysis since it is responsible for extracting frequently occurring events, patterns, or items in data. Insights from such pattern analysis offer important benefits in decision-making processes (Luna et al. 2019).

Apriori was the first algorithm, according to Luna et al. (2019). Many algorithms have been proposed for improvements based on the fact that the Apriori continues to be used as a baseline for a considerable number of studies. One of the most significant improvements of Apriori was FP-Growth (Han et al. 2000), which uses a pattern tree structure for storing and compressing information about patterns. Another key algorithm is ECLAT (Equivalent CLAss Transformation), which associates each pattern to a list of transactions that occur. Sometime later, dECLAT (diffsets with ECLAT) (Zaki and Gouda 2003) outperformed the running time of ECLAT and FP-Growth on dense datasets. In 2004, the algorithm named LCM (Linear time Closed itemset Miner) won the FIMI'04[1] competition on Frequent Itemset Mining Implementations, in which plenty of implementations and solutions with remarkably high performance for the time were shown.

Liu et al. (2022) proposed a new algorithm named Evolutive Frequent Pattern Tree-based Incremental Knowledge Discovery (EFPT-IKD). It is based on double-evolving frequent pattern trees that can trace the dynamically evolving data by an incremental sliding window. One tree records frequent patterns from the historical data, and the other records incremental frequent items. The structures of the double frequent pattern trees and their relationships are updated periodically according to the emerging data and a sliding window. According to the author's experiment, this incremental knowledge discovery algorithm surpassed FUP (Fast Update) (Cheung et al. 1996a) and PARMTRD (Parallel Association Rules Based Multiple-Topic Relationships Detection) (Liu et al. 2018).

Based on this analysis, the strongest candidates for general frequent pattern mining are dECLAT and LCM once they present clear improvements over Apriori, FP-Growth and ECLAT. However, it is important to note that these three algorithms remain relevant since they are frequently reported and serve as bases for new algorithm proposals. Finally, EFPT-IKD is the latest algorithm found so far. It probably performs better despite not being directly compared with dECLAT and the other algorithms cited in this review.

There is a subgroup in ARM where the individual events could be arranged in predefined hierarchies or temporal sequences that demonstrate different properties for different applications. In particular, hierarchy mining, such as FEM (Frequent Episode Mining), reported by Ao et al. (2019), allows mining the events in episodes that belong to different levels of the event hierarchy, containing abstractive concepts that are not clear in the original input data. On the other hand, temporal sequences mining is related to techniques that use constructs from the field of business process modelling to represent frequent patterns that go beyond sequential patterns and can express rich ordering relations that include concurrent execution, choices, and repetition (Tax et al. 2018). It usually includes features related to the timestamp or location, and in some applications, it is represented using graphs instead of a flat representation (Luna et al. 2019). LA-FEMH (LArge-scale Frequent Episode Mining with Hierarchies), presented by Ao et al. (2019), aims to mine events in a hierarchy-aware partition strategy, which divides the input sequence to produce a balanced workload partition but is also a scalable distributed mining framework.

[1] https://ceur-ws.org/Vol-126/.

Table 4 General sequential ARM

Algorithm	Citations	Reference
FP-Growth	10,104	Han et al. (2000)
dECLAT	900	Zaki and Gouda (2003)
LCM	557	Uno et al.(2004)
EFPT-IKD	5	Liu et al. (2022)

This view is expressed in Table 4, which identifies the evolution and latest number of citations detected for each category of algorithm.

4.3.2 High Utility Improvement in Mining

High utility itemset mining is a prevalent data mining problem that considers utility factors. The designed pruning strategies help reduce the visitation of unnecessary nodes in the search space, which reduces the time required by the algorithm (Wu et al. 2019).

According to Wu et al. (2019), High-Utility Itemset (HUI) mining is a data mining problem that considers utility factors, such as quantity and the unit profit of items, aside from the frequency measure from transactional mining. In this field, they designed HUI-PR, a HUI algorithm that applies pruning strategies to reduce the visitation of unnecessary nodes in the search space. In addition to the utility mining strategy, their approach reduces the time required for mining. The memory usage of the algorithm also outperforms the state-of-the-art approach D2HUP (Liu et al. 2012) and EFIM algorithms (Zida et al. 2015) using six real datasets (chess, mushroom, connect, accidents, and retail) that showed the effectiveness of HUI-PR.

LA-FEMH + (Ao et al. 2019) is an extension of the previous algorithm called LA-FEMH, which focused on the use of the concept of maximal and closed episodes in the context of event hierarchies to support other episode mining tasks such as maximal and closed episodes in the context of event hierarchies. They demonstrated the effectiveness of the approach through the implementation of MapReduce on Apache Spark and performed experimental studies on both synthetic and real-world datasets, including financial sequences and natural language text.

In addition to the traditional frequent item mining for contexts where it is desired to avoid getting too many useless rules, LA-FEMH + and HUI-PR are the newest and most promissory candidates identified in this search, as shown in Table 5.

Table 5 High utility ARM

Algorithm	Citations	Reference
LA-FEMH	29	Ao et al. (2019)
HUI-PR	77	Wu et al. (2019)

4.3.3 Parallel ARM

This group presents the algorithms that generally use parallel sequence mining logic based on multi-thread architectures with shared memory, or in other words, a single computer with multiple processors, usually by extending the existing serial algorithms (Luna et al. 2019).

Phan (2018) advocates that closed frequent itemset mining is one of the fundamental tasks in ARM. However, it is time-consuming, and most algorithms find closed frequent items set on search space items that are not reused for mining the next time. To solve this issue, he proposed the NOV-CFI algorithms, a novel approach to quickly detect closed frequent item sets from transactional databases using an array of co-occurrences and occurrences of kernel items in at least one transaction. Besides its ability to be reused, it is also easily expanded in distributed systems. His experimental results show that the algorithms are better than NAFCP (Le and Vo 2015) and CHAM (Zaki and Hsiao 2002).

