International Journal of Data Warehousing and Mining
Volume 21 • Issue 1 • January-December 2025

MRE-KDD+:
An Innovative Multi-Resolution, Ensemble Framework for Supporting OLAM-Based Big Data Analytics Over Big Data Warehouses

Alfredo Cuzzocrea
https://orcid.org/0000-0002-7104-6415
University of Calabria, Italy

Pablo Garcia Bringas
https://orcid.org/0000-0003-3594-9534
University of Deusto, Spain
Received: January 15th, 2025 | **Accepted:** November 19th, 2025

ABSTRACT

Big data settings are currently evolving from classical systems that focus on supporting advanced decision-support processes—as applied to many real-life scenarios, which are typically populated by distributed and heterogeneous data sources, such as conventional distributed data warehousing environments—to cooperative information systems. Different data formats contribute to define challenging big data systems, in which the main issue consists in supporting modern big data analytics involving massive amounts of data. As a consequence, a relevant research challenge is how to efficiently integrate, process, and mine such distributed knowledge, which composes the foundations of final big data analytics processes. Starting from these considerations, in this paper the authors propose an online analytical mining-based framework for supporting big data analytics, along with a formal model underlying this framework, called Multi-Resolution Ensemble-Based Model for Advanced Knowledge Discovery in Big Data Warehouses.

KEYWORDS

Big Data, Big Data Management, Big Data Analytics, Big Data Warehouses, OLAM-Based Big Data Analytics

INTRODUCTION

Big data settings (see Desgourdes and Ram, 2024; Fernandes et al., 2023; Zhang et al., 2015) are currently evolving from classical systems that focus on supporting advanced decision-support processes—as applied to many real-life scenarios, which are typically populated by distributed and heterogeneous data sources, such as conventional distributed data warehousing environments—to cooperative information systems. Different data formats contribute to define challenging big data systems, in which the main issue consists in supporting modern *big data analytics* (see Cuzzocrea et al., 2011; Russom, 2011; Tsai et al., 2015) from massive amounts of data. Indeed, in such a big data system, data repositories exhibit widely varied formats, and knowledge representation schemes are accordingly highly heterogeneous. As a consequence, a relevant research challenge is how to

DOI: 10.4018/IJDWM.395849

efficiently integrate, process, and mine such distributed knowledge, which composes the foundations of final big data analytics processes.

Indeed, the problem of supporting advanced decision-support processes arises in many fields of real-life big data applications, ranging from distributed corporate data storage systems to e-commerce systems, and from supply-chain management systems to *big data warehouses* (see Cuzzocrea et al., 2013; Jukić et al., 2015; Santos et al., 2017), where huge, heterogeneous big data repositories are cooperatively integrated in order to produce, process, and mine useful knowledge. Here, the use of artificial intelligence (AI) in data warehousing (DW) and data mining (DM) (see Chai et al., 2023; Li, 2024; Luo & Xu, 2024; Masmoudi et al., 2024; Sun, 2024; Tian et al., 2024; Wang et al., 2022) plays a major role. Our research proposal falls in this context.

In such big data scenarios, intelligent applications run on top of vast, heterogeneous data sources, ranging from transactional data to extensible markup language data and from workflow-process log-data to sensor network data. Here, collected data are typically represented, stored, and queried in large big data warehouses, which, without any loss of generality, define a collection of *distributed and heterogeneous big data sources*, each of them executing as a singleton data-intensive software component (e.g., DBMS server, DW server, XDBMS server, etc.). Contrary to this distributed setting, intelligent applications attempt to extract integrated, summarized knowledge from data sources in order to make strategic decisions for a business. Nevertheless, heterogeneity of data and platforms, as well as distribution of architectures and systems, seriously hamper the achievement of this goal, in the specific big data settings. In response to this challenge, research communities have devoted a great deal of attention to this problem, with a wide array of proposals (Fayyad et al., 1996) ranging from DM tools, which concern algorithms for extracting *patterns* and *regularities* from data to knowledge discovery in databases (KDD) techniques, which concern the overall process of discovering useful knowledge from data. These techniques have now migrated to the innovative, mature big data analytics context (see Dedić & Stanier, 2017; Grady, 2016).

Among the plethora of techniques proposed in the active literature to overcome the above-highlighted gap between data and knowledge in big data systems, online analytical mining (OLAM; Do et al., 2015) is a successful solution that integrates online analytical processing (OLAP; Chaudhuri & Dayal, 1997; Gray et al., 1997) with DM in order to provide an integrated methodology for extracting useful knowledge from large databases and data warehouses. The benefits of OLAM have been previously demonstrated (Han, 1997), namely the following:

- DM algorithms can execute on integrated, OLAP-based multidimensional views that are already pre-processed and cleaned.
- Users/applications can take advantage of the interactive, exploratory nature of OLAP tools to decisively enhance the knowledge fruition experience.
- Users/applications can take advantages from the flexibility of OLAP tools in making available a wide set of DM solutions for a given KDD task, so that, thanks to OLAP, different DM algorithms become easily interchangeable in order to decisively enhance the benefits coming from cross-comparative data analysis methodologies over large amounts of data.

Largely, these paradigms have now been applied within the emerging context of big data analytics (see Cuzzocrea, 2015, 2020; Cuzzocrea et al., 2013) with success.

Starting from these considerations, in this paper, we propose an OLAM-based framework for supporting big data analytics, along with a formal model underlying this framework, called Multi-Resolution Ensemble-Based Model for Advanced Knowledge Discovery in Big Data Warehouses (MRE – KDD$^+$), and a reference architecture for such a framework. This leads to a novel paradigm for big data research, the so-called *OLAM-based big data analytics*. On the basis of OLAP principles, MRE – KDD$^+$, which can be reasonably considered as an innovative contribution in this research

field, provides a formal, rigorous methodology for implementing advanced KDD processes in big data settings, but with particular regard to two specialized instances represented by the following:

- a general application scenario populated by distributed and heterogeneous big data sources, such as a conventional big data warehousing environment (e.g., like those that one can find in B2B and B2C *e*-commerce systems; Hu, 2019), and
- the integration/data layer of open big data systems, in which different data sources are integrated in a unique middleware in order to make KDD processes against these data sources transparent to the user.

The notion of complex patterns refers to patterns having a nature more advanced than that of traditional ones such as sequences, trees, graphs, etc. Examples of complex patterns for KDD are multidimensional domains, hierarchical structures, clusters, etc.

Besides the widely acknowledged benefits of integrating OLAM within its core layer (Han, 1997), MRE – KDD$^+$ allows data-intensive applications adhering to the methodology it defines to take advantage of other relevant characteristics, among which we recall the following:

- the multi-resolution support offered by popular OLAP operators/tools (Han & Kamber, 2000), which allows us to execute DM algorithms over integrated and summarized multidimensional views of data at different level of granularity and perspective of analysis, thus sensitively improving the quality of KDD processes;
- the ensemble-based support, which, briefly, consists in meaningfully combining results coming from different DM algorithms executed over a collection of multidimensional views in order to generate the final knowledge and provide facilities at the knowledge fruition layer.

Another contribution of our work is represented by the proposal of KBMiner, a MRE – KDD$^+$–based big data visualization tool (see Cuzzocrea et al., 2007; Keim et al., 2013) that allows us to edit the so-called knowledge discovery tasks (KDT), which realize a graphical formalism for extracting useful knowledge from big data warehouses according to the guidelines drawn by MRE – KDD$^+$. The use of visualization metaphors is widely accepted in the big data community as a state-of-the-art proposal for enhancing the knowledge discovery processes from big data sets.

The remaining part of this paper is organized as follows. In the next section, we survey principles and models of OLAM. In the following section, we outline related work. Next, we present MRE – KDD$^+$ in detail. The subsequent section provides a reference architecture implementing the framework we propose, and a description of its key components. Next, we illustrate the main functionalities of KBMiner, along with some running examples. We then provide three interesting case studies that proof the use of MRE – KDD$^+$ in real-life big data applications. Finally, we outline conclusions of our work and envision future activities in this research field. This paper extends a previous conference paper (Cuzzocrea, 2007) in which we presented the embryonic ideas of the proposed framework. With respect to the previous efforts, in this paper we provide the following:

- an improved focus on emerging big data warehouses;
- complete formal models and algorithms of the MRE – KDD$^+$ framework;
- An extended description of the reference architecture;
- complete case studies of real-life big data applications that confirm the benefits deriving from our research proposals; and
- enhanced bibliographical study.

OLAM: COMBINING OLAP AND DATA MINING

OLAM is a powerful technology for supporting knowledge discovery from large databases and data warehouses that combines OLAP functionalities for representing/processing data with DM algorithms for extracting regularities (e.g., patterns, association rules, clusters, etc.) from data. In so doing, OLAM realizes a proper KDD process.

OLAM was proposed by Han in his fundamental paper (1997), and, in another paper, along with the OLAP-based DM system DBMiner (Han et al., 1996), which can be reasonably considered as the practical implementation of OLAM. In order to emphasize and refine the capability of discovering useful knowledge from huge amounts of data, OLAM reaps the best of both technologies (i.e., OLAP and DM). From OLAP, it derives the following:

- an excellent capacity for storing data, which has been of relevant interest during recent years (see Agrawal et al., 1996; Harinarayan et al., 1996; Karayannidis & Sellis, 2003; Vitter & Wang, 1999; Zhao et al., 1997);
- support for multidimensional and multi-resolution data analysis (Chaudhuri & Dayal, 1997);
- the richness of OLAP operators (Han & Kamber, 2000), such as *roll-up*, *drill-down*, slice-&-dice, *pivot*, etc.;
- the wide availability of a number of query classes, such as *range-* (Ho et al., 1997), *top-k* (Xin et al., 2006) and iceberg (Fang et al., 1998) queries, which have been extensively studied during recent years and which can be used as a baseline for implementing even complex KDD tasks.

From DM, OLAM takes the broad range of techniques available in the literature, each of them oriented to cover a specific KDD task; among these techniques, some are relevant for OLAM, such as mining association rules in transactional or relational databases (Ben-Efraim et al., 2025; Guo et al., 2023; Mokkadem et al., 2024; Nisar & Shaheen, 2023; Shakhovska et al., 2018; Wang & Song, 2019), mining classification rules (Cheeseman & Stutz, 1996; Elder & Pregibon, 1996; Piatetsky-Shapiro, 1991; Quinlan, 1993; Ziarko, 1994), cluster analysis (Ester et al., 1995; Ng & Han, 1994; Zhang et al., 1996), summarizing and generalizing data using data cube (Chaudhuri & Dayal, 1997; Gray et al., 1997; Harinarayan et al., 1996; Ledmi et al., 2023) or attribute-oriented inductive (Han & Fu, 1996; Han et al., 1993) approaches.

According to the guidelines given by Han (1997), there exist various alternatives to mingle the capabilities of OLAP and DM, depending mainly on the way of combining the two technologies. Of these the most relevant ones are the following:

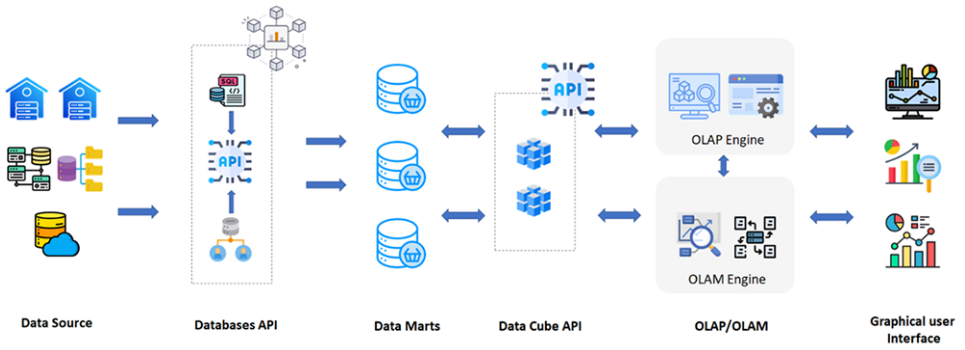
- *cubing-then-mining*, in which the power of OLAP operators generalized in the primitive *cubing* (i.e., the universal operator for generating a new data cube from another one or from a collection of data cubes; Han, 1997) is used to select, pre-process, and interactively process the portion of OLAP data on which DM algorithms are being execute; and
- mining-then-cubing, in which the power of DM is applied to the target data cube directly and then OLAP operators/tools are used to further analyze results of DM in order to improve the overall quality and refinement of the extracted knowledge.

In our opinion, the most effective alternative is that provided by the cubing-then-mining mode, which is, in fact, the preferred solution in most application scenarios. In this mode, OLAP allows us to obtain summarized multidimensional views over large amounts of data, and DM algorithms successfully run on these views to extract knowledge. In other words, in this vest, OLAP is recognized as a very efficient data-support/pre-processing tool for DM algorithms, which also enables high performance in comparison to traditional online transactional processing (OLTP – the technology of relational databases) operators/tools.

In a data warehousing environment, subject-oriented multidimensional views of data sources are very often materialized into the so-called *data marts* (Chaudhuri & Dayal, 1997), which can be generally intended as a sort of specialized data cube*s* built to support specific analysis requirements (e.g., sales, inventory, balance, etc.). Usually, given a collection of heterogeneous data sources, a data warehouse application makes use of several data marts, each of them focused on supporting multidimensional, multiresolution data analysis over a specific application context via OLAP operators/tools. In OLAM, DM algorithms are enabled to run over both a singleton data mart and, more interestingly, *multiple data marts*, thus taking advantage of the amenity of extracting useful knowledge by means of complex JOIN-based OLAP operations over multiple sources.

A reference architecture for the cubing-then-mining OLAM mode (Leprince et al., 2021) is depicted in Figure 1. Here, the *OLAP Engine* and the *OLAM Engine* run in a combined manner in order to extract useful knowledge from a collection of subject-oriented data marts. Beyond the above-described OLAM features, this architecture also supports a leading OLAM functionality, the so-called online interactive mining (Han & Kamber, 2000), which consists in iteratively executing DM algorithms over *different* views extracted from the *same* data mart via cubing. In this case, the effective "add-on" value given by OLAP is represented by a powerful information gain that cannot be easily supported by traditional OLTP operators/tools without introducing excessive computational overheads.