According to Ao et al. (2019), Frequent Episode Mining (FEM) aims to mine frequent sub-sequences from a single long event sequence and is one of the essential building blocks for the sequence mining research field. However, episode frequencies may fail to hold the anti-monotonicity property. For instance, all occurrence, minimal occurrence, and head frequency may lead to the frequency of a sub-episode inferior to its super-episode. Thus, they developed PEM (Peak Episode Miner), a specialized local miner that performs efficient specialized episode mining in local processes with the help of the proposed tree-like structure and concise scanning.

Yildirim Taşer et al. (2020) presented MTARM (Multitask Association Rule Miner). Instead of discovering rules from single tasks, it focuses on discovering frequent association rules by responding to different tasks and applying an algorithm that considers all tasks collectively along with the relation between them. The underlying assumption is that the rules of all tasks, or at least a subset of them, are familiar to one mutual rule set with a slight difference, and a rule may not be frequent in the entire dataset but be frequent in a group of specific tasks. Their experiment did not compare to other algorithms, but it had three different versions, including one of the well-known

reference algorithms. The Eclat version outperformed the one based on FP-Growth, which performed better than the Apriori version. Note that this result should be expected due to the evolution line of these algorithms, where they historically evolved precisely in this sequence, as presented by Luna et al. (2019) and Wu et al. (2019). Despite that, it is one of the latest algorithms identified in this review and seems to present a good solution for mining frequent rules related to multiple tasks. Table 6 summarizes the algorithms for parallel ARM.

4.3.4 Distributed Algorithms

This group presents algorithms based on distributed architectures. It is distinct from parallel architecture because, in distributed computing, each processor shares nothing and has its own private main memory and storage.

In their survey, Luna et al. (2019) designates AprioriMR (Luna et al. 2018), BIGMiner (Chon and Kim 2018), MRQAR (Martín et al. 2018) and G3P-LSC (Padillo et al. 2018) as the latest published approaches thus far. AprioriMR is a series of algorithms based on MapReduce and Hadoop that improves the performance of sequential mining in the distributed big data environment. It outperformed distEclat according to the AprioriMR experiments. BIGMiner is a fast and scalable MapReduce-based frequent itemset mining method that generates equal-sized sub-databases called transaction chunks and performs support counting only based on transaction chunks and bitwise operations without generating and shuffling intermediate data. MRQAR is a framework for sequential quantitative ARM in Big Data based on the MapReduce on Apache Spark.

Ao et al. (2019) propose LA-FEMH (LArge-scale Frequent Episode Mining with Hierarchies), a scalable distributed framework for frequent episode mining from complex sequences with event hierarchies. They implement the proposed framework on Apache Spark.

According to some authors, distributed systems allow quasi-linear scalability; thus, such approaches are becoming increasingly common. It is also notable that most of the existing proposals based on distributed computing considered the MapReduce framework (Luna et al. 2019). This group of algorithms should be used when scalability is a critical factor. In this sense, LA-FEMH is the latest algorithm identified, followed by AprioriMR, BIGMiner, MRQAR and G3P-LSC, all of which were published in 2018. For Distributed ARM, the algorithms are summarized in Table 7.

This section demonstrated some classes of algorithms to apply to diverse needs, mainly if sequential local mining is enough, including hierarchy, time series and utility mining. Depending on whether some performance booster is needed, a parallel approach may be helpful. Also, in case of huge volumes and reduced run time needs, a distributed approach seems to be the right choice. Each of these classes has different requirements in terms of infrastructure, from the more elementary, which requires a good computer, to the more complex, which requires a complex set of infrastructure components and tools.

4.4 Validity Evaluation

This review followed a protocol that was developed and based on a well-known method for the literature review (Kitchenham and Charters 2007) so that other reviews following the same protocol may achieve similar results. Nevertheless, the main threat to validity is presenting a research bias. Hence, to mitigate this risk, the research objectives and the protocol were previously aligned among the authors to increase confidence in the results achieved. Furthermore, ARM is a much larger field than the one presented here. This study left many methods out of this analysis due to the small number of citations, even though they may be used to extract information and knowledge for the EA model. However, this review prioritized those most cited for each category analyzed, implying that some techniques and opportunities may remain unexplored. The objective of presenting and exploring the use of ARM in the context of EA mining was achieved. Nonetheless, this

Table 6 Parallel ARM

Algorithm	Citations	Reference
NOV-CFI	10,104	Han et al. (2000)
MTARM	900	Zaki and Gouda (2003)
PEM	557	Uno et al. (2004)
UDDAG	48	Chen (2010)
MPSoc	4	Sinaei and Fatemi (2018)
Tax et Al	13	Tax et al. (2018)

Table 7 Distributed ARM

Algorithm	Citations	Reference
FDM	728	Cheung et al. (1996b)
AprioriMR	68	Luna et al. (2018)
G3P-LSC	47	Padillo et al. (2018)
BIGMiner	33	Chon and Kim (2018)
MRQAR	30	Martín et al. (2018)
LA-FEMH	29	Ao et al. (2019)

paper presents a partial and non-exhaustive view of the field's opportunities, leaving space for further explorations.

5 Discussion

This section provides a consolidated view of the literature previously presented, discusses the findings and opportunities for using ARM techniques for EA model mining and creates hypotheses for its application.

5.1 Findings

Although ARM has been applied to a variety of specific fields, this research did not find papers related to the direct application of ARM to EA modelling, which reinforces the opportunities in this field.

It is a fact that there is a large quantity and variety of algorithms for ARM. This variety makes the task of choosing which algorithm to use more difficult. For example, according to Gan et al. (2019), SPMF (Sequential Pattern Mining Framework),[2] an ARM library for Java, provides a collection of 120 algorithms, and it does not yet cover any parallel algorithms. Related to the selection of the ARM approach and algorithms, it is also challenging to correlate studies and algorithms to find the best and latest advances, mainly due to the high number and variations of the algorithms.