Figure 1. A Reference Architecture for Online Analytical Mining



It should be noted that currently, the issue of supporting OLAP/OLAM over big data plays a major role in big data research (e.g., Chen et al., 2017; Song et al., 2015). In fact, it has already been proven that these methodologies perfectly marry with the goal of applying multidimensional and multiresolution analysis over big data sets. As a consequence, many proposals have focused attention on methodological (Sautot et al., 2021), algorithmic (Tardío et al., 2020), or performance/optimization aspects of this research (Song & Ge, 2011). This evidence conclusively confirms the relevance of the scientific area we investigate in this paper.

RELATED WORK ANALYSIS

Starting from the first DM and KDD experiences, the idea of developing a framework for supporting advanced KDD processes from large databases and data warehouses via embedding complex data representation techniques (like OLAP) and sophisticated knowledge extraction methodologies (like DM) has been of vital interest to the research community. The most important limitations to this ambition are represented by the difficulty of integrating different-in-nature data

representation models/methodologies, and the difficulty of rendering heterogeneous DM algorithms inter-communicating.

While the literature contains a plethora of data representation techniques and DM algorithms, each of them developed for a particular application scenario, frameworks that, with a large vision, integrate several techniques coming from different contexts via synthesizing data warehousing, DM, and KDD principles are very few. Furthermore, while there exists an extremely wide set of DM and KDD tools (a comprehensive overview can be found in Goebel & Gruenwald, 1999), mainly focused on covering a specific KDD task (e.g., association rule discovery, classification, clustering, etc.), very few of them integrate heterogeneous KDD-oriented techniques and methodologies in a unique environment. Among these, the most significant experiences that have deeply influenced our work are DBMiner and WEKA (Witten & Frank, 2005). In the following discussion, we refer to both the environments in the vest of "realizations" of the respective underlying models.

DBMiner is a powerful OLAM-inspired system that allows us both to extract and represent knowledge from large databases and data warehouses, and to mine knowledge via a wide set of very useful data analysis functionalities, mainly OLAP-inspired, such as data/patterns/results browse, exploration, visualization, and intelligent querying. Specifically, at the representation/storage layer, DBMiner makes use of the popular *data cube* model (the foundation of OLAP), first proposed by Gray et al., 1997, in which relational data are aggregated on the basis of a multidimensional and multiresolution vision of data. On the basis of the data cube model, DBMiner makes available to the user a wide set of innovative functionalities ranging from time series analysis to *prediction* of the data distribution of relational attributes and the mining of complex objects (like those found in a GIS); furthermore, DBMiner also offers a data mining query language (DMQL) for supporting the standardization of DM functionalities and their integration with conventional DBMS. Finally, the graphical user interface of DBMiner supports various attractive, user-friendly forms implementing the above-listed features and making them available to the user.

WEKA is a machine learning (ML) environment for efficiently supporting DM activities against large databases; it was designed to aid in decision-support processes by enabling an understanding of which information is relevant for a specific application context, and, consequently, make prediction faster. Similarly to DBMiner, WEKA offers a graphical environment in which users can edit ML technique, test it against external data sets, and study its performance under the stress of various metrics. Moreover, WEKA users, like DBMiner users, are permitted to mine the output knowledge of ML techniques by means of several advanced intelligent visualization components. Unlike DBMiner, WEKA does not make use of a particular data-representation/storage solution to improve data access/management/processing.

It should be noted here that, owing to the nature and goals of both the outlined environments/models, we can claim that DBMiner is closer to our work than it is to WEKA. On the other hand, it is readily apparent that DBMiner overlaps more with the idea of big data analytics than it does with WEKA. This is another point of similarity between MRE – KDD$^+$ and DBMiner, thus showing continuity with previous authoritative research experiences.

In the following discussion, we provide an overview of some recent initiatives that we consider very salient to our investigated scenario, which can be reasonably intended as big data analytics processes over big data warehouses.

Shahid et al. (2021), in the BigO project, presented an innovative DW architecture to address the challenges of handling large-scale and sensitive data in emerging sectors such as healthcare. This flexible data warehouse architecture consists of three layers, including

- a back-end layer for data collection, de-identification, and anonymization;
- an access control layer, which is responsible for role-based permissions and secured views; and
- a controller layer responsible for regulating data access protocols.

Moreover, the proposed architecture employs cutting-edge database systems such as MongoDB for structured data and Cassandra for high-volume time series data, which provides scalability and efficiency. Furthermore, the architecture employs Hadoop HDFS, which provides features that manage large datasets and facilitate frequent updates. By addressing challenges such as *data integration*, large distributed data, and privacy-aware analytics, the contribution of this work provides a straightforward method for developing scalable and safe large data warehouses, particularly for healthcare-sensitive applications. These improvements have a significant impact on the creation of effective and efficient big data analytics frameworks over big data warehouse systems.

Ngo et al. (2020) proposes a robust agricultural data warehouse (ADW) designed to address the challenges of managing large-scale, complex, and heterogeneous agricultural data. ADW architecture provides high storage capacity; flexible schema design using constellation schema; and compatibility with cloud systems. ADW combines high-performance analytics, scalability, and real-time data processing capabilities and leverages modern and high-performance tools such as Hive, MongoDB, and Cassandra. While previous research, like the star schema and entity-relationship models, facilitated basic data integration, they often lacked the ability to handle the multidimensional, non-standardized, and spatial-temporal nature of agricultural data. ADW handles the multidimensional agricultural data by employing a constellation schema, which provides multiple fact tables to address the analytical and decision making requirements. Moreover, it includes advanced extraction, transformation, and loading (ETL) processes and hybrid OLAP for efficient data querying and decision support, significantly improving operational insights.

Sautot et al. (2021) proposed a methodology built upon existing DW methodologies in order to address challenges in refining multidimensional models within constellation schemas. The proposed methodology focuses on the creation of dimension hierarchies that are derived from factual data. Thus, it addresses the limitations of state-of-the-art methods that usually do not provide features for such transformations. Furthermore, the proposed technique provides coherence by maintaining relationships between facts and dimensions while allowing for the creation of dimensions with computed hierarchies. To address issues of usability in large-scale schemas, the study presents reduction strategies with the aim of limiting the complexity of generated hierarchies. These include *roll-up* operations to reduce granularity, clustering of source facts to minimize redundancy, and naive grouping of calculated hierarchies to combine similar structures.

Azgomi & Sohrabi (2021) introduced a new map-reduce-based technique for building multiple view processing plans (MVPPs) in large data applications. The solution uses the *MapReduce* programming method to optimize the join operator, which is a critical relational operator in terms of time complexity for building MVPP and answering queries. The emphasis on large data applications involves considering all join operations as set similarity joins and leveraging the parallel processing capabilities of the MapReduce framework to enhance efficiency. The MR-MVPP technique is consistent with previous research on employing parallel and distributed systems to optimize query processing. This technique, which combines MapReduc*e* with hashing and cost-reduction algorithms, expands on current frameworks to solve scalability and performance concerns in big data contexts.

Martin and Davis (2021) investigated the challenges and solutions in logical data warehouse (LDW) architecture, focusing on large-scale sectors such as healthcare. Existing research has highlighted the inefficiency of traditional physical data warehouses (PDWs) in managing sparse, hyperdimensional data, columnar database architecture, and query optimization. This study proposes unique techniques to improve data clustering and storage tier assignment for better query performance in LDWs. The study proposes multi-tiered storage methods that make use of the columnar design of SAP HANA. Moreover, the study introduces new algorithms to improve workload preprocessing, cluster identification for OLAP queries, and cluster assignment to tiered storage (hot, warm, and cold tiers). These advancements enable LDW to improve query workload coverage with fast storage devices, thus enhancing efficiency.

Ngo et al. (2019) addressed the development of DW linked to agricultural big data, emphasizing the limitations of existing approaches in handling its volume, variety, velocity, and veracity. Researchers had explored agricultural data integration and analytics using ML or simple DW models. However, these techniques often lacked robust schema designs and performance optimization. For example, star schema models were inadequate for complex agricultural datasets; moreover, solutions based on relational databases did not meet high-performance demands. This study built on these efforts by combining Hive, MongoDB, and Cassandra, leveraging their advanced features such as scalability, schema flexibility, and real-time updates, thus designing a constellation schema for effectively managing the particularities of of agricultural data.

Santos et al. (2017) proposed a methodology designed to transform multidimensional data models into Hive tables, offering a structured approach to organizing data at various levels of granularity. This approach allows for different types of queries, therefore providing flexibility for users to retrieve both high-level summaries and detailed data, depending on the analytical needs. The rules facilitate an efficient mapping from the conceptual model of the data to a practical implementation in Hive, ensuring that both performance and scalability are optimized.

Di Tria et al. (2014) introduced a graph-based multidimensional modeling approach, which enables automation of the design process through logical programming. This approach ensures scalability and agility by supporting structured and unstructured data integration. Traditional data warehouse methodologies, including data-oriented and requirement-oriented approaches, struggle with the complexities of big data environments. Data-oriented methods focus on structured data reengineering, often excluding unstructured sources, while requirement-oriented methods prioritize user needs but risk neglecting data availability during ETL processes. Hybrid methodologies address these limitations by integrating user requirements and reconciling diverse data sources. Thus, the proposed methodology allows for fast evolution, aligning with the requirements of big data warehouses in order to perform advanced analytical capabilities.

In contrast to traditional data warehouses and decision support systems (DSS), which primarily focus on the extraction and management of structured data, Nemati et al. (2002) proposed a comprehensive and integrative architectural model called the knowledge warehouse (KW). This architecture extends the capabilities of existing data warehouses by incorporating mechanisms for capturing, cleansing, storing, organizing, leveraging, and disseminating not only data and information but also firm knowledge. This approach combines principles derived from knowledge management, AI, and DSS to enable a more intelligent and adaptive decision making environment. The main contribution of this paper lies in redefining the purpose of DSS, from supporting decisions based on data to actively enhancing knowledge, learning, and the mental models of decision makers. The proposed architecture includes knowledge acquisition, transformation, storage, analysis, and communication modules. Additionally, it runs on an object-oriented framework that facilitates flexibility and scalability. To facilitate knowledge discovery and internalization, the study also describes how AI-driven analytical techniques, such as deep explanatory arguments and inductive model analysis, can be integrated into KW. With this conceptual change, KW is now positioned as the cornerstone of next-generation enterprise systems that support ongoing organizational learning and knowledge evolution in addition to managing information. Thus, by bridging the gap between knowledge-centric organizational intelligence systems and data-centric DSS, the work significantly advances the field.

In medical practice, physiological indicators are crucial because they serve as a foundation for expressing the physiological health status of the human body. One of the major areas of research in recent years has also been association rules. Liu (2024) proposed an AI, with association rule data mining system for biochemical markers of sports training as the goal of this project. In order to create networks and train systems, this study employed Markov logic. It also examined whether the training system might be linked to the Markov logic network. On the basis of the results, it is possible to develop biochemical indicators for sports training using a Markov logic network, and the

system has universal, directing, and constructive significance. The accuracy and recall rate obtained are approximately 90%.

The increasing volume and complexity of data in e-commerce have driven the application of data mining and classification algorithms to improve user analysis and targeted marketing. While previous studies have applied various data mining and classification methods to e-commerce platforms, many struggle with low accuracy and inefficiency when dealing with massive, high-dimensional datasets. Yang & Qi (2024) proposed an e-commerce data processing model based on data mining and an enhanced K-Nearest Neighbor (KNN) classification algorithm to address the current issue of low accuracy and time-consuming data mining and classification techniques used in e-commerce platforms. In order to thoroughly mine the enormous amount of e-commerce data, the model initially combines the dimensional control mechanism with the Spark mechanism. The extracted data is then classified using the KNN method, which uses a dynamic K-value selection strategy to the resolve misclassification issues common in traditional KNN. The findings suggest that the proposed approach improves the accuracy of e-commerce data classification, supports precision marketing, and provides new ideas for the strategic transformation of e-commerce platforms.

Guo et al. (2023) discussed the development of an intelligent manufacturing management system using data mining and AI to address the limitations of traditional production management modes and improve enterprise development in general. The system aims to improve the accuracy and timeliness of production site conditions, provide practical production plans and instructions, and optimize resource utilization. The approach involves a full system that comprises five sub-functional modules: order management, material management, mixed model assembly line balance, assembly line logistics scheduling, and system management. The system improves manufacturing processes by utilizing a number of essential features. One important function is assembly line balancing, which determines the best assembly scheme using complex algorithms. The results of this balancing process are then used in logistics scheduling, which focuses on route and operating duration of the automated guided vehicle (AGV) in order to maximize the logistics efficiency of the system. Additionally, data handling makes sure that different records, such as order information, material information, workstation status, and product priority, are used to efficiently manage both static and dynamic data. The functionality of the system, including such aspects as logistics scheduling and assembly line balancing, depends on this data management. The operational findings indicate that the total energy consumption of 10,000 yuan industrial production value is 401.19 kg of standard coal/10,000 yuan, which represents a 6.96%annual decline. Thus, the study promises many benefits for the best possible management of the manufacturing industry.

MRE – KDD⁺: ANATOMY AND FUNDAMENTAL MODELS

MRE – KDD⁺ is the innovative model underlying the framework we propose, and it has been designed to efficiently support complex knowledge pattern discovery from big data warehouses according to a multi-resolution, ensemble-based approach. Basically, MRE – KDD⁺ follows the cubing-then-mining approach (Han, 1997), which, as highlighted in the "OLAM: Combining OLAP and Data Mining" section, is the most promising OLAM solution for real-life applications. In this section, we indicate the definition and main properties of MRE – KDD⁺.