It is also observable that most papers do not clearly describe the research methodology followed, especially regarding how they gathered the state of the art. Thus, there is low confidence in the quality of the related work presented by these papers in comparative experiments, which may have been biased or ignored relatively new advances in working time. Most methods were compared with the different evolution of the same classical or ancient versions of algorithms, such as those based on the Apriori (Agrawal et al. 1993), FUP (Cheung 1997) and FP-Growth (Han et al. 2004) algorithms, implying that some of them look similar even though there hasn't been a comparison between them yet. Thus, it might be an outstanding contribution to compare the latest proposed algorithms under the same topic, as listed in Appendix A: Algorithm Comparison, and previously introduced in the result of the review in response to RQ 2 (online appendices are available via http://link.springer.com).

[2] https://www.philippe-fournier-viger.com/spmf/index.php?link=algorithms.php.

5.2 Implications

This review did not successfully identify studies demonstrating concrete examples and challenges to using ARM for EA Mining. It does not mean there are no references to aid research advances in the field, as ARM has been used in many fields. Some applications and algorithms may be adapted to the EA mining context. Moreover, the existence of a large number of algorithms and their variations to attend to various specific needs proves ARM's flexibility and suggests the feasibility of its application to EA mining.

5.3 Application Hypothesis

At this point, a hypothesis for applying some of the approaches found in the EA modelling demonstrates a high level and initial view on how EA modelling could use ARM. Despite not being found in the literature through ARM, it is virtually possible to extract a variety of models and viewpoints. However, it is possible to simulate ARM application to EA modelling to share insights for future works, even at an initial and superficial level. In this regard, Fig. 11 shows the ArchiMate structure for EA views and provides the starting point for the analysis.

Firstly, it is possible to identify rules and extract viewpoints related to Business Processes using time series mining. Niazmand (2022) applied some knowledge to Graph Mining that identifies and relates data entities. The same approach seems to be easily adapted to correlate business entities that will allow extracting business objects and business processes, including identifying actors and their roles in the business process. Frequent Event Mining fits to gather behavior in any of the tree layers. Combined with a temporal event perspective, it can help to identify business processes dynamically. In this sense, Tax et al. (2018) identify a sequence of events through local process mining. Thus, to adapt the idea to EA mining, it is necessary to analyze the right granularity of the processes to be adequate for EA and map what kind of event represents this granularity from the EA perspective. The Lin et al. (2020) approach, used to detect malfunctions in services, hardware and software, could be adapted to mining from this kind of solution knowledge by enabling components to be fulfilled dynamically at the technology layer, such as IT network systems.

These few hypotheses just explore possible uses of ARM to model some aspects of an EA model. However, the crucial diagnostic from this review is that ARM is virtually applicable to almost any application that demands knowledge from data, which is the case of EA mining. It is a field which few studies have explored and presents an excellent opportunity for future research.

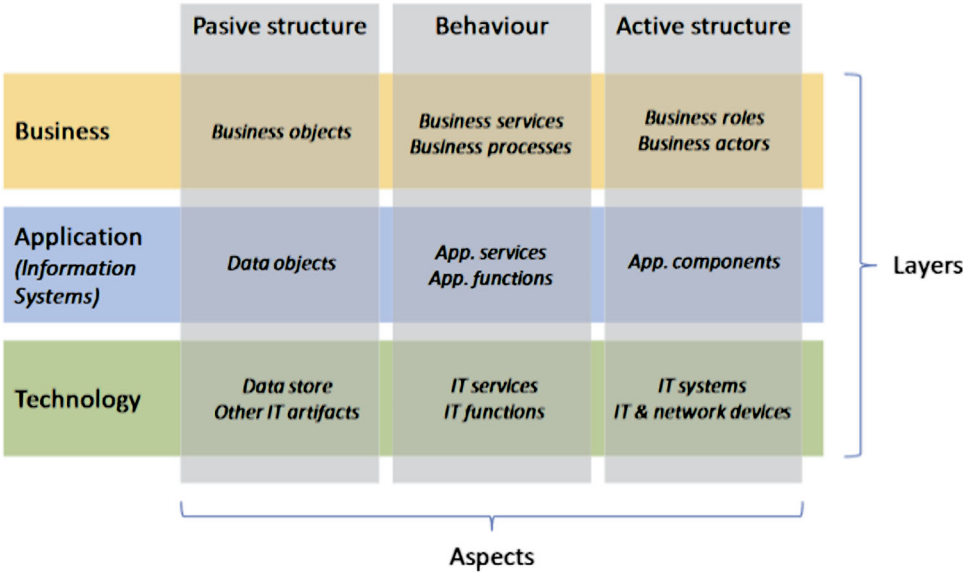


Fig. 11 ArchiMate 3 core framework (adapted from The Open Group 2019) (Perez-Castillo et al. 2019)

EA usually deals with the general views of company architecture, encompassing all business activities, capabilities, information, and technology of an enterprise. Here, this study describes some opportunities to explore the application of identified ARM techniques to contribute to the discovery, build the current state of some EA viewpoints, and address enterprise transversal architecture-wise concerns. It is not limited to the TOGAF framework. However, this study uses TOGAF and ArchiMate as references (The Open Group 2018, 2019). According to Greefhorst and Proper (2011), TOGAF is a standardized method for enterprise architecture maintained by The Open Group, a consortium of hundreds of organizations, including companies, governmental organizations and research institutes. ArchiMate is a modelling language adopted by the Open Group, developed as part of a research project to provide a language for describing enterprise architectures. Together, they define a method and a language for building enterprise architectures by identifying relevant building blocks in three domains: Business Architecture, Information System Architecture (with Data and Application Architecture), and Technological Architecture.

For the business architecture domain, it seems feasible to create rules based on indicators that may help identify and correlate events and process flow that occurs among the business process within the enterprise architecture. It may enable a deeper control of the process, identify hidden process flows and adapt the business process views in the architecture to these actual flows. Also, it may correlate with relevant events, such as those related to some enterprise's key performance indicators (KPI) of business processes. Business objectives are usually defined in the business architecture model, which is broken down into one or more Objective Key Results (OKR) that must be actively monitored. For instance, through ARM, it may be possible to associate events related to the enterprise business objective of reducing customer churn, which happens when the customer stops consuming the company's products or services. Thus, ARM expands the EA management beyond the definition of OKR, enabling its active monitoring as a tool of the enterprise architecture itself. It allows quick modelling and drives courses of corrective action to continuously improve processes related to these enterprise business indicators.

Still, in mapping business processes, temporal-related mining may identify patterns of business interactions from inter or intra-organizational processes and depict how other partner companies, or even internal business areas, interact with some line of business, for example. It may help identify how the best partners interact with the business product and services and the behavior of more problematic partners. Thus, the enterprise and business architect may plan a course of action to help business partners improve their business interaction with the product.

In the information system domain, it seems feasible to automatically:

- Obtain a model of enterprise applications supporting one or more process executions,
- Identify services and application functions used by the process,
- Search for patterns of processing inefficiencies, such as finding frequent disruptions that may indicate bad architectural design or event problems related to gaps in data and absent or noisy information,
- Based on mining objectives, it may generate new indicators and information measures to improve the

monitoring of the application's architectural performance and check new behavior patterns of applications.

In addition, combining EA mining with process mining (van der Aalst et al. 2012) takes advantage of existing process mining approaches as an EA mining accelerator. Process mining is a DM approach focused on learning the actual execution flow of processes. Despite this focus, it can aid in extracting more aggregated information about the processes in a bottom-up approach to build an enterprise architecture viewpoint. The challenge is to find a suitable way to incorporate and aggregate the process mining results into the EA model visualizations.

In the technology architecture, using the ARM approach can contribute to demonstrating how stakeholder concerns are being addressed by collecting indicators for these concerns from events as close as possible to real-time. For instance, some key EA stakeholders may be concerned about monitoring customer success. In this sense, it may be feasible to use ARM to correlate customer behaviors with product or service details that positively or negatively impact these customers. Another example is the frequency patterns at which consumers access customer service or product/service support or identify patterns in the customer experience that reinforce or hinder their success in their journeys with the company. It may also map and analyze the relationship between application components and the technology that supports it and show how applications use technology like cloud services. For example, in a multi-cloud environment, it may analyze cloud and cost consumption patterns by enterprise applications to find ways to optimize the IT cost operation. The IT operation may be a subject of pattern mining to show the frequency of outages, high latency peaks and others contributing to automatic monitoring services, but mainly to get insight related to services and IT-based product performance. It can also provide an extended view of matrix diagrams depicting the software distribution, how technology platforms support applications, and how intensively the technology is used for applications. It can be depicted using a tree map chart or another kind of graph that can be incorporated into the enterprise architecture modelling tool.

Furthermore, it is vital to proceed with gap analyses on overall architecture related to all the architecture domains to validate that the EA models support the business objectives, principles, policies, and constraints. For instance, a frequent pattern of process failure or the most frequent failure to scale up services could demonstrate inefficiency related to the business objective to keep customers experiencing excellence or denounce patterns of interruptions that may offend the business continuity as an architectural principle. In addition, ARM techniques may support and complement views on core relationships and hidden dependencies among the business, application, or technology components. It can also provide new ways to perform impact analysis, essential to understanding any hidden and broader impacts, providing tools to monitor the real impacts of changes made in an EA component as they occur and identify architecture bottlenecks at the run-time of those components.

Exploring ARM techniques in a bottom-up modelling strategy to discover enterprise architectural viewpoints can bring economic benefits to organizations, consuming less specialized time than gathering information based on interviews and inspections to discover and keep the current architecture up to date. It reduces the risk of errors provoked by manual modelling and misunderstanding and supports better and faster architectural decision-making (Perez-Castillo et al. 2019). On the other hand, it might provide new ways for architecture visualization and planning top-down courses of action based on insights obtained through applying ARM to EA mining.

However, based on the works gathered in this review, the application of ARM in enterprise architecture models is still in its infancy. It constitutes a promising research field that may help automate an enterprise architect's tasks. It can provide new viewpoints and visualizations for EA concerns to support a more agile EA management, finding new ways to optimize the architecture regarding services, infrastructure, and business alignment.

5.4 Example

To exemplify, let us consider a company with physical stores, a mobile app and an e-commerce portal, a logistic partner responsible for product delivery and a partner that provides payment services. Now let us consider the following hypothetical sequence of service calls through API (Application Program Interface) calls for a general e-commerce context where the APIs are the most common behavior of users following a path in search of products. Once they identify the desired product, they put it in a shopping cart created when the first product is added. In the sequence, the users add and remove products from the cart. Then, when satisfied, users check out the cart and are asked to create an account or log in. Once logged, they pay, and once the payment is approved, it creates an order that is sent to the logistics sector for delivery. At the end of this hypothetical typical process, both logistics and the user confirm the product delivery and its reception by the user. Figure 12 illustrates this hypothetical Process.

For this study, the first API call is the one that creates the shopping cart. To build an AE Process View, it could apply an ARM process that correlates *Process Events* in a temporal sequence, considering each API request as an atomic event and identifying in the sequence which event