MRE–KDD⁺ OLAP-BASED DATA REPRESENTATION AND MANAGEMENT LAYER

Let $\mathcal{S} = \{S_0, S_1, \ldots, S_{K-1}\}$ be a set of K distributed and heterogeneous big data sources, and $\mathcal{D} = \{D_0, D_1, \ldots, D_{P-1}\}$ be a set of P (big) data marts defined over data sources in $\mathcal{S}$. The first component of MRE – KDD⁺ is the so-called multidimensional mapping function (MMF), defined as a tuple $MMF = \langle MMF^{\mathcal{M}}, MMF^{\mathcal{F}} \rangle$, which takes as input a subset of M data sources in $\mathcal{S}$, denoted by $\mathcal{S}^{\mathcal{M}} = \{S_m, S_{m+1}, \ldots, S_{m+M-1}\}$, and returns as output a data mart D_k in $\mathcal{D}$,

computed over data sources in $\mathcal{S}^{\mathcal{M}}$ according to the construct $MMF^{\mathcal{K}}$ that models the *definition* of D_k. $MMF^{\mathcal{K}}$ is in turn implemented as a conventional OLAP conceptual schema, such as *star* or *snowflake schemas* (Colliat, 1996; Han & Kamber, 2000). $MMF^{\mathcal{F}}$ is the construct of MMF that properly models the underlying function, defined as shown in Equation 1:

$$MMF^{\mathcal{F}} : \mathcal{S}\ \mathcal{D} \tag{1}$$

Given a MMF G, we introduce the concept of *degree* of G, denoted by G^{Δ}, which is defined as the number of data sources in $\mathcal{S}$ over which the data mart provided by G (i.e., D_k) is computed, i.e. $G^{\Delta} \equiv |\mathcal{S}^{\mathcal{M}}|$.

Owing to the strongly data-centric nature of MRE – KDD^{+}, management of OLAP data assumes a critical role; this is also the case with respect to performance issues, which must be taken into relevant consideration in big data applications like those addressed by OLAM. To this end, we introduce *the* multidimensional cubing function (MCF), defined as a tuple $MCF = \langle MCF^{\mathcal{K}}, MCF^{\mathcal{F}} \rangle$, which takes as input a data mart D_k in $\mathcal{D}$, and returns as output a data mart D_h in $\mathcal{D}$, according to the construct $MCF^{\mathcal{K}}$ that models an OLAP operator/tool. In more detail, $MCF^{\mathcal{K}}$ can be one of the following OLAP operators/tools:

- multidimensional view extraction $\mathcal{V}$, which computes $\boldsymbol{D}_h$ as a multidimensional view extracted from $\boldsymbol{D}_k$ by means of a set of ranges $\boldsymbol{R}_0, \boldsymbol{R}_1, \ldots, \boldsymbol{R}_{N-1}$ defined on the N dimensions of $\boldsymbol{D}_k$, $\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{N-1}$, respectively, being each range $\boldsymbol{R}_j$ defined as a tuple $\boldsymbol{R}_j = \langle \boldsymbol{L}_l, \boldsymbol{L}_U \rangle$, with $\boldsymbol{L}_l < \boldsymbol{L}_U$, such that $\boldsymbol{L}_l$ is the lower and $\boldsymbol{L}_U$ is the upper bound on $\boldsymbol{d}_j$, respectively;
- range aggregate que*ry* $\mathcal{Q}$, which computes $\boldsymbol{D}_h$ as a one-dimensional view with cardinality equal to 1 (i.e., an *aggregate value*) given by the application of a SQL aggregate operator (such as SUM, COUNT, AVG, etc.) applied to the collection of (OLAP) cells contained within a multidimensional view extracted from $\boldsymbol{D}_k$ by means of the operator $\mathcal{V}$;
- top-k query $\mathcal{K}$, which computes $\boldsymbol{D}_h$ as a multidimensional view extracted from $\boldsymbol{D}_k$ by means of the operator $\mathcal{V}$, and containing the (OLAP) cells of $\boldsymbol{D}_k$ whose values are the first $\mathcal{K}$ greatest values among cells in $\boldsymbol{D}_k$;
- drill-down $\mathcal{U}$, which computes $\boldsymbol{D}_h$ via decreasing the level of detail of data in $\boldsymbol{D}_k$;
- roll-up $\mathcal{R}$, which computes $\boldsymbol{D}_h$ via increasing the level of detail of data in $\boldsymbol{D}_k$;
- pivo*t* $\mathcal{P}$, which computes $\boldsymbol{D}_h$ via re-structuring the dimensions of $\boldsymbol{D}_k$ (e.g., changing the ordering of dimensions).

Formally, $MCF^{\mathcal{K}} = \{\mathcal{V}, \mathcal{Q}, \mathcal{K}, \mathcal{U}, \mathcal{R}, \mathcal{P}\}$. Finally, $MCF^{\mathcal{F}}$ is the construct of MCF that properly models the underlying function, defined as shown in Equation 2:

$$MCF^{\mathcal{F}} : \mathcal{D}\ \mathcal{D} \tag{2}$$

It should be noted that the construct $MCF^{\mathcal{K}}$ of MCF operates on a singleton data mart to extract another data mart. In order to improve the quality of the overall KDD process, we also introduce the extended multidimensional cubing function (MCF$_E$), defined as a tuple $MCF_E = \langle MCF_E^{\mathcal{K}}, MCF^{\mathcal{F}} \rangle$, which extends MCF by providing a different, complex OLAP operator/tool (i.e., $MCF_E^{\mathcal{K}}$) instead of the "basic" $MCF^{\mathcal{K}}$. $MCF_E^{\mathcal{K}}$ supports the amenity of executing $MCF^{\mathcal{K}}$ over multiple data marts, modeled as a subset of B data marts in $\mathcal{D}$, denoted by $\mathcal{D}^{\mathcal{B}} = \{\mathcal{D}_b, \mathcal{D}_{b+1}, \ldots, \mathcal{D}_{b+B-1}\}$, being these data marts combined by means of the operator JOIN performed with respect to schemas of data marts. Specifically, $MCF_E^{\mathcal{K}}$ operates according to two variants: In the first one, we first apply an instance of $MCF^{\mathcal{K}}$ to each data mart in $\mathcal{D}^{\mathcal{B}}$, thus obtaining a set of *transformed* data marts $\mathcal{D}^{\mathcal{T}}$, and then the

operator JOIN to data marts in $\mathscr{D}^{\mathscr{T}}$. In the second one, we first apply the operator JOIN to data marts in $\mathscr{D}^{\mathscr{B}}$, thus obtaining a *unique* data mart $\mathscr{D}^{\mathscr{U}}$, and then an instance of $MCF^{\mathscr{K}}$ to the data mart $\mathscr{D}^{\mathscr{U}}$.

For example, let $\mathscr{D}^{\mathscr{B}} = \{\mathscr{D}_0, \mathscr{D}_1, \mathscr{D}_2\}$ be the target subset of data marts; then, according to the first variant, a possible instance of $MCF_E^{\mathscr{K}}$ could be: $\mathscr{V}(\mathscr{D}_0) \rhd \lhd \mathscr{K}(\mathscr{D}_1) \rhd \lhd \mathscr{U}(\mathscr{D}_2)$; contrary to this, according to the second variant, a possible instance of $MCF_E^{\mathscr{K}}$ could be: $\mathscr{U}(\mathscr{D}_0 \rhd \lhd \mathscr{D}_1 \rhd \lhd \mathscr{D}_2)$. Note that, in both cases, the result of the operation is still a data mart belonging to the set of data marts $\mathscr{D}$ of $\mathscr{MRE} - \mathscr{KDD}^+$.

Formally, we model $MCF_E^{\mathscr{K}}$ as a tuple $MCF_E^{\mathscr{K}} = \langle \mathscr{D}^{\mathscr{B}}, \mathscr{Y} \rangle$, such that (*i*) $\mathscr{D}^{\mathscr{B}}$ is the subset of data marts in $\mathscr{D}$ on which $MCF_E^{\mathscr{K}}$ operates to extract the final data mart, and (*ii*) $\mathscr{Y}$ is the set of instances of $MCF^{\mathscr{K}}$ used to accomplish this goal. Specifically, Figure 2 shows the MCF_E Algorithm, which is designed to create a final data mart $\mathscr{D}$ from a collection of heterogeneous and diverse data sources. The first step of the algorithm is to choose a relevant subset of various sources; this is according to specific analytical requirements. Each selected data source has a preliminary data mart built and set up with the proper structure. A join of these data marts is then named $\mathscr{D}^{\mathscr{B}}$. Depending on the user condition, the algorithm follows one of two paths. If the condition is met, each data mart in $\mathscr{D}^{\mathscr{B}}$ is further processed through the MCF function. As shown in Figure 3, the MCF function applies a series of analytical and structural transformations such as extracting views, aggregating queries, performing top-K queries, and executing OLAP operations like drill-down, roll-up, and pivot. The results are integrated into a transformed data mart. Each of these transformed data marts is collected and subsequently joined to form the final output of MCF. Subsequently, these data marts are then joined together to produce the final data mart $\mathscr{D}^{\mathscr{T}}$. Otherwise, if the user condition is not met, the algorithm directly joins all base data marts in $\mathscr{D}^{\mathscr{B}}$ into an intermediate data mart $\mathscr{D}^{\mathscr{U}}$, which is then passed through the MCF function to produce the final data mart. Ultimately, the result is a unified transformed data mart tailored to the needs of the user.

Figure 2. MCF_E Algorithm

Algorithm 1 $\mathrm{MCF_E}$ Algorithm

Input: Set of Distributed and Heterogeneous Data Sources $\mathcal{S}$
Output: Data Mart D

Begin
Data Source s;
Set of Data Marts $\mathcal{D}_{\mathcal{B}}$, $\mathcal{D}_{\mathcal{T}}$;
Data Mart D_h, D_k, D_U;
$\mathcal{S}_f \leftarrow createSubSet(\mathcal{S})$;
for (s *in* $\mathcal{S}_f$) **do**
  $D_k \leftarrow createDataMart(s)$;
  $D_k \leftarrow confSchema(D_k, schema)$;
  $\mathcal{D}_{\mathcal{B}}.add(D_k)$;
end for
if ($UserCondition$) **then**
  for (d *in* $\mathcal{D}_{\mathcal{B}}$) **do**
    $D_h \leftarrow MCF(d)$;
    $\mathcal{D}_{\mathcal{T}}.add(D_h)$;
  end for
  $D \leftarrow join(\mathcal{D}_{\mathcal{T}})$;
else
  $D_U \leftarrow join(\mathcal{D}_{\mathcal{B}})$
  $D \leftarrow MCF(D_U)$;
end if
return D;
End

Figure 3. MCF Algorithm

Algorithm 2 MCF Algorithm
Input: Data Mart D_k
Output: Transformed Data Mart D_h

Begin
$D_{h1} \leftarrow extractView(D_k)$;
$D_{h2} \leftarrow aggregateQuery(D_k)$;
$D_{h3} \leftarrow topKQuery(D_k)$;
$D_{h4} \leftarrow drillDown(D_k)$;
$D_{h5} \leftarrow rollUp(D_k)$;
$D_{h6} \leftarrow pivot(D_k)$;
$D_h \leftarrow transDataMart(D_{h1}, D_{h2}, D_{h3}, D_{h4}, D_{h5}, D_{h6})$;
return D_h;
End

As depicted in Figure 2, the MCF_E Algorithm uses the MCF Algorithm (shown in Figure 3), which is a fundamental procedure designed to process data marts by applying OLAP operations previously described. By extending MCF, the MCF_E Algorithm enhances the KDD process by supporting complex operations over multiple data marts, incorporating *JOIN* operations in its two operational variants to derive the final transformed data marts.

MRE – KDD⁺ Data Mining Layer

DM algorithms defined in MRE – KDD⁺ are modeled by the set $\mathscr{A} = \{\mathscr{A}_0, \mathscr{A}_1, \ldots, \mathscr{A}_{T-1}\}$. These are classical DM algorithms focused on covering specific instances of consolidated KDD tasks, such as the discovery of patterns and regularities, discovery of association rules, classification, clustering, etc., with the novelty of being applied to multidimensional views (or, equally, data marts) extracted from the data mart domain $\mathscr{D}$ of $\mathscr{MRE} - \mathscr{KDD}^+$ via complex OLAP operators/tools implemented by the components MCF and MCF_E. Formally, an algorithm $\mathscr{A}_h$ of $\mathscr{A}$ in MRE – KDD⁺ is modeled as a tuple $\mathscr{A}_h = \langle \mathscr{I}_h, \mathscr{D}_h, \mathscr{O}_h \rangle$, such that: (*i*) $\mathscr{I}_h$ is the instance of $\mathscr{A}_h$ (properly, $\mathscr{A}_h$ models the *class* of the particular DM algorithm); (*ii*) $\mathscr{D}_h$ is the data mart on which $\mathscr{A}_h$ is executed to extract knowledge; and (*iii*) $\mathscr{O}_h$ is the output knowledge of $\mathscr{A}_h$. Specifically, $\mathscr{O}_h$ representation depends on the nature of algorithm $\mathscr{A}_h$, meaning that if, for instance, $\mathscr{A}_h$ is a clustering algorithm, then $\mathscr{O}_h$ is represented as a collection of clusters (reasonably, modeled as sets of items) extracted from $\mathscr{D}_h$.