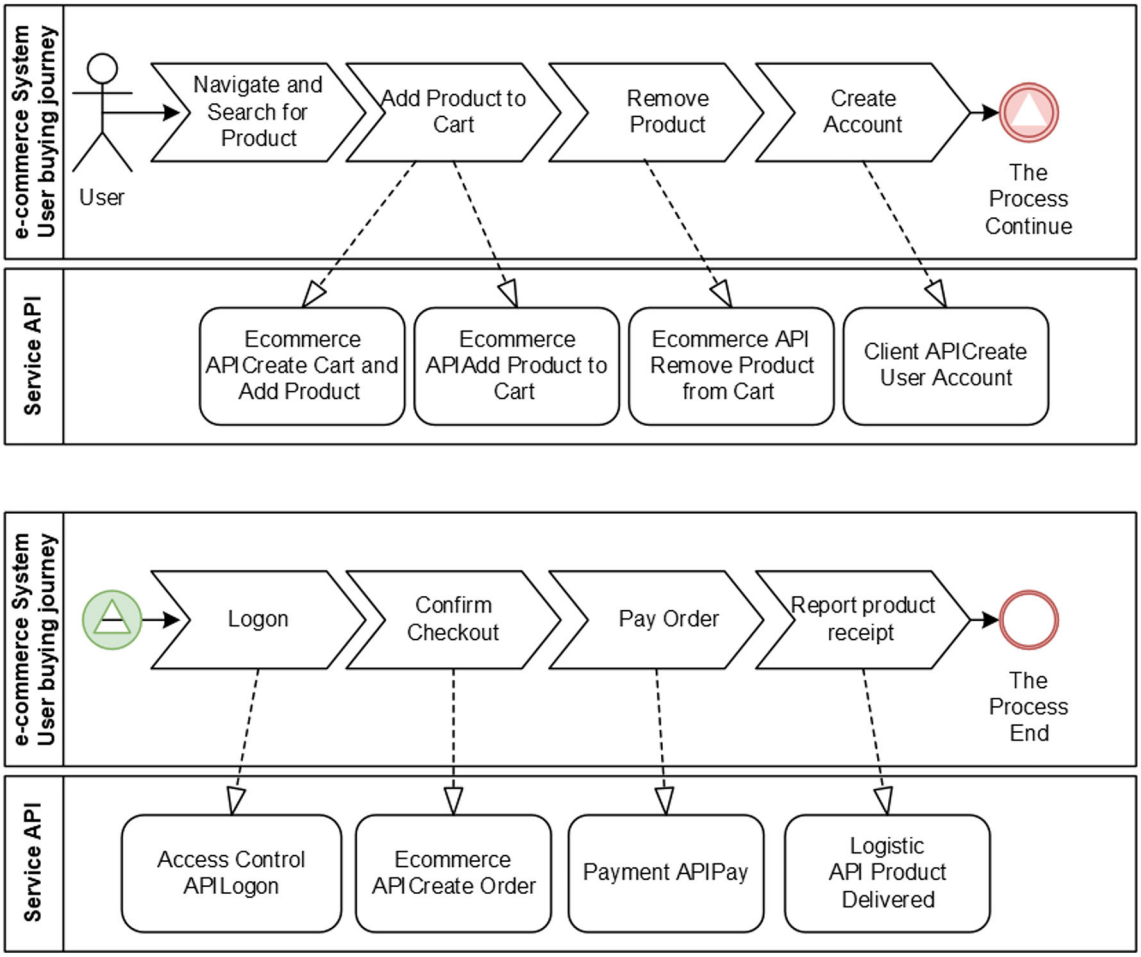


Fig. 12 Example of user buying process

represents the *Antecedent Activity* and which one is the *Consequent Activity,* grouping and classifying the event context. The candidate activities are identified and grouped considering the same caller of the APIs. Then, through temporal association rules, it is possible to identify the sequence candidate and assign a classification context to each sequence, identifying and building individual instances for each Process, which is composed of an activities chain with antecedents and consequents under the same context. The *Process Instances* may be grouped based on the first and last activities. Thus, the inner different path sequences may be seen as variations of the same process group and filtered with what is considered irrelevant at the EA level. Thus, these activities should be mapped to ArchiMate elements.

The resulting model is an ArchiMate viewpoint depicted in Fig. 13. Despite some differences in the sequences that do not correspond precisely to the same graph, all paths will start with the creation of the cart and finish with the final update of the order status. In the end, the process sequence for the view is assembled based on the most frequent sequence involving all the activity once.

The previous example is based on a hypothetical scenario, and it is also a high-level demonstration of how to apply ARM to build a process to transform data mined through ARM to an ArchiMate model. This is only one possibility among many others. Nonetheless, this view still needs implementation with an actual case in future work.

6 Conclusion and Future Work

Exploring ARM techniques to discover enterprise architectural viewpoints is the most valuable innovation proposed by this paper. It promises to enable more agile discovery and maintenance of EA models and improve its governance through architectural performance visualizations by discovering hidden knowledge from architecture execution data. Hence, this review also aimed to find the state-of-the-art ARM techniques applicable to discover

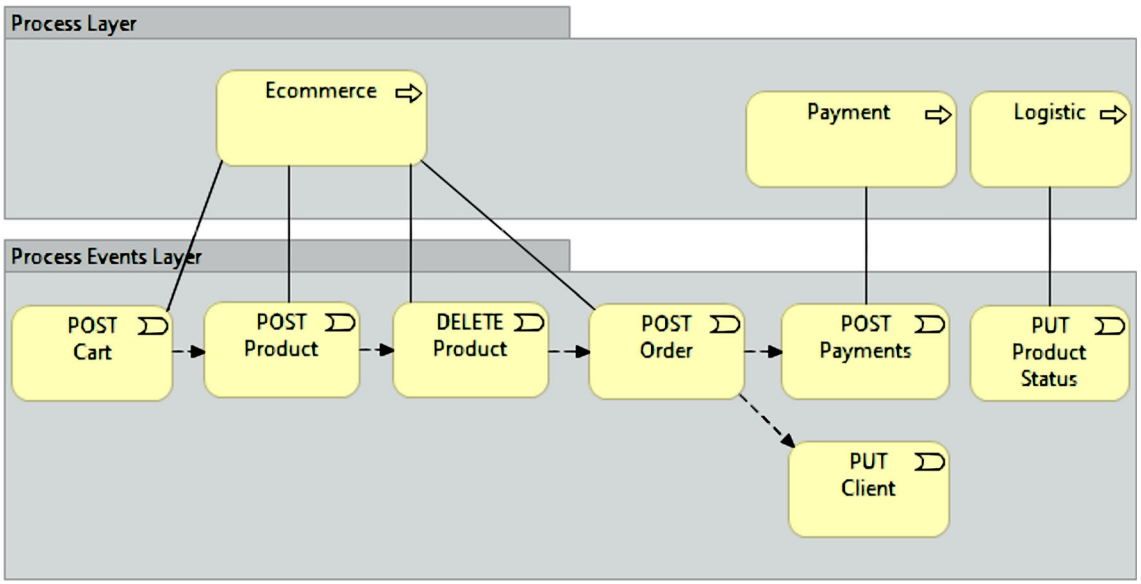


Fig. 13 ArchiMate business process view prototype

significant association rules for building enterprise architectural models and viewpoints.

ARM, in general, seems to be easily described in enterprise architecture as an application tool for data architecture to support diverse needs, analogous to any other tools of business intelligence or data science. However, this research presented ARM as a tool to capture, design, and evolve the enterprise architecture model itself in all its domains. The discussion presented and exemplified ARM application cases related to some projected results for the enterprise architecture as described in the TOGAF framework, which is one of the most known EA frameworks. However, these examples are most likely only a small part of the possibilities which were described in the initial and high-level insights and probably open space for deepening these and other opportunities.

One important recommendation is that in the case discussed where ARM is used for EA mining, ARM should be designed and modelled in the EA and stay present as part of the EA model. It provides new complementary visualizations that enable enriching knowledge about the architecture with less effort and can support bottom-up EA modelling strategies. This capability is especially helpful in designing further architectural improvements in the organizational structures, products, services, and adjacent technologies. It also potentially contributes to linking the high-level and abstract world of EA to its concrete realization based on the actual data collected from business processes and application execution, making the architecture management work more dynamically with information near real-time. It is also more realistic, with fewer misunderstandings and errors, and more insightful with knowledge discovery from data.

Based on the few papers available that investigate the application of some data mining techniques to automatize EA modelling tasks, it is possible to affirm that few studies have explored the field, in contrast with the high number found in ARM research. However, some possibilities of its application were explored in this paper using as a basis some viewpoints within the TOGAF domains: business, information system and technology, which have the potential to guarantee accurate information on the actual and current state of the enterprise architecture, saving experts time, avoiding architecture misunderstandings and supporting faster and better decisions driven by data, using ARM techniques.

From the perspective of the algorithms suitable for an EA mining strategy, fourteen candidate algorithms were identified for four distinct topics incorporating more than 21 overpassed algorithms, embracing a total of 35 ARM algorithms that were covered. Despite the great diversity and difficulty in relating these algorithms and tracking their evolution, the newer and most cited algorithms were mapped. It also indicates those algorithms with lower performance. The generated map will help future analysis focus on the best algorithms while avoiding wasting time analyzing an outdated algorithm. This analysis can also be a starting point for further investigations in the field.

In future work, it will be helpful to conduct laboratory experiments to compare the performance of candidate algorithms identified in this research against the requirements for its application in the EA modelling context. Furthermore, it seems necessary to confirm the comparison

among the latest algorithms with new experiments to generate a firmer list of selected algorithms for EA mining.

The opportunities for architecture viewpoints presented in this research were a very preliminary insight into ARM applicability to EA mining. It strongly indicates its utility. However, it must be complemented with a comprehensive observation to provide a broader view of the opportunities of its applications.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12599-023-00844-5.

Acknowledgements This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020. The second author was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 (INESC-ID).

Funding Open access funding provided by FCT|FCCN (b-on).

References

Agarwal RC, Aggarwal CC, Prasad VVV (2000) Depth first generation of long patterns. In: Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining, Boston. ACM, pp 108–118. https://doi.org/10.1145/347090.347114

Aggarwal A, Toshniwal D (2018) Spatio-temporal frequent itemset mining on web data. In: 2018 IEEE international conference on data mining workshops, pp 1160–1165. https://doi.org/10.1109/ICDMW.2018.00166

Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on management of data. ACM, New York, pp 207–216. https://doi.org/10.1145/170035.170072

Agrawal R, Shafer J (1996) Parallel mining of association rules: design, implementation and experience. IBM Research Division, San Jose

Ao X, Shi H, Wang J, Zuo L, Li H, He Q (2019) Large-scale frequent episode mining from complex event sequences with hierarchies. ACM Trans Intell Syst Technol 10(4):1–26. https://doi.org/10.1145/3326163

Barkhordari M, Niamanesh M (2018) Kavosh: an effective map-reduce-based association rule mining method. J Big Data 5(1):25. https://doi.org/10.1186/s40537-018-0129-4

Cai K, Chen H, Ai W, Miao X, Lin Q, Feng Q (2022) Feedback convolutional network for intelligent data fusion based on near-infrared collaborative IoT technology. IEEE Trans Ind Inform 18(2):1200–1209. https://doi.org/10.1109/TII.2021.3076513

Chen J (2010) An updown directed acyclic graph approach for sequential pattern mining. IEEE Trans Knowl Data Eng 22(7):913–928. https://doi.org/10.1109/TKDE.2009.135

Cheung D, Han J, Ng V, Wong C (1996a) Maintenance of discovered association rules in large databases: an incremental updating technique. In: Proceedings 1996 international conference on data engineering, New Orleans. https://doi.org/10.1109/ICDE.1996.492094

Cheung DW, Han J, Ng VT, Fu AW, Fu Y (1996b) A fast distributed algorithm for mining association rules. In: Fourth international conference on parallel and distributed information systems, pp 31–42. https://doi.org/10.1109/PDIS.1996.568665

Cheung DW, Lee SD, Kao B (1997) A general incremental technique for maintaining discovered association rules. In: Database systems for advanced applications '97, pp 185–194. https://doi.org/10.1142/9789812819536_0020

Chon K-W, Kim M-S (2018) BIGMiner: a fast and scalable distributed frequent pattern miner for big data. Cluster Comput 21(3):1507–1520. https://doi.org/10.1007/s10586-018-1812-0

da Cunha DS, Xavier RS, Ferrari DG, Vilasbôas FG, de Castro LN (2018) Bacterial colony algorithms for association rule mining in static and stream data. Math Probl Eng 2018:e4676258. https://doi.org/10.1155/2018/4676258

Datta S, Mali K (2021) Significant association rule mining with high associability. In: 5th international conference on intelligent computing and control systems, pp 1159–1164. https://doi.org/10.1109/ICICCS51141.2021.9432237