KDD process in MRE – KDD⁺ are governed by the component execution scheme (ES), which rigorously models *how* algorithms in $\mathscr{A}$ must be executed over multidimensional views of $\mathscr{D}$. To this end, *ES* establishes (*i*) how to combine multidimensional views and DM algorithms (i.e., which algorithm must be executed on which view); and (*ii*) the temporal sequence of executions of DM algorithms over multidimensional views. To formally model this aspect of the framework, we introduce the knowledge discovery function (KDF), which takes as input a collection of *R* algorithms $\mathscr{A}^R = \{\mathscr{A}_r, \mathscr{A}_{r+1}, \ldots, \mathscr{A}_{r+R-1}\}$ and a collection of *W* data marts $\mathscr{D}^w = \{\mathscr{D}_w, \mathscr{D}_{w+1}, \ldots, \mathscr{D}_{w+W-1}\}$, and returns as output an execution scheme ES_p. KDF is defined as shown in Equation 3:

$$KDF : \mathscr{A}^{\mathscr{R}} \times \mathscr{D}^{\mathscr{W}} \langle \mathscr{I}^{\mathscr{R}} \times \mathscr{D}^{\mathscr{T}}, \varphi \rangle \tag{3}$$

such that: (i) $\mathscr{I}^{\mathscr{R}}$ is a collection of instances of algorithms in $\mathscr{A}^{\mathscr{R}}$, (ii) $\mathscr{D}^{\mathscr{T}}$ is a collection of transformed data marts obtained from $\mathscr{D}^{\mathscr{W}}$ by means of cubing operations provided by the components MCF or MCF_E of the framework, and (iii) φ is a collection determining the temporal sequence of

instances of algorithms in $\mathcal{I}^{\mathcal{R}}$ over data marts in $\mathcal{D}^{\mathcal{T}}$ in terms of ordered pairs $\langle \mathcal{I}_r, \mathcal{D}_k^{\mathcal{T}} \rangle$, such that the ordering of pairs indicates the temporal ordering of executions. From Equation 3, we derive the formal definition of the component *ES* of MRE – KDD$^+$

as shown in Equation 4:

$$ES = \langle \mathcal{I} \times \mathcal{D}, \varphi \rangle \tag{4}$$

Finally, the execution scheme ES_p provided by KDF can be one of the following alternatives:

- singleton execution $\langle \boldsymbol{\mathcal{I}}_r \times \boldsymbol{\mathcal{D}}_k^{\mathcal{T}}, \boldsymbol{\varphi} \rangle$: execution of the instance $\boldsymbol{\mathcal{I}}_r$ of the algorithm $\boldsymbol{\mathcal{A}}_r$ over the transformed data mart $\boldsymbol{\mathcal{D}}_k^{\mathcal{T}}$, with $\boldsymbol{\varphi} = \left\{ \left\langle \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{D}}_k^{\mathcal{T}} \right\rangle \right\}$;
- $1 \times \boldsymbol{N}$ multiple execution $\langle \boldsymbol{\mathcal{I}}_r \times \left\{ \boldsymbol{\mathcal{D}}_k^{\mathcal{T}}, \boldsymbol{\mathcal{D}}_{k+1}^{\mathcal{T}}, \ldots, \boldsymbol{\mathcal{D}}_{k+N-1}^{\mathcal{T}} \right\}, \boldsymbol{\varphi} \rangle$: execution of the instance $\boldsymbol{\mathcal{I}}_r$ of the algorithm $\boldsymbol{\mathcal{A}}_r$ over the collection of transformed data marts $\left\{ \boldsymbol{\mathcal{D}}_k^{\mathcal{T}}, \boldsymbol{\mathcal{D}}_{k+1}^{\mathcal{T}}, \ldots, \boldsymbol{\mathcal{D}}_{k+N-1}^{\mathcal{T}} \right\}$, with $\boldsymbol{\varphi} = \left\{ \left\langle \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{D}}_k^{\mathcal{T}} \right\rangle, \left\langle \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{D}}_{k+1}^{\mathcal{T}} \right\rangle, \ldots, \left\langle \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{D}}_{k+N-1}^{\mathcal{T}} \right\rangle \right\}$;
- $\boldsymbol{N} \times 1$ multiple execution $\langle \left\{ \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{I}}_{r+1}, \ldots, \boldsymbol{\mathcal{I}}_{r+N-1} \right\} \times \boldsymbol{\mathcal{D}}_k^{\mathcal{T}}, \boldsymbol{\varphi} \rangle$: execution of the collection of instances $\left\{ \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{I}}_{r+1}, \ldots, \boldsymbol{\mathcal{I}}_{r+N-1} \right\}$ of the algorithms $\left\{ \boldsymbol{\mathcal{A}}_r, \boldsymbol{\mathcal{A}}_{r+1}, \ldots, \boldsymbol{\mathcal{A}}_{r+N-1} \right\}$ over the transformed data mart $\boldsymbol{\mathcal{D}}_k^{\mathcal{T}}$, with $\boldsymbol{\varphi} = \left\{ \left\langle \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{D}}_k^{\mathcal{T}} \right\rangle, \left\langle \boldsymbol{\mathcal{I}}_{r+1}, \boldsymbol{\mathcal{D}}_k^{\mathcal{T}} \right\rangle, \ldots, \left\langle \boldsymbol{\mathcal{I}}_{r+N-1}, \boldsymbol{\mathcal{D}}_k^{\mathcal{T}} \right\rangle \right\}$;
- $\boldsymbol{N} \times \boldsymbol{M}$ multiple execution $\langle \left\{ \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{I}}_{r+1}, \ldots, \boldsymbol{\mathcal{I}}_{r+N-1} \right\} \times \left\{ \boldsymbol{\mathcal{D}}_k^{\mathcal{T}}, \boldsymbol{\mathcal{D}}_{k+1}^{\mathcal{T}}, \ldots, \boldsymbol{\mathcal{D}}_{k+M-1}^{\mathcal{T}} \right\}, \boldsymbol{\varphi} \rangle$: execution of the collection of instances $\left\{ \boldsymbol{\mathcal{I}}_r, \boldsymbol{\mathcal{I}}_{r+1}, \ldots, \boldsymbol{\mathcal{I}}_{r+N-1} \right\}$ of the algorithms $\left\{ \boldsymbol{\mathcal{A}}_r, \boldsymbol{\mathcal{A}}_{r+1}, \ldots, \boldsymbol{\mathcal{A}}_{r+N-1} \right\}$ over the collection of transformed data marts $\left\{ \boldsymbol{\mathcal{D}}_k^{\mathcal{T}}, \boldsymbol{\mathcal{D}}_{k+1}^{\mathcal{T}}, \ldots, \boldsymbol{\mathcal{D}}_{k+N-1}^{\mathcal{T}} \right\}$, with $\boldsymbol{\varphi} = \left\{ \ldots, \left\langle \boldsymbol{\mathcal{I}}_{r+p}, \boldsymbol{\mathcal{D}}_{k+q}^{\mathcal{T}} \right\rangle, \ldots \right\}$, such that $0 \leq \boldsymbol{p} \leq \boldsymbol{N} - 1$ and $0 \leq \boldsymbol{q} \leq \boldsymbol{M} - 1$.

Figure 4 depicts the KDF algorithm and its role inside the MRE – KDD$^+$ architecture. KDF algorithm serves as the control center for analytical execution and manages the execution scheme *ES* by enabling flexible, modular, and scalable deployment of data mining techniques across a heterogeneous data environment. It ensures that each algorithm is applied systematically and that the resulting knowledge is aggregated in a structured and meaningful way. As shown in Figure 4, KDF iterates through each data mining algorithm in the set $\mathcal{A}$, and for each, it loops over all data marts in the set $\mathcal{D}$. Then, for every pairing, the selected data mart D_h is first transformed using the MCF function, after which the algorithm A_h from $\mathcal{A}$ is executed on the transformed data. The output O_h is collected, and the triple (A_h, D_h, O_h) is stored in the execution scheme *ES*. Thereafter, the execution scheme *ES* is passed to the function sequence execution, which organizes these tasks into a logically ordered flow ($ES_{\mathcal{O}}$). This flow is then executed step by step, with each result being collected into the final output set $\mathcal{O}$.

Figure 4. KDF Algorithm

Algorithm 3 KDF Algorithm

Input: Set of DM Algorithms $\mathcal{A}$, Set of Data Mart $\mathcal{D}$
Output: Collection of Outputs $\mathcal{O}$

```
Begin
  Data Mart D_h, V;
  DM Algorithm A_h;
  ES ← [ ];
  ES_O ← [ ];
  for (A_h in A) do
    for (D_h in D) do
      V ← MCF(D_h);
      O_h ← A_h.run(V);
      ES.add((A_h, D_h, O_h));
    end for
  end for
  ES_O ← sequenceExecution(ES);
  for (es in ES_O) do
    O_s ← execute(es);
    O.add(O_s);
  end for
  return O;
End
```

MRE – KDD⁺ Ensemble Layer

As stated in Section 1, at the output layer, MRE – KDD⁺ adopts an ensemble-based approach. The so-called mining results (MR) coming from the executions of DM algorithms over collections of data marts must be finally merged in order to provide the end-user/application with the extracted knowledge that is presented in the form of complex patterns. It should be noted that this is a relevant task in our proposed framework, as very often end-users/applications are interested in extracting useful knowledge by means of *correlated*, *cross-comparative* KDD tasks, rather than a singleton KDD task, according to real-life DM scenarios. Combining results coming from different DM algorithms is a non-trivial research issue, as recognized in the literature. In fact, as highlighted in the previous subsection, the output of a DM algorithm depends on the nature of that algorithm, so that in some cases, MR coming from very different algorithms cannot be combined directly.

In MRE – KDD⁺, we face-off this problematic issue by making use of OLAP technology again. We build multidimensional views over MR provided by execution schemes of KDF, thus giving support to a unifying manner of exploring and analyzing final results. It should be noted that this approach is well motivated in view of the fact that usually end-user/applications are interested in analyzing final results on the basis of a certain mining metrics provided by KDD processes (e.g., confidence interval of association rules, density of clusters, recall of IR-style tasks, etc.), and this technique is perfectly suitable to be implemented within OLAP data cubes where (*i*) output of DM algorithms (e.g., item sets) is the data source, (*ii*) user-selected features of the output of DM algorithms are the (OLAP) dimensions, and (*iii*) the above-mentioned mining metrics are the (OLAP) measures. Furthermore, this approach also provides the benefit of efficiently supporting the *visualization* of final results by means of attracting user-friendly, graphical formats/tools such as multidimensional bars, charts, plots, etc., similarly to the functionalities supported by DBMiner and WEKA.

The multidimensional ensembling function (MEF) is the component of MRE – KDD⁺ that is responsible for supporting the above-described knowledge presentation/delivery task. It takes as input a collection of $\mathcal{Q}$ output results $\mathcal{O} = \{\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_{\mathcal{Q}-1}\}$ provided by KDF-formatted execution schemes and the definition of a data mart Z, and returns as output a data mart $\mathcal{Z}$, which we name as

knowledge visualization data mart (KVDM), built over data in $\mathcal{O}$ according to $\mathcal{X}$. Formally, MEF is defined as shown in Equation 5:

$$MEF : \langle \mathcal{O}, \mathcal{X} \rangle \ \mathcal{L} \tag{5}$$

It should be noted that the KVDM $\mathcal{L}$ becomes part of the set of data marts $\mathcal{D}$ of $\mathcal{MRE} - \mathcal{KD}$ $\mathcal{D}^+$, but, contrary to the previous data marts, which are used for knowledge processing purposes, it is used for knowledge exploration/visualization purposes. Figure 5 presents the MEF algorithm, which is responsible for transforming the outputs of the knowledge discovery phase (produced by KDF) into a final data mart prepared for visualization. It iterates through each output object from KDF and generates an OLAP-based data view using user-selected features and mining metrics. These individual views are collected and then combined into a comprehensive view, from which the final visualization-oriented data mart is generated and returned.

Figure 5. MEF Algorithm

Algorithm 4 `MEF Algorithm`

Input: Collection of KDF Outputs $\mathcal{O}$, Data Mart Z, User Selected Feature F, Mining Metrics M
Output: Data Mart L

Begin
Data View V_s, V_c;
Set of Data View $\mathcal{V}$;
KDF Output O_s;
for $(O_s \ in \ \mathcal{O})$ **do**
$V_s \leftarrow buildOLAPView(O_s, F, M)$;
$\mathcal{V}.add(V_s)$;
end for
$V_C \leftarrow combinedViews(\mathcal{V})$;
$L \leftarrow generateVisulizationDataMart(V_C)$;
return L;
End

A REFERENCE ARCHITECTURE FOR SUPPORTING OLAM-BASED BIG DATA ANALYTICS OVER BIG DATA WAREHOUSES

Figure 6 shows the reference architecture implementing the framework we propose. This architecture is suitable for implementation on top of any distributed software platform, like Clouds, under the design guidelines given by component-oriented software engineering best practices. Despite being orthogonal to any distributed big data environment, as stated in the introductory section, this architecture is particularly useful in a general application scenario populated by distributed and heterogeneous big data sources, and in the integration/data layer of cooperative information systems. As we will demonstrate throughout the remaining part of this Section, components of the reference architecture implement constructs of the underlying model MRE – KDD$^+$, according to a meaningful abstraction between formal constructs and software components.

As shown in Figure 6, in the proposed architecture, distributed and heterogeneous data sources, located at the data source layer, are first processed by means of ETL tasks implemented by the *ETL* engine and then integrated into a common relational data layer, in order to ensure flexibility at the next data processing/transformation steps, and take advantage of mining correlated knowledge. The data mart builder, which implements the component MMF of MRE – KDD$^+$ is responsible for

constructing a collection of subject-oriented data marts, which populate *the data mart layer*, via accessing data at the relational data layer, and according to specific requirements of the target big data application running on top of the proposed architecture. The OLAP engine, which implements the components MCF and MCF_E of MRE – KDD$^+$, provides conventional and complex OLAP operators/tools over data marts of the data mart layer, thus originating a collection of multidimensional views located at the OLAP view layer. These views constitute the input of the OLAM engine, which, by accessing a set of conventional DM algorithms stored in the DM algorithm repository, implements the component KDF of MRE – KDD$^+$ via combining views and algorithms to execute even complex KDD processes. Finally, the mining result merging component, which implements the component MEF of MRE – KDD$^+$, combining different MR to obtain the final knowledge, and meaningfully supports the knowledge fruition experience via complex patterns such as multidimensional domains, hierarchical structures, and clusters.

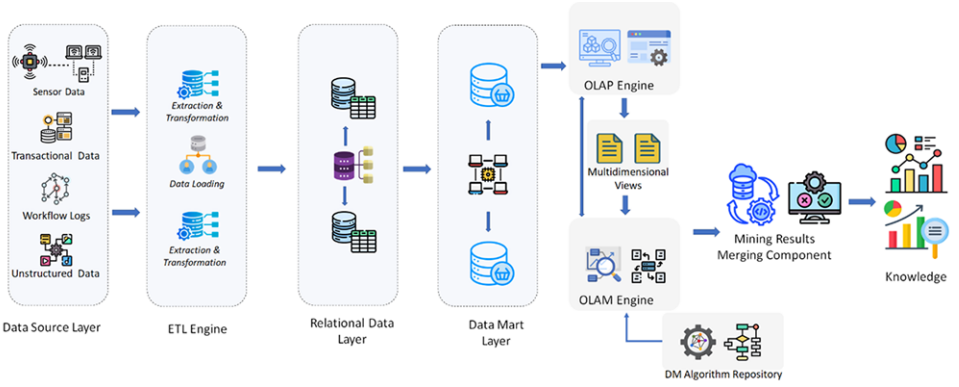
Another integrated component of the proposed architecture is the MRE – KDD$^+$–based visual tool KBMiner, which is not depicted in Figure 6 for the sake of simplicity. KBMiner can connect to the architecture in order to efficiently support the editing of KDT (see the introductory section) for discovering useful knowledge from big data warehouses. However, being completely independent of the particular implementation, the proposed architecture can also be realized as a core-, inside-, stand-alone-platform within distributed big data environments, where KDD functionalities are available in the vest of component-oriented API to intelligent applications executing KDD processes rather than in the vest of plug-in components of KBMiner to knowledge workers authoring KDD processes (similarly to the former one, in the latter case the edited KDT must ultimately be executed by an ad-hoc software component implementing a "general-purpose" MRE – KDD$^+$ engine).