Djenouri Y, Djenouri D, Belhadi A, Cano A (2019) Exploiting GPU and cluster parallelism in single scan frequent itemset mining. Inform Sci 496:363–377. https://doi.org/10.1016/j.ins.2018.07.020

Farwick M, Schweda CM, Breu R, Hanschke I (2016) A situational method for semi-automated enterprise architecture documentation. Softw Syst Model 15(2):397–426. https://doi.org/10.1007/s10270-014-0407-3

Gan W, Lin JC-W, Fournier-Viger P, Chao H-C, Yu PS (2019) A survey of parallel sequential pattern mining. ACM Trans Knowl Discov Data 13(3):1–34. https://doi.org/10.1145/3314107

Greefhorst D, Proper E (2011) The role of enterprise architecture. In: Greefhorst D, Proper E (eds) Architecture principles: the cornerstones of enterprise architecture. Springer, Heidelberg, pp 7–29. https://doi.org/10.1007/978-3-642-20279-7_2

Gullo F (2015) From patterns in data to knowledge discovery: what data mining can do. Phys Proc 62:18–22. https://doi.org/10.1016/j.phpro.2015.02.005

Gustavsson PM, Planstedt T (2005) The road towards multi-hypothesis intention simulation agents architecture—fractal information fusion modeling. In: Proceedings of the winter simulation conference. https://doi.org/10.1109/WSC.2005.1574548

Han JW, Pei J, Yin YW (2000) Mining frequent patterns without candidate generation. SIGMOD Rec 29(2):1–12. https://doi.org/10.1145/335191.335372

Han J, Pei J, Yin Y, Mao R (2004) Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min Knowl Discov 8(1):53–87. https://doi.org/10.1023/B:DAMI.0000005258.31418.83

Karthik S, Medvidovic N (2019) Automatic detection of latent software component relationships from online Q&A sites. In: IEEE/ACM 7th international workshop on realizing artificial

intelligence synergies in software engineering, pp 15–21. https://doi.org/10.1109/RAISE.2019.00011
Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. EBSE technical report EBSE-2007–01. Keele, Staffs, and Durham. https://citeseerx.ist.psu.edu/doc/10.1.1.117.471. Accessed 6 Mar 2022
Kitchenham B (2004) Procedures for performing systematic reviews. Keele University. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=29890a936639862f45cb9a987dd599dce9759bf5. Accessed 9 May 2022
Kiteley R, Stogdon C (2014) Literature reviews in social work. Sage, London. https://doi.org/10.4135/9781473957756
Laudon K, Laudon JP (2021) Management information systems: managing the digital firm, global edition. Pearson. https://books.google.com.br/books?id=AqJXzgEACAAJ. Accessed 14 Nov 2021
Le T, Vo B (2015) An N-list-based algorithm for mining frequent closed patterns. Expert Syst Appl 42(19):6648–6657. https://doi.org/10.1016/j.eswa.2015.04.048
Li H, Wang Y, Zhang D, Zhang M, Chang EY (2008) Pfp: parallel fp-growth for query recommendation. In: Proceedings of the ACM conference on recommender systems, pp 107–114. ACM, New York. https://doi.org/10.1145/1454008.1454027
Liang Y-H, Wu S-Y (2015) Sequence-growth: a scalable and effective frequent itemset mining algorithm for big data based on MapReduce framework. In: IEEE international congress on big data, pp 393–400. https://doi.org/10.1109/BigDataCongress.2015.65
Lin F, Muzumdar K, Laptev NP, Curelea M-V, Lee S, Sankar S (2020) Fast dimensional analysis for root cause investigation in a large-scale service environment. Proc ACM Meas Anal Comput Syst 4(2):1–23. https://doi.org/10.1145/3392149
Lin M-Y, Lee P-Y, Hsueh S-C (2012) Apriori-based frequent itemset mining algorithms on MapReduce. In: Proceedings of the 6th international conference on ubiquitous information management and communication. ACM, New York. https://doi.org/10.1145/2184751.2184842
Liu X, Zhang X, Wang Y, Zhou J, Helal S, Xu Z, Cao S (2018) PARMTRD: parallel association rules based multiple-topic relationships detection. In: Jin H et al (eds) Web Services—ICWS 2018. Springer, Cham, pp 422–436. https://doi.org/10.1007/978-3-319-94289-6_27
Liu X, Niu X, Fournier-Viger P (2021) Fast Top-K association rule mining using rule generation property pruning. Appl Intell 51(4):2077–2093. https://doi.org/10.1007/s10489-020-01994-9
Liu X, Zheng L, Zhang W, Zhou J, Cao S, Yu S (2022) An evolutive frequent pattern tree-based incremental knowledge discovery algorithm. ACM Trans Manag Inf Syst 13(3):1–20. https://doi.org/10.1145/3495213
Liu J, Wang K, Fung BCM (2012) Direct discovery of high utility itemsets without candidate generation. In: IEEE 12th international conference on data mining, pp 984–989. https://doi.org/10.1109/ICDM.2012.20
Luna JM, Padillo F, Pechenizkiy M, Ventura S (2018) Apriori versions based on MapReduce for mining frequent patterns on big data. IEEE Trans Cybern 48(10):2851–2865. https://doi.org/10.1109/TCYB.2017.2751081
Luna JM, Fournier-Viger P, Ventura S (2019) Frequent itemset mining: a 25 years review. Wires Data Min Knowl Discov 9(6):e1329. https://doi.org/10.1002/widm.1329
Martín D, Martínez-Ballesteros M, García-Gil D, Alcalá-Fdez J, Herrera F, Riquelme-Santos JC (2018) MRQAR: a generic MapReduce framework to discover quantitative association rules in big data problems. Knowl-Based Syst 153:176–192. https://doi.org/10.1016/j.knosys.2018.04.037
Menaga D, Saravanan S (2021) GA-PPARM: CONSTRAINT-based objective function and genetic algorithm for privacy preserved association rule mining. Evolut Intell. https://doi.org/10.1007/s12065-021-00576-z
Modaresnezhad M, Vahdati A, Nemati H, Ardestani A, Sadri F (2019) A rule-based semantic approach for data integration, standardization and dimensionality reduction utilizing the UMLS: application to predicting bariatric surgery outcomes. Comput Biol Med 106:84–90. https://doi.org/10.1016/j.compbiomed.2019.01.019
Moens S, Aksehirli E, Goethals B (2013) Frequent itemset mining for big data. In: IEEE international conference on big data, pp 111–118. https://doi.org/10.1109/BigData.2013.6691742
Neaga EI, Harding JA (2005) An enterprise modeling and integration framework based on knowledge discovery and data mining. Int J Prod Res 43(6):1089–1108. https://doi.org/10.1080/00207540412331322939
Niazmand E (2022) Enhancing query answer completeness with query expansion based on synonym predicates. In: Companion proceedings of the web conference, pp 354–358. ACM, New York. https://doi.org/10.1145/3487553.3524198
Noori FM, Riegler M, Uddin MZ, Torresen J (2020) Human activity recognition from multiple sensors data using multi-fusion representations and CNNs. ACM Trans Multimed Comput Commun Appl 16(2):1–19. https://doi.org/10.1145/3377882
Onan A (2019) Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering. IEEE Access 7:145614–145633. https://doi.org/10.1109/ACCESS.2019.2945911
Padillo F, Luna JM, Herrera F, Ventura S (2018) Mining association rules on big data through MapReduce genetic programming. Integr Comput-Aided Eng 25(1):31–48. https://doi.org/10.3233/ICA-170555
Perez-Castillo R, Ruiz-Gonzalez F, Genero M, Piattini M (2019) A systematic mapping study on enterprise architecture mining. Enterp Inform Syst 13(5):675–718. https://doi.org/10.1080/17517575.2019.1590859
Pérez-Castillo R, Ruiz F, Piattini M (2020) A decision-making support system for enterprise architecture modelling. Decis Support Syst 131:113249. https://doi.org/10.1016/j.dss.2020.113249
Pérez-Castillo R, Caivano D, Ruiz F, Piattini M (2021) ArchiRev—reverse engineering of information systems toward archimate models an industrial case study. J Softw Evol Proc 33(2):e2314. https://doi.org/10.1002/smr.2314
Phan H (2018) NOV-CFI: a novel algorithm for closed frequent itemsets mining in transactional databases. In: Proceedings of the VII international conference on network, Communication and computing, pp 58–63. ACM, New York. https://doi.org/10.1145/3301326.3301363
Pinheiro CR, Guerreiro S, Mamede HS (2021) Automation of enterprise architecture discovery based on event mining from API gateway logs: state of the art. In: IEEE 23rd conference on business informatics, pp 117–124. https://doi.org/10.1109/CBI52690.2021.10062
Sinaei S, Fatemi O (2018) Run-time mapping algorithm for dynamic workloads using association rule mining. J Syst Arch 91:1–10. https://doi.org/10.1016/j.sysarc.2018.09.005
De Stefano M, Pecorelli F, Tamburri DA, Palomba F, De Lucia A (2020) Splicing community patterns and smells: a preliminary study. In: Proceedings of the IEEE/ACM 42nd international conference on software engineering workshops, pp 703–710. ACM, New York. https://doi.org/10.1145/3387940.3392204
Tax N, Sidorova N, Haakma R, van der Aalst WMP (2018) Mining local process models with constraints efficiently: applications to the analysis of smart home data. In: 14th international