Despite OLAP technology's having already reached relevant performance, however, modern data-driven applications that rely on OLAP systems or that embed an OLAP engine inside their core layer to enable rapid, multidimensional analysis of large datasets are facing a problematic bottleneck, owing to the fact that accessing and processing OLAP data surfing from performance issues. With exponential data growth and real-time analytical demands, maintaining optimal performance in OLAP workflows causes critical challenges. While TB/PB is the typical data magnitude achieved for legacy applications, it becomes explosive in the specific context of big data applications (see Warnke et al., 2024).

A very efficient solution to this problem is represented by data-cube-compression/approximate-query-answering techniques, which allow us to sensitively speed up data access activities and query evaluation tasks against OLAP data by (*i*) reducing the size of data cubes, thus obtaining *compressed representations* of them, and (*ii*) implementing efficient algorithms capable of evaluating queries over these representations, thus obtaining approximate *answers* that are perfectly tolerable in OLAP (Cuzzocrea, 2023).

In an OLAM architecture such as that proposed in Figure 6, which is *intrinsically* OLAP-based, these techniques can be easily integrated (specifically, with regard to our proposed architecture, inside the OLAP Engine) and successfully exploited in order to improve the overall performance of even-complex KDD processes, since they allow us to reduce the complexity of resource-intensive operations (i.e., multidimensional data access and management). Among the plethora of data-cube-compression/approximate-query-answering techniques proposed in the literature, we recall *analytical synopses* (Cuzzocrea, 2025) and *sampling-based approaches* (e.g., (Cuzzocrea & Gunopulos, 2014)), which represent relevant results in this research field. From these traditional approaches, even recently modern big-data-oriented initiatives have focused the attention on this *ever-green* research topic (e.g., (Li *et al*., 2022; Qi *et al*., 2023)).

Figure 6. Reference Architecture Implementing the Proposed Multi-Resolution Ensemble-Based Model for Advanced Knowledge Discovery in Big Data Warehouses Framework



KBMINER: A BIG DATA VISUALIZATION TOOL LEVERAGING MRE – KDD+

It is widely recognized that knowledge discovery is *intrinsically a semi-automatic* process, meaning that it requires the interaction of the system author, who is usually also an expert of the investigated application scenario. Building on the core principles of knowledge discovery and the complexities introduced by big data environments, this section introduces KBMiner, a visual and modular tool designed to handle the presented principles. It demonstrates how the MRE – KDD+ framework can be effectively applied to support the intuitive design and execution of complex KDD processes. Including the following: (*i*) defining the goals of the target KDD process, (*ii*) defining the parameters of the target KDD process, (*iii*) setting the default/input values of such parameters, (*iv*) checking the alignment and the correctness of intermediate results generated by the execution of the target KDD process, (*v*) composing the final results, (*vi*) understanding and mine the final results. On the other hand, very often the system author is not an ICT expert; as an example, this is a common case in the business intelligence (BI) context, where OLAP/OLAM technology has been widely applied, and knowledge workers are typically non-ICT-expert business managers and administrators. As a consequence of both aspects, in real-life systems/applications, there is an urgent need for visual authoring tools able to efficiently support the editing of even complex KDD processes in a user-friendly manner. In other words, these tools must be capable of allowing system authors to design KDD processes in a simple, intuitive and interactive manner, by means of meaningful metaphors offered by visual programming. It should be noted that DBMiner and WEKA adhere to this evidence. It should also be noted that, from the classical settings of OLAP/OLAM and BI, this assumption and derived concepts have later evolved, with similar hyperboles, in the novel (and hereditary) big data context (see Liu et al., 2014).

KBMiner is a MRE – KDD+-based visual tool supporting the editing of KDT for the discovery of useful knowledge from big data warehouses according to the MRE – KDD+ guidelines. As highlighted in the introductory section, KDT allows the system author to "codify" KDD processes, being the underlying "programming language" based on the constructs of MRE – KDD+. The key elements of KBMiner (see the MRE – KDD+ UML Class Diagram section, below) show its capabilities in knowledge discovery in large datasets. It includes a range of core features required to support data preparation, exploration, transformation, and the design and execution of mining tasks within the system.

A KDT is a directed graph in which nodes represent algorithms/tasks, and arcs represent data flows or inter-algorithm/tasks operations. Furthermore, a KDT also includes the multidimensional-views/data-marts on which the previous algorithms/tasks execute, and other components modeling the

composition of MR according to the ensemble-based approach defined by MRE – KDD$^+$. Furthermore, in KBMiner, the system author is also allowed to associate the so-called mining rules (MR), which are logical rules defined on MRE – KDD$^+$ entities, to KDT arcs, in order to codify on top of the target KDT a sort of control algorithm that is in charge of *driving* the overall KDD process via controlling the values (TRUE or FALSE) given by the evaluation of MR across intermediate tasks of the process. As an example, given the domain of $\mathcal{MRE} - \mathcal{KDD}^+$ entities $\left\{\{\mathcal{A}_h, \mathcal{A}_k, \mathcal{A}_z\}, \{\mathcal{D}_m, \mathcal{D}_n, \mathcal{D}_p\}\right\}$, such that $\mathcal{A}_i$ with $i \in \{h,k,z\}$ is a clustering algorithm, and $\mathcal{D}_i$ with $i \in \{m,n,p\}$ is a data mart, a MR r, after the execution of $\mathcal{A}_h$ on $\mathcal{D}_m$ producing the output $\mathcal{O}_h$ in terms of a collection of clusters, could decide to run $\mathcal{A}_k$ on $\mathcal{D}_n$ or, alternatively, $\mathcal{A}_z$ on $\mathcal{D}_p$ on the basis of the fact that densities of clusters in $\mathcal{O}_h$ are greater than a given threshold V or not.

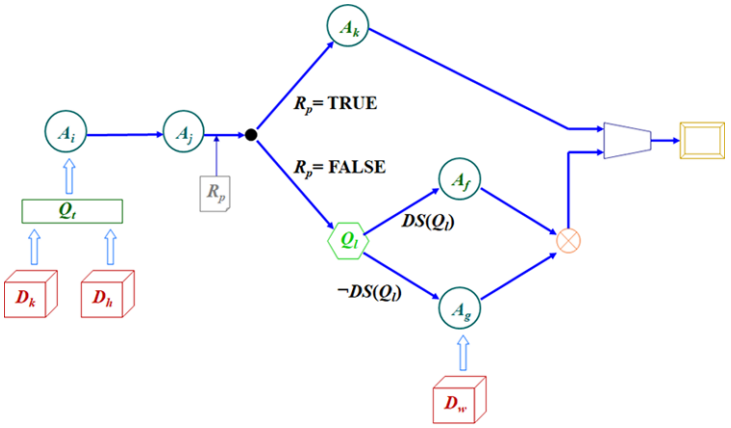
Other two relevant features supported by KBMiner are the following: (*i*) *interface operations* between DM algorithms, which establish how the output $\mathcal{O}_i$ of an algorithm $\mathcal{A}_i$ must be provided as input $\mathcal{D}_i$ to another algorithm $\mathcal{A}_j$ – the simplest way of implementing this operation is to directly transfer $\mathcal{O}_i$ to$\mathcal{D}_j$, but more complex models can be devised, such as generating $\mathcal{D}_j$ from $\mathcal{O}_i$ via a given OLAP operator/tool; (*ii*) merging operations between MR, which establish how MR must be combined at both the intermediate tasks of the target KDD process and the output layer in order to produce the final knowledge – some examples of operations used to this end are union, intersection, OLAP-query-based projection, etc.

Running Example

A meaningful example of KDT that can be edited in KBMiner is depicted in Figure 7. The KDD process modeled by the KDT of Figure 7 is composed of the following tasks:

1) DM algorithm $\mathcal{A}_i$ is executed over the multidimensional view originated by the OLAP query Q_t against the data marts $\mathcal{D}_k$ and $\mathcal{D}_h$, and it produces a multidimensional view representing the output $\mathcal{O}_i$;
2) DM algorithm $\mathcal{A}_j$ is executed over $\mathcal{O}_i$, and it produces the output $\mathcal{O}_j$;
3) MR R_p is evaluated against $\mathcal{O}_j$: if the value of R_p is TRUE, then DM algorithm $\mathcal{A}_k$ is executed over $\mathcal{O}_j$, thus producing the output $\mathcal{O}_k$; otherwise, if the value of R_p is FALSE, then $\mathcal{O}_j$ is partitioned into two multidimensional views by means of the OLAP query Q_l. The first view, given by the result of Q_l, denoted by $\neg DS(Q_l)$, constitutes the input of the DM algorithm $\mathcal{A}_f$, which produces the output $\mathcal{O}_f$. The second view, given by the complementary set of the results of Q_l, denoted by $\neg DS(Q_l)$, constitutes, along with the data mart $\mathcal{D}_w$, the input of DM algorithm $\mathcal{A}_g$, which produces the output $\mathcal{O}_g$;
4) The final result of the KDD process modeled by the KDT of the running example is obtained in dependence on the value of the MR R_p against $\mathcal{O}_j$: if that value is TRUE, then the final result corresponds to $\mathcal{O}_k$; otherwise, if that value is FALSE, then the final result corresponds to the multidimensional view given by $\mathcal{O}_f \cap \mathcal{O}_g$.

Figure 7. Knowledge Discovery Tasks in KBMiner



As shown by the running example above, KBMiner is able to efficiently model and execute KDD processes according to a simple and intuitive metaphor, which, however, allows us to author even-complex KDD tasks via meaningful MR; the combined effects of these two aspects contribute to make KBMiner a very useful tool for real-life data-intensive applications.

It should be noted, here, that this visual approach is particularly convenient in the emerging big data setting, due to the fact that, in such *context, data scientists* do not have a-priori knowledge about the intrinsic characteristics of target big datasets (e.g., data distributions, data ranges, data values, etc.) so that having available a visual tool that can simplify and make semi-automatic and iterative the desired knowledge discovery processes has a pivotal significance, as highlighted in recent studies (e.g., Andrienko et al., 2020).

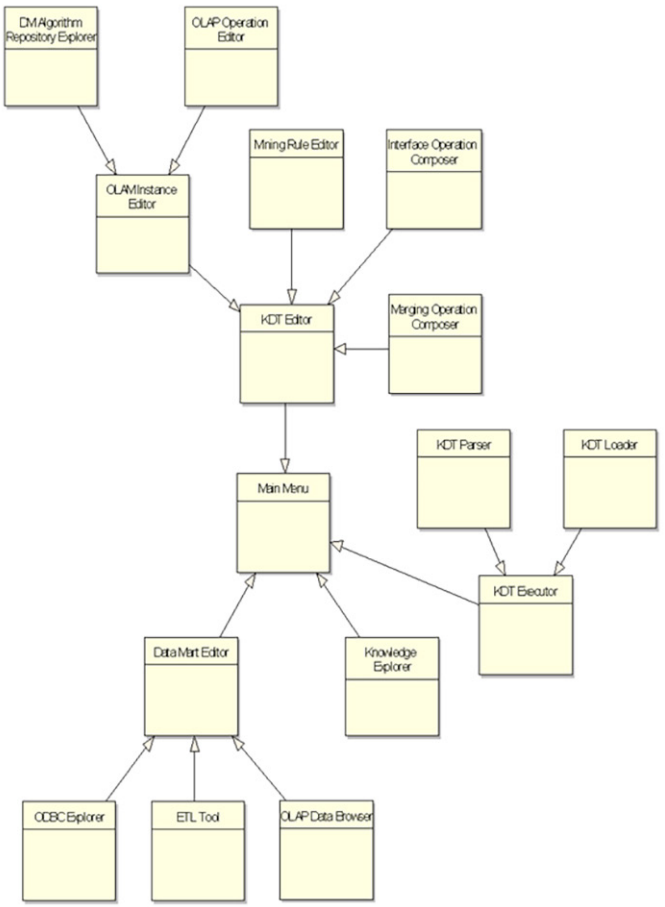
MRE – KDD$^+$ UML Class Diagram

KBMiner is a set of features essential to effectively support a complex knowledge discovery process, starting from data integration and preprocessing, to multidimensional exploration, analytical modeling, and result interpretation. These features allow users to apply different analysis approaches, build and configure mining workflows, prepare and manage data, and visualize results. Figure 8 shows the KBMiner UML class diagram, which puts in evidence the various components of KBMiner, and how these components interact to discover useful knowledge from large databases and data warehouses according to the guidelines of $\mathscr{MRE} - \mathscr{KDD}^+$. As shown in Figure 8, KBMiner components are the following:

- Main Menu, which coordinates all the KBMiner components;
- Data Mart Editor, which allows users to edit and build subject-oriented data marts (it corresponds to the construct MMF of $\mathscr{MRE} - \mathscr{KDD}^+$);
- OBDC Explorer, which provides data access and data source linkage functionalities;
- ETL tool, which supports ETL tasks;
- OLAP data browser, which allows users to access, explore and query OLAP-data-cubes/data-marts;
- KDT editor, which allows users to edit KDT (it corresponds to the construct KDF of $\mathscr{MRE} - \mathscr{KDD}^+$);
- OLAM instance editor, which builds an $\mathscr{MRE} - \mathscr{KDD}^+$-based mining model starting from an input KDT;

- OLAP operation editor, which supports the editing of conventional and complex OLAP operators/tools (it corresponds to the constructs MCF and $\mathbf{MCF}_{E}$ of $\mathcal{MRE} - \mathcal{KDD}^{+}$);
- DM algorithm repository explorer, which allows users to access and explore the DM algorithm repository;
- mining rule editor, which allows users to edit MR;
- interface operation composer, which allows users to edit interface operations between DM algorithms;
- merging operation composer, which allows users to edit merging operations between MR at intermediate tasks of a KDD process as well as at the output layer (it corresponds to the construct MEF of $\mathcal{MRE} - \mathcal{KDD}^{+}$);
- KDT executor, which executes KDT;
- KDT parser, which parses KDT and builds the corresponding $\mathcal{MRE} - \mathcal{KDD}^{+}$-based mining models;
- KDT loader, which loads KDT in the vest of $\mathcal{MRE} - \mathcal{KDD}^{+}$-based mining models to be executed;
- knowledge explorer, which allows users to load and explore a previously-edited $\mathcal{MRE} - \mathcal{KDD}^{+}$-based mining model, and, if needed, re-execute it.