conference on intelligent environments, pp 56–63. https://doi.org/10.1109/IE.2018.00016

The Open Group (2018) The TOGAF® standard, version 9.2. https://publications.opengroup.org/standards/togaf/c182. https://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html. Accessed 28 Apr 2022

The Open Group (2019) ArchiMate® 3.1 Specification. https://pubs.opengroup.org/architecture/archimate3-doc/. Accessed 15 Apr 2022

Uno T, Kiyomi M, Arimura H (2004) LCM ver. 2: efficient mining algorithms for frequent/closed/maximal itemsets. FIMI '04, p 126. https://ceur-ws.org/Vol-126/uno.pdf. Accessed 27 Feb 2022

van der Aalst W, Adriansyah A, de Medeiros AKA, Arcieri F, Baier T, Blickle T, Wynn M (2012) Process mining manifesto. In: Daniel F et al (eds) Business Process Management Workshops. Springer, Heidelberg, pp 169–194. https://doi.org/10.1007/978-3-642-28108-2_19

Wu JM-T, Lin JC-W, Tamrakar A (2019) High-utility itemset mining with effective pruning strategies. ACM Trans Knowl Discov Data 13(6):1–22. https://doi.org/10.1145/3363571

Xun Y, Zhang J, Qin X (2016) FiDoop: parallel mining of frequent itemsets using MapReduce. IEEE Trans Syst Man Cybern: Syst 46(3):313–325. https://doi.org/10.1109/TSMC.2015.2437327

Yildirim Taşer P, Birant KU, Birant D (2020) Multitask-based association rule mining. Turk J Elec Eng Comput Sci 28(2):933–955. https://doi.org/10.3906/elk-1905-88

Zaki MJ (2000) Scalable algorithms for association mining. IEEE Trans Knowl Data Eng 12(3):372–390. https://doi.org/10.1109/69.846291

Zaki MJ, Gouda K (2003) Fast vertical mining using diffsets. In: Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining, pp 326–335. ACM, New York. https://doi.org/10.1145/956750.956788

Zaki MJ, Hsiao C-J (2002) CHARM: an efficient algorithm for closed itemset mining. In: Proceedings of the SIAM international conference on data mining, pp 457–473. https://doi.org/10.1137/1.9781611972726.27

Zida S, Fournier-Viger P, Lin JC-W, Wu C-W, Tseng VS (2015) EFIM: a highly efficient algorithm for high-utility itemset mining. In: Sidorov G, Galicia-Haro SN (eds) Advances in artificial intelligence and soft computing. Springer, Cham, pp 530–546. https://doi.org/10.1007/978-3-319-27060-9_44