Figure 8. KBMiner UML Class Diagram



It should be noted, here, that the modular software nature of KBMiner allows us to obtain an easily maintainable tool that can also be easily extended as to include innovative components of the fundamental knowledge discovery phase over big datasets.

CASE STUDIES

In this section, we present three case studies that demonstrate the use of MRE – KDD$^+$ in real-life big data applications. Specifically, the KBMiner has been used to show how MRE – KDD$^+$ can be applied to solve complex problems, and lead OLAM-based big data analytics effectively and efficiently. Additionally, these case studies show how we can use KBMiner for the editing of KDT models and to efficiently support the editing of even-complex KDD processes in a user-friendly manner. This also proves the usability of our proposed framework.

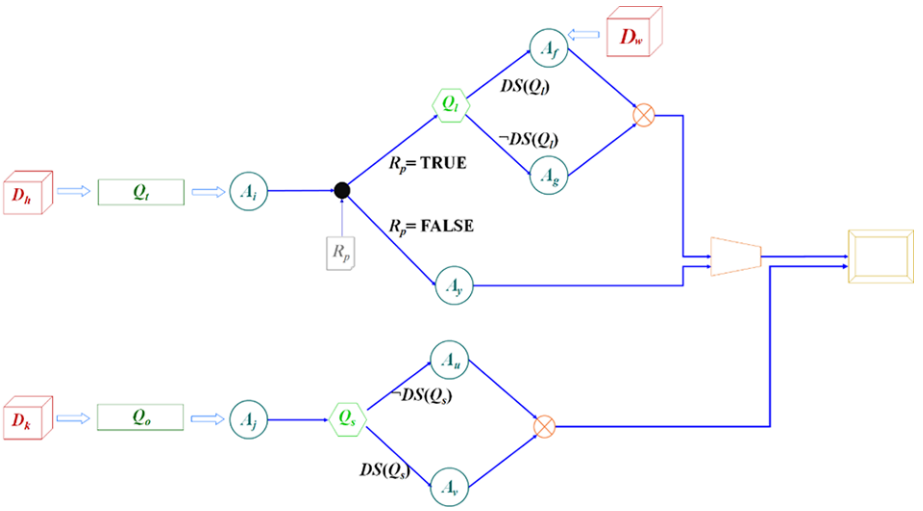
Optimizing E-Commerce Company Customer Service Processes

The first case study examines the use of KBMiner in enhancing customer service processes (CSP) for *e*-commerce (see Cox & Dale, 2001), and how our architecture can be employed in this specific real-life instance. Incorporating OLAP features as well as advanced DM algorithms, our

framework helps in revealing some of the latent (and critical) aspects of the CSP system, which are the response rates, complaint resolutions, and trends of cancellations. The focus is on providing the decision-makers with actionable information that can be useful for enhancing the overall system, decreasing the rate of canceled orders, and increasing the satisfaction of the clients, thereby boosting the performance of the overall system. An ideal dataset for this case study is e-commerce customer service dataset (Kabir, 2025), which is a customer satisfaction dataset for an e-commerce platform, with over one-month period records, resulting in 85,907entries spreads over 20 features, including customer scores, item pricing, customer comments, interaction type, and agent and management data. The dataset closely resembles real-life structures. It is ideal for CSP analytics and other exploratory data analysis that are used to assess customer service performance, predict customer satisfaction, and examine client behavior in the e-commerce industry.

Figure 9 displays the KDT model in KBMiner for this case study. Here, data marts $\mathscr{D}_k$ and $\mathscr{D}_h$ are exploited to store and manage customer service data, including user cancellations, complaints, response times, and product information such as categories, orders, quality, and returns.

Figure 9. Customer Service Processes Knowledge Discovery Tasks: A Case Study in KBMiner



The data mart $\mathscr{D}_h$ stores product information. The OLAP query Q_t executes against the data mart $\mathscr{D}_h$, resulting in a multidimensional view that represents various dimensions, such as product categories, regions, and order history. The result of this query, i.e., the multidimensional view, is referred to as $\mathcal{O}_i$.

The KBMiner tool defines specific logical conditions, MR, that control the process through the KDT and dynamically guide knowledge extraction by evaluating whether rules return TRUE or FALSE. These rules can be as follows. If a data cluster exceeds a defined threshold (high returns in a particular Region), apply a pre-defined classification algorithm to reduce dimensionality (by Time). In the running example of Figure 9, the MR R_p is checked according to the information stored in the Q_t's query result $\mathcal{O}_i$.

If R_p is False, then $\mathcal{O}_i$ is subjected to the DM algorithm A_y, which is the association rule mining algorithm Apriori (Agrawal et al., 1996). Algorithm Apriori can help to uncover relationships between cancellation reasons and other variables. The goal is to reveal patterns related to order cancellations of specific features and to investigate the association between specific weekdays and higher cancellation rates.

If R_p is TRUE, then $\mathcal{O}_i$ is divided into two multidimensional views using the OLAP query Q_l, namely $DS(Q_l)$ and $\neg DS(Q_l)$, respectively, being the split based on *Time*. The first view $DS(Q_l)$ is for cancellations on weekends, and the second view $\neg DS(Q_l)$ is for other days. For the cancellations on weekends, we use another data mart, $\mathcal{D}_w$, which stores delayed orders (not shown in the KDT for the sake of simplicity). Then, using another OLAP query, we integrate all results in the multidimensional views named $\mathcal{O}_f$, Finally, we use the same previous Apriori mining algorithm A_y over $\mathcal{O}_f$, resulting in the multidimensional view named $\mathcal{O}_g$.

Therefore, summarizing, the final result of the upper part of the KDT for the example CSP system depends on the evaluation of the MR R_p against $\mathcal{O}_i$. If R_p returns FALSE, the final result is $\mathcal{O}_y$. By the contrary, if R_p returns *TRUE*, the final result is derived from the *intersection* of the multidimensional views $\mathcal{O}_f$ and $\mathcal{O}_g$.

The lower part of the KDT for the example CSP system is straightforward. The data mart $\mathcal{D}_k$ stores customer information. The OLAP query Q_o on $\mathcal{D}_k$ produces a multidimensional view $\mathcal{O}_j$ representing the aggregations of all the spending on the *e*-commerce and cancelation rate. Next, a multidimensional view is represented by the result of the OLAP query Q_s on output $\mathcal{O}_j$ to split the view into two partitions, owing to the massive size of data, namely $DS(Q_s)$ and $\neg DS(Q_s)$, respectively. This can also help us to perform multiple DM algorithms, each for specific group of dimensions. In this running example, the split is based on two age groups: the first one above 40 years, and the second one under 40 years. For customers above 40, we apply again the Apriori algorithm, in order to discover the patterns between high cancellations and total spending. For customers under 40, we apply the same algorithm, however, this time to uncover relationships between high cancellations and education levels. The final results are the combination of the two outputs.

Finally, as shown in Figure 9, the final result is represented, according to the ensemble model, by the combination of *all* the two partial results, by using the *Mining Results Merging Component* (see Section 5).

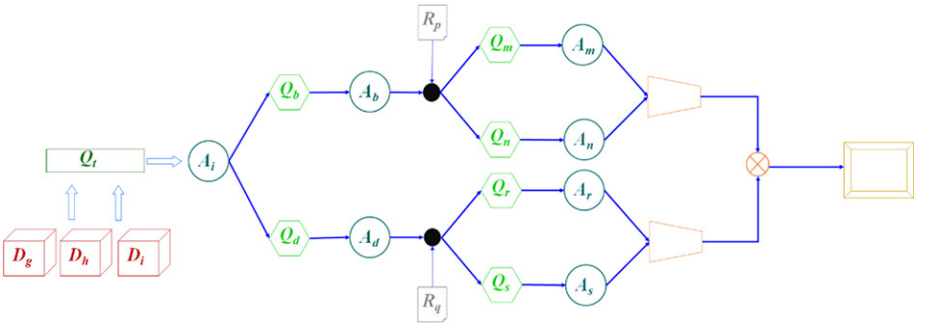
Improving Company Procure-to-Pay Cycles

In the second case study, we explored how our framework can be applied differently to optimize the procure-to-pay (P2P) process in a company (e.g., (Trautmann & Lasch, 2020)). We used procurement KPI analysis dataset (Himanshi, 2025), which is an anonymized dataset with 700 real-life purchase orders, that cover the procurement activities of a multinational corporation 2022 to 2023. It includes common supply chain problems like defects in products, supplier delays, and compliance problems. It is perfect for evaluating vendor performance, cost optimization, and procurement efficiency since it includes data from five different suppliers and integrated metrics for cost savings, defect rates, and on-time delivery. The dataset is useful for forecasting, supplier risk assessment and compliance checks since it incorporates real-life complexities like incomplete deliveries and missing data, as well as market movements like inflation.

By integrating OLAP and advanced DM techniques, even in this second case study, MRE – KDD$^+$ helped to uncover hidden patterns in obtaining actions, such as order approval times, vendor performance, and stock management inefficiencies. The final goal was to provide actionable insights for decision-makers to streamline operations, reduce delays, and improve overall procurement efficiency, ultimately leading to better business performance. In the P2P study, we mined procurement and vendor data to discover hidden patterns between them and to nicely support OLAM-based big data analytics.

The KDT model for this case study is depicted in Figure 10. Here, we introduce three data marts, namely $\mathcal{D}_i$, $\mathcal{D}_g$ and $\mathcal{D}_h$. The first data mart $\mathcal{D}_i$ stores procurement data (such as purchase orders, vendor details, product details, and approval status). The second data mart $\mathcal{D}_g$ stores inventory data (including stock levels, product SKUs, aging stock, and reserved stock), Finally, the third data mart $\mathcal{D}_h$ stores finance data (i.e., comprising invoices, payments, price changes, and purchase order costs).

Figure 10. Procure-to-Pay Cycle Knowledge Discovery Tasks: A Case Study in KBMiner



In the reference case study, all the relevant data of the target P2P process were grouped, filtered, and summarized with a suitable collection of OLAP queries. We noticed that the P2P process had many dimensions, including purchase orders, vendors, products, and approval timelines; therefore, it could be complex as it requires collating information from different sources. Things like total lead times, approval time delays, and vendor performance associated with various transactions are some of the insights that these queries extract.

By combining this data into a unique multidimensional OLAP view, namely $\mathcal{O}_i$, we could simply have complete control of the underlying DM process. This enabled us to discover bottlenecks, compare vendor efficiency, and improve the P2P cycle. Furthermore, combining all important data into a single view simplified complicated analysis, making it easier to identify patterns.

After the OLAP queries had aggregated the necessary data into the unified multidimensional view $\mathcal{O}_i$ of the P2P process, we applied the multidimensional association rule mining algorithm A_i (Xu & Wang, 2006). This DM algorithm is the best solution for discovering hidden patterns that can deal with a multidimensional dataset, such as dealing with vendors, product categories, approval time, and procurement delay. For instance, it can be found that certain products associated with some vendors take relatively more time to approve. This way, we can generate rules to develop correlations between various aspects of the P2P process. For instance, a rule may look like this: if a product comes from vendor X and has an enormous quantity, the approval period is likely to be long. Once these rules have been produced, the algorithm computes their confidence and support values. The confidence value represents how frequently the rule applies, whereas the support value tells how frequently the linked item set exists in the dataset. These values contributed to determining the strength and relevance of each rule, which guided the procedure in the following phase.

Next (see Figure 10), we developed the insights gained from association rules formed previously with the help of two new OLAP queries, providing two additional views, namely $\mathcal{O}_b$ and $\mathcal{O}_d$, respectively. These views focused on facets of the P2P process and provided deeper insights into vendor-specific behavior and inventory dynamics, respectively. These views would be essential in moving forward the KDT process by bringing forth better data for analysis.

The first vendor-specific efficiency view $\mathcal{O}_b$ focused on the performance of different vendors with order quantities, lead times, and approval times. It grouped vendors by order volumes and aggregates the following metrics: average lead time (time from order placement to delivery), approval time (time taken to approve purchase orders), and order quantities (small, medium, large). The second inventory view $\mathcal{O}_d$ focused on inventory management by examining how stock moved through the system and how long it stayed in storage. The query filtered stock by time (how long it has been in inventory) and utilization rate, and it aggregated data related to inventory flow and aging stock.

The generated outputs underwent additional analysis by means of specialized algorithms that could derive knowledge from the respective views. This mining process was controlled by mining

rules, which served as criteria that helpd decide which procedures would be employed in the next phase, on the basis of the setup conditions.

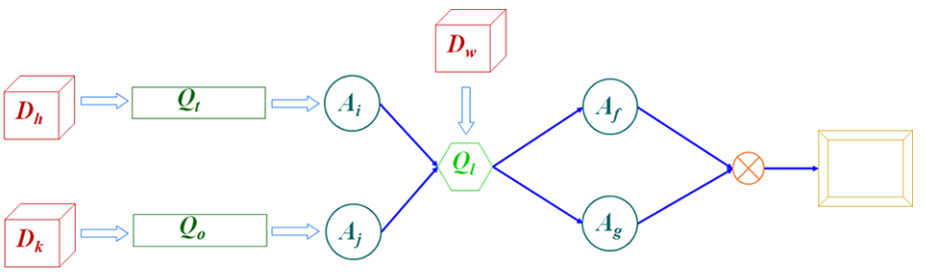
For the vendor-specific efficiency view $\mathcal{O}_b$, the output data underwent clustering analysis using algorithm K-means (MacQueen, 1967) A_b. K-means groups vendors based on similar performance metrics (e.g., approval times, lead times, and order volumes). Afterwards, the MR rule R_p checked if a vendor has consistently high approval times for large orders, the mining rule triggered further analysis with the outlier detection algorithm DBSCAN (Ester et al., 1996) A_m, in order to identify vendors that significantly deviate from expected behavior. Otherwise, we applied the decision tree algorithm (Von Winterfeldt & Edwards, 1986) A_n, in order to predict which vendors were likely to remain high-performing based on historical data. Thus, the final result was determined by the MR rule R_p. We then applied the same process to the second multidimensional view $\mathcal{O}_d$. Finally, all the partial results were ensembled (see the "Reference Architecture for Supporting OLAM-Based Big Data Analytics Over Big Data Warehouses" section, below.

Mining the Order-to-Cash Process to Accelerate Company Workflows

This case study demonstrates how our framework can be effectively used to enhance another real-life scenario, the order-to-cash (O2C) process. The O2C process encompasses multiple stages, from the placing of an order by the customer to the organization receipt of the entire amount paid for the goods or services provided. Improving the efficiency of the O2C process is vital as it has a direct impact on critical aspects of the target organization, such as: (*i*) the cash flow of the organization, (ii) the amount of time taken before closing a sale, and (iii) the satisfaction of customers. This case study addresses these challenges to some extent using the advanced tool KBMiner, in order to help identify the patterns in the O2C process and provide useful information to decision makers. The E-Commerce Analytics dataset (Dee Dee, 2024) was used for this case study, this dataset primarily focuses on three different platforms and offers thorough insights into the e-commerce food delivery industry. It encompasses important O2C processes and customer experience features such as delivery timings, reviews, service ratings, and purchase information, including payment method, pricing, and discounts. Additionally, the dataset is ideal for evaluating client satisfaction, comparing platform performance, and *streamlining delivery logistics*. Therefore, this dataset is a strong basis for *operational* and *strategic* decision-making in O2C analysis, which also supports use cases such as *sentiment analysis*, *revenue trend analysis*, and *predictive modeling for delays*.

Figure 11 shows the KDD process modeled by the O2C KDT in KBMiner. The process begins with providing the right data into the system. After, it should be noted that the O2C process is complex in nature, and it is controlled by multiple components and variables. For this reason, we divided the process into two parts, each one dealing with a certain partition of data and employing appropriate DM algorithms to carefully discover crucial insights for each part.

Figure 11. Order-to-Cash Process Knowledge Discovery Task: A Case Study in KBMiner



At the input, we had two data marts, namely $\mathscr{D}_h$ and $\mathscr{D}_k$. $\mathscr{D}_h$ stores data of customers such *as* payment history, order patterns, loyalty scores, credit ratings, and demographic details. $\mathscr{D}_{\mathrm{k}}$ stores data related to the sales process, including sales orders, products sold, pricing information, discounts, sales regions, and approval statuses. It should be noted that there was a possibility for the following stages where other data marts might be incorporated into the target process.

As shown in Figure 11, the initial step included the extraction of data from the data marts. In the first part of the process, we began by executing the OLAP query Q_t against the data mart $\mathscr{D}_h$, which produced in output a multidimensional view $\mathscr{O}_i$. Q_t focused on extracting data related to customer payment behaviors from $\mathscr{D}_h$, by analyzing payment history, order frequency, and loyalty patterns. In the second part of the process, we performed the OLAP query Q_o against the data mart $\mathscr{D}_{\mathrm{k}}$, which produced in output a multidimensional view $\mathscr{O}_{\mathrm{j}}$. $\mathscr{O}_{\mathrm{j}}$ retrieved sales performance by taking into account key elements such as sales orders, products sold, pricing information, discounts, sales regions, and approval statuses.

After generating the OLAP-based multidimensional views, we applied DM algorithms on both sides to uncover hidden relationships and patterns within the O2C process. In particular, we apply the Apriori algorithm (35), as in the first case study, to customer data extracted via Q_t. This algorithm helps to identify frequent patterns and relationships between customer order behaviors and payment delays, thus identifying actionable business intelligence results (e.g., customers with total orders exceeding $10,000and a high order frequency, the likelihood of delayed payment increases by 40%). The algorithm was also applied to sales data extracted via Q_o, as to uncover patterns similar to the following rule: large quantities orders from point of sale X consistently face approval delays exceeding 7 days. Cooperatively, these DM algorithms help uncover relationships that may not be immediately obvious, such as correlations between customer types and payment behaviors, or between product categories and order approval times. The data retrieved were combined in the data mart $\mathscr{D}_w$, on top of which the OLAP query Q_l applies to extract join (i.e., combined) data.

Finally, other two instances of Apriori were applied to the combined results, namely A_f and A_g, with similar goals as described in the first part of the target O2C KDT, and then we combined the derived results. This approach allowed us to create a comprehensive view of the O2C process. The implemented combinations enabled decision makers to understand how various factors, such as customer factors and order management, interact and impact on the overall cash flow and selling progress efficiency. Similarly, the target O2C KDT may be integrated with other different DM algorithms to the new view to extract various insights. Still, the final result corresponds to the combination of all the partial results.

EXPERIMENTAL ASSESSMENT

To validate the effectiveness and scalability of our proposed framework, we conducted a comprehensive experimental assessment. The experiments were designed to reflect real-world big data scenarios, and we compared the performance and capabilities of our framework against two well-known environments: WEKA and *SQL* Server analysis services (SSAS). The assessment was divided into two main parts: First, we detailed the implementation aspects of the framework, and then we presented and discussed the experimental results obtained from various evaluations.

Implementation Details

The proposed framework was implemented with a focus on scalability, modularity, and compatibility with modern big data processing technologies. The OLAP modeling and aggregation components were realized using Apache Hive, a powerful data warehouse infrastructure built on top of Apache Hadoop. Apache Hive facilitates efficient storage, querying, and multidimensional aggregation of large-scale datasets using HiveQL, a SQL-like language. Through Hive's support for

external tables and its integration with Hadoop distributed file system (HDFS), the framework cold perform distributed processing of OLAP cubes with optimized query execution.

The overall framework was developed in Java, allowing seamless integration with both the OLAP and OLAM engines and ensuring platform independence. Java's rich ecosystem of libraries and its performance in handling multi-threaded operations make it a robust choice for implementing data-intensive applications such as ours.

Once the OLAP engine completes the computation of the multidimensional OLAP data views, the output is forwarded to the OLAM engine. This engine is responsible for conducting advanced mining operations on the aggregated data. The OLAM component interacts with a repository of predefined DM models and algorithms, which it uses to extract insightful patterns, correlations, or trends from the OLAP data. Following the mining step, the framework includes a visualization module that renders the mining results in an intuitive and user-friendly format. This step is critical to help analysts, researchers, and decision-makers interpret the patterns and insights extracted by the OLAM engine. The visualizations are customizable, supporting both tabular and graphical formats.

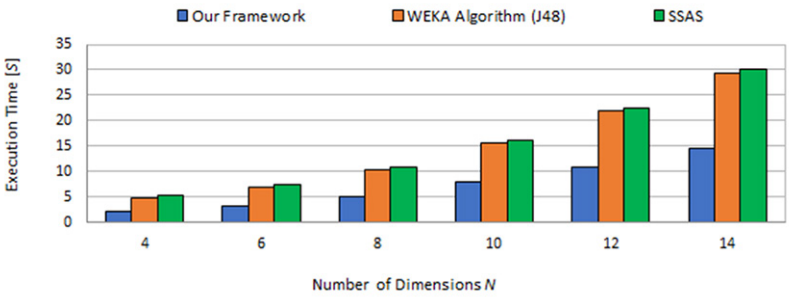
The tight coupling between OLAP, OLAM, and visualization layers ensures that the end-to-end pipeline from raw data ingestion to analytical insight generation is both streamlined and efficient.

Experimental Results

To evaluate the performance of our proposed framework, we conducted two main experimental analyses. The first analysis focused on comparing the execution time across increasing dimensionality, while the second evaluated the throughput in terms of queries processed per second. In both analyses, our framework is compared against two widely used data mining environments: WEKA (i.e., J48 decision tree algorithm (Liang et al., 2023) and SQL server analysis services (SSAS).

Figure 12 shows the execution time (in seconds) required by each system as the number of dimensions in OLAP data cube increases.

Figure 12. Execution Time Analysis Between Our Framework, WEKA, and Server Analysis Services Across Varying Dimension Numbers

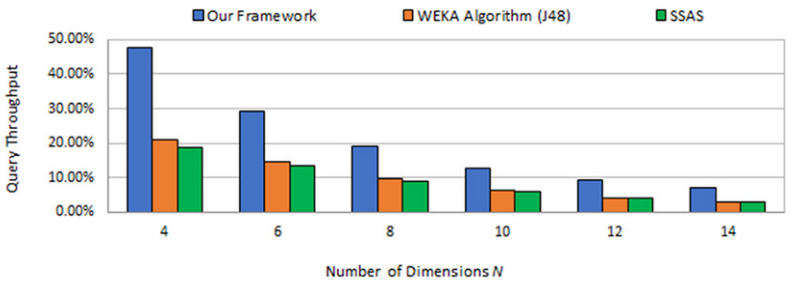


As shown in Figure 12, execution time increases with dimensionality for all systems. However, our framework consistently outperforms both WEKA and SSAS across all dimensions.

This performance gain is attributed to the integration of OLAP operations with optimized Java-based data mining modules and the use of Apache Hive for efficient multidimensional aggregation. WEKA, which operates in-memory and lacks native support for distributed processing, exhibits a steep rise in execution time as the number of dimensions increases. SSAS performs better than WEKA but remains slower than our framework because of its limited flexibility in mining workflows and its dependency on pre-defined data cube structures.

On the other hand, Figure 13 shows throughput metrics, measured in queries per second, of each system under the same dimensional settings.

Figure 13. Throughput Analysis Between Our Framework, WEKA, and Server Analysis Services Across Varying Dimension Numbers



From Figure 13, it is revealed that as the number of dimensions increases, throughput decreases for all three systems due to the increased computational complexity. However, our framework maintains significantly higher throughput across all tested dimensionalities.

At lower dimensions, the performance gap is moderate, but as dimensionality grows, our framework shows much stronger resilience. This demonstrates the scalability of the architecture and the effectiveness of our parallel processing and hybrid OLAP-OLAM integration. WEKA shows the sharpest decline in throughput, reaffirming its limitations in high-dimensional data mining tasks. SSAS exhibits a slower drop, yet it is still outperformed by our approach.

LIMITATIONS AND FUTURE RESEARCH DIRECTIONS

Although the MRE – KDD$^+$ architecture and its visual tool, KBMiner, offer an important contribution in the OLAM-based big data analytics area, there are still certain challenges that need to be addressed in further studies. One of the main limitations is scalability. Large data volumes and high-concurrency scenarios, which are typical in real-life settings, have not yet been effectively verified. The suggested framework shows its effectiveness in distributed big data contexts; however, MRE – KDD$^+$ could be further improved and optimized for high scalability.

Another limitation is the lack of real-time analytics support, which adds a significant constraint. The current system is primarily designed for batch processing, making it less suitable for domains requiring low-latency, high-throughput analytics, such as IoT systems or financial fraud detection. Future extensions will explore the integration of real-time stream processing platforms such as Apache Kafka, which handles dynamic knowledge discovery activities and continuous data flows, and stream processing systems (Lavanya, 2020). Additionally, the framework currently focuses on traditional data mining algorithms and does not yet natively support advanced ML. While DBMiner, which inspired aspects of our work, aligns more closely with OLAP/OLAM paradigms, integrating modern ML models could offer hybrid analytical capabilities that enhance prediction accuracy, anomaly detection, and adaptive learning. Enabling interoperability between OLAP-based methods and state-of-the-art ML pipelines would significantly broaden the applicability of MRE – KDD$^+$.

Privacy and security also remain pressing concerns. Although $\mathscr{MRE} - \mathscr{KDD}^+$ can integrate heterogeneous data sources, but it does not yet include embedded mechanisms for privacy-preserving analytics. Future versions of the framework may incorporate support for differential privacy, federated

learning, and secure multiparty computation to ensure compliance with data protection regulations in sensitive domains (Cuzzocrea & Soufargi, 2025).

Beyond these aspects, MRE – KDD$^+$ opens several promising research directions that could shape the evolution of OLAM-based big data analytics:

- Dynamic workflow adaptability: Current execution schemes are defined statically through KBMiner. Future enhancements could enable adaptive workflow construction, guided by meta-learning or runtime performance metrics.
- Cross-domain generalization: While the framework has been validated across e-commerce and industrial case studies, its underlying architecture holds potential for broader application in domains such as bioinformatics, smart cities, and environmental monitoring.
- Improved visual reasoning: While KBMiner provides visual modeling for KDD tasks, extending its reasoning capabilities, for instance, through interactive what-if analysis or explainable AI components, would improve end-user engagement and interpretability.
- Data quality and preprocessing: The framework assumes a relatively clean and structured input. Introducing automated data cleaning, temporal alignment, and schema mapping modules would enhance robustness across noisy, heterogeneous sources.

CONCLUSIONS AND FUTURE WORK

Starting from successful OLAM technologies, in this paper, we have presented a complete framework for supporting advanced knowledge discovery from big data warehouses, which is useful for any big data e setting, but with particular emphasis on a general application scenario populated by distributed and heterogeneous data sources, and the integration/data layer of open big data systems. To this end, we have formally provided principles, definitions, and properties of MRE – KDD$^+$, the model underlying the framework we propose. Other contributions of our work are the following:

- a reference architecture implementing the framework, which can be realized in any distributed software platform, under the design guidelines given by component-oriented software engineering best practices;
- KBMiner, a visual tool that allows users to edit even-complex KDD processes according to the MRE – KDD$^+$.
- guidelines in a simple, intuitive and interactive manner;
- definition and implementation of three case studies that clearly demonstrate the use of MRE – KDD$^+$ in real-life big data applications.

Future work is oriented toward extending the actual capabilities of MRE – KDD$^+$ along three main directions:

- embedding novel functionalities for supporting the prediction of events in new DM activities edited by users/applications on the basis of the "history" given by logs of previous KDD processes implemented in similar or correlated application scenarios;
- embedding novel functionalities for supporting the privacy of users while executing the KDT codified by MRE – KDD$^+$ since they usually access and process sensitive ranges of data, and privacy breaches can arise (e.g., (Jain et al., 2016)));
- integrating the proposed framework with novel and exciting AI methods that have recently leaded the state-of-the-art research scene (e.g., (Wu et al., 2013; Howlader et al., 2018; Hossain Faruk et al., 2021; Masum et al., 2021)).

FUNDING

This research was supported by the ICSC National Research Centre for High Performance Computing, Big Data and Quantum Computing within the NextGenerationEU program (Project Code: PNRR CN00000013).

COMPETING INTERESTS

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

CORRESPONDING AUTHOR

Correspondence should be addressed to Alfredo Cuzzocrea: alfredo.cuzzocrea@unical.it.

REFERENCES

Agarwal, S., Agrawal, R., Deshpande, P., Gupta, A., Naughton, J. F., Ramakrishnan, R., & Sarawagi, S. (1996). On the computation of multidimensional aggregates. *22nd International Conference on Very Large Data Bases*, 506–521.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast discovery of association rules. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.), *Advances in knowledge discovery and data mining* (pp. 307–328). AAAI/MIT Press.

Andrienko, N., Andrienko, G., Fuchs, G., Slingsby, A., Turkay, C., & Wrobel, S. (2020). *Visual analytics for data scientists*. Springer. DOI: 10.1007/978-3-030-56146-8

Azgomi, H., & Sohrabi, M. K. (2021). MR-MVPP: A map-reduce-based approach for creating MVPP in data warehouses for big data applications. *Information Sciences*, *570*, 200–224. DOI: 10.1016/j.ins.2021.04.004

Ben-Efraim, H., Davidson, S. B., & Somech, A. (2025). SHARQ: Explainability framework for association rules on relational data. *Proceedings of the ACM on Management of Data, 3*(1), Article 76, 1–25. DOI: 10.1145/3709726

Chai, C., Tang, N., Fan, J., & Luo, Y. (2023). Demystifying artificial intelligence for data preparation. In *SIGMOD '23: Companion of the 2023 ACM International Conference on Management of Data* (pp. 13–20). Association for Computing Machinery. DOI: 10.1145/3555041.3589406

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *SIGMOD Record*, *26*(1), 65–74. DOI: 10.1145/248603.248616

Cheeseman, P. C., & Stutz, J. C. (1996). Bayesian classification (AutoClass): Theory and results. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.), *Advances in knowledge discovery and data mining* (pp. 153–180). AAAI/MIT Press.

Chen, W., Wang, H., Zhang, X., & Lin, Q. (2017). An optimized distributed OLAP system for big data. In *2nd IEEE International Conference on Computational Intelligence and Applications* (pp. 1–5). IEEE. DOI: 10.1109/CIAPP.2017.8167056

Colliat, G. (1996). OLAP, relational, and multidimensional database systems. *SIGMOD Record*, *25*(3), 64–69. DOI: 10.1145/234889.234901

Cox, J., & Dale, B. G. (2001). Service quality and e-commerce: An exploratory analysis. *Managing Service Quality*, *11*(2), 121–131. DOI: 10.1108/09604520110387257

Cuzzocrea, A. (2007). An OLAM-based framework for complex knowledge pattern discovery in distributed and heterogeneous data sources and cooperative information systems. In Song, I. Y., Eder, J., & Nguyen, T. M. (Eds.), *9th International Conference on Data Warehousing and Knowledge Discovery* (pp. 181–198). Springer. DOI: 10.1007/978-3-540-74553-2_17

Cuzzocrea, A. (2015). Aggregation and multidimensional analysis of big data for large-scale scientific applications: Models, issues, analytics, and beyond. In Gupta, A., & Rathbun, S. (Eds.), *27th ACM International Conference on Scientific and Statistical Database Management* (pp. 1–6). Association for Computing Machinery. DOI: 10.1145/2791347.2791377

Cuzzocrea, A. (2022). Multidimensional big data analytics over big web knowledge bases: Models, issues, research trends, and a reference architecture. In *8th IEEE International Conference on Multimedia Big Data* (pp. 1–6). IEEE. DOI: 10.1109/BigMM55396.2022.00008

Cuzzocrea, A. (2023). A reference architecture for supporting multidimensional big data analytics over big web knowledge bases: Definitions, implementation, case studies. *International Journal of Semantic Computing*, *17*(4), 545–568. DOI: 10.1142/S1793351X2364002X

Cuzzocrea, A. (2025). Privacy-preserving OLAP against big query workloads: Innovative theories and theorems. *Distributed and Parallel Databases*, *43*(1), 2. DOI: 10.1007/s10619-024-07445-5

Cuzzocrea, A., Bellatreche, L., & Song, I. Y. (2013). Data warehousing and OLAP over big data: Current challenges and future research directions. In *16th ACM International Workshop on Data Warehousing and OLAP*, (pp. 67–70). Association for Computing Machinery. DOI: 10.1145/2513190.2517828

Cuzzocrea, A., & Gunopulos, D. (2014). A decomposition framework for computing and querying multidimensional OLAP data cubes over probabilistic relational data. *Fundamenta Informaticae*, *132*(2), 239–266. DOI: 10.3233/FI-2014-1042

Cuzzocrea, A., Saccà, D., & Serafino, P. (2007). Semantics-aware advanced OLAP visualization of multidimensional data cubes. *International Journal of Data Warehousing and Mining*, *3*(4), 1–30. DOI: 10.4018/jdwm.2007100101

Cuzzocrea, A., Song, I. Y., & Davis, K. C. (2011). Analytics over large-scale multidimensional data: The big data revolution! In *14th ACM International Workshop on Data Warehousing and OLAP* (pp. 101–104). Association for Computing Machinery. DOI: 10.1145/2064676.2064695

Cuzzocrea, A., & Soufargi, S. (2025). Privacy-preserving multidimensional big data analytics models, methods and techniques: A comprehensive survey. *Expert Systems with Applications*, *270*, 126387. DOI: 10.1016/j.eswa.2025.126387

Dedić, N., & Stanier, C. (2017). Towards differentiating business intelligence, big data, data analytics and knowledge discovery. In Piazolo, F., Geist, V., Brehm, L., & Schmidt, R. (Eds.), *Innovations in Enterprise Information Systems Management and Engineering* (pp. 114–122). Springer. DOI: 10.1007/978-3-319-58801-8_10

Dee Dee. (2024). *eCommerce Customer Service Satisfaction V3*. Kaggle. https://www.kaggle.com/datasets/ddosad/ecommerce-customer-service-satisfaction

Desgourdes, C., & Ram, J. (2024). The role of big data analytics for decision-making in projects: Uses and challenges. *Enterprise Information Systems*, *18*(4), 2317153. DOI: 10.1080/17517575.2024.2317153

Di Tria, F., Lefons, E., & Tangorra, F. (2014). Big data warehouse automatic design methodology. In Hu, W., & Kaabouch, N. (Eds.), *Big data management, technologies, and applications* (pp. 115–149). IGI Global. DOI: 10.4018/978-1-4666-4699-5.ch006

Do, N., Bae, S., & Park, C. (2015). Interactive analysis of product development experiments using on-line analytical mining. *Computers in Industry*, *66*, 52–62. DOI: 10.1016/j.compind.2014.09.003

Elder, J. F.IV, & Pregibon, D. (1996). A statistical perspective on knowledge discovery in databases. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.), *Advances in knowledge discovery and data mining* (pp. 83–113). AAAI/MIT Press.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Simoudis, E., Han, J., & Fayyad, U. (Eds.), *2nd ACM International Conference on Knowledge Discovery and Data Mining* (pp. 226–231). AAAI Press.

Ester, M., Kriegel, H., & Xu, X. (1995). Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. In Egenhofer, M. J., & Herring, J. R. (Eds.), *4th International Symposium on Advances in Spatial Databases* (pp. 67–82). Springer. DOI: 10.1007/3-540-60159-7_5

Fang, M., Shivakumar, N., Garcia-Molina, H., Motwani, R., & Ullman, J. D. (1998). Computing iceberg queries efficiently. In Gupta, A., Shmueli, O., & Widom, J. (Eds.), *24th International Conference on Very Large Data Bases* (pp. 299–310). Morgan Kaufman Publishers.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery: An overview. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.), *Advances in knowledge discovery and data mining* (pp. 1–34). AAAI/MIT Press.

Fernandes, E., Moro, S., & Cortez, P. (2023). Data science, machine learning and big data in digital journalism: A survey of state-of-the-art, challenges and opportunities. *Expert Systems with Applications*, *221*, 119795. DOI: 10.1016/j.eswa.2023.119795

Goebel, M., & Gruenwald, L. (1999). A survey of data mining and knowledge discovery software tools. *SIGKDD Explorations*, *1*(1), 20–33. DOI: 10.1145/846170.846172

Grady, N. W. (2016). KDD meets big data. In Joshi, J., Karypis, G., Liu, L., Hu, X., Ak, R., Xia, Y., Xu, W., Sato, A.-H., Rachuri, S., Ungar, L., Yu, P. S., Govindaraju, R., & Suzumura, T. (Eds.), *2016 IEEE International Conference on Big Data* (pp. 1603–1608). IEEE. DOI: 10.1109/BigData.2016.7840770

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., & Pirahesh, H. (1997). Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub totals. *Data Mining and Knowledge Discovery*, *1*(1), 29–53. DOI: 10.1023/A:1009726021843

Guo, Y., Zhang, W., Qin, Q., Chen, K., & Wei, Y. (2023). Intelligent manufacturing management system based on data mining in artificial intelligence energy-saving resources. *Soft Computing*, *27*(7), 4061–4076. DOI: 10.1007/s00500-021-06593-5

Guo, Y., Zhou, J., Qin, Q., Wei, Y., & Zhang, W. (2023). An Improved algorithm and implementation of data mining for intelligent manufacturing association rules based on pattern recognition. *IEEE Consumer Electronics Magazine*, *12*(2), 94–99. DOI: 10.1109/MCE.2022.3149210

Han, J. (1997). OLAP Mining: An integration of OLAP with data mining. In Spaccapietra, S., & Maryanski, F. (Eds.), *IFIP TC2 WG2.6/DS-7 1997* (pp. 3–20). Springer.

Han, J., Cai, Y., & Cercone, N. (1993). Data-driven discovery of quantitative rules in relational databases. *IEEE Transactions on Knowledge and Data Engineering*, *5*(1), 29–40. DOI: 10.1109/69.204089

Han, J., & Fu, Y. (1996). Attribute-oriented induction in data mining. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 399–421). AAAI/MIT Press.

Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., Li, D., Lu, Y., Rajan, A., Stefanovic, N., Xia, B., & Zaïane, O. R. (1996). DBMiner: A system for mining knowledge in large relational databases. In Simoudis, E., Han, J., & Fayyad, U. (Eds.), *2nd ACM International Conference on Knowledge Discovery and Data Mining* (pp. 250–255). AAAI Press.

Han, J., & Kamber, M. (2000). *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers.

Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1996). Implementing data cubes efficiently. *SIGMOD Record*, *25*(2), 205–216. DOI: 10.1145/235968.233333

Himanshi. (2025). *E-commerce analytics: Swiggy, zomato, blinkit.* Kaggle. https://www.kaggle.com/datasets/logiccraftbyhimanshi/e-commerce-analytics-swiggy-zomato-blinkit

Ho, C., Agrawal, R., Megiddo, N., & Srikant, R. (1997). Range queries in OLAP data cubes. *SIGMOD Record*, *26*(2), 73–88. DOI: 10.1145/253262.253274

Hossain Faruk, M. J., Shahriar, H., Valero, M., Barsha, F. L., Sobhan, S., Khan, M. A., Whitman, M. E., Cuzzocrea, A., Lo, D. C., Rahman, A., & Wu, F. (2021). Malware detection and prevention using artificial intelligence techniques. In Chen, Y., Ludwig, H., Tu, Y., Fayyad, U., Zhu, X., Hu, X., Byna, S., Liu, X., Zhang, J., Pan, S., Papalexakis, V., Wang, J., Cuzzocrea, A., & Ordonez, C. (Eds.), *2021 IEEE International Conference on Big Data* (pp. 5369–5377). IEEE. DOI: 10.1109/BigData52589.2021.9671434

Howlader, P., Pal, K. K., Cuzzocrea, A., & Kumar, S. D. M. (2018). Predicting Facebook-users' personality based on status and linguistic features via flexible regression analysis techniques. In *33rd Annual ACM Symposium on Applied Computing* (pp. 339–345). Association for Computing Machinery. DOI: 10.1145/3167132.3167166

Hu, J. (2019). E-commerce big data computing platform system based on distributed computing logistics information. *Cluster Computing*, *22*(6), 13693–13702. DOI: 10.1007/s10586-018-2074-6

Jain, P., Gyanchandani, M., & Khare, N. (2016). Big data privacy: A technological perspective and review. *Journal of Big Data*, *3*(1), 25. DOI: 10.1186/s40537-016-0059-y

Jukić, N., Sharma, A., Nestorov, S., & Jukić, B. (2015). Augmenting data warehouses with big data. *Information Systems Management*, *32*(3), 200–209. DOI: 10.1080/10580530.2015.1044338

Kabir, S. (2025). *Procurement KPI analysis dataset* (2025). Kaggle. https://www.kaggle.com/datasets/shahriarkabir/procurement-kpi-analysis-dataset

Karayannidis, N., & Sellis, T. K. (2003). SISYPHUS: The Implementation of a chunk-based storage manager for OLAP data cubes. *Data & Knowledge Engineering*, *45*(2), 155–180. DOI: 10.1016/S0169-023X(02)00178-7

Keim, D., Qu, H., & Ma, K. L. (2013). Big-data visualization. *IEEE Computer Graphics and Applications*, *3*(4), 20–21. DOI: 10.1109/MCG.2013.54

Lavanya, K., Venkatanarayanan, S., & Bhoraskar, A. A. (2020). Real-time weather analytics: An End-to-end big data analytics service over Apache Spark with Kafka and long short-term memory networks. *International Journal of Web Services Research*, *17*(4), 15–31. DOI: 10.4018/IJWSR.2020100102

Ledmi, M., Souidi, M. E. H., Hahsler, M., Ledmi, A., & Kara-Mohamed, C. (2023). Mining association rules for classification using frequent generator itemsets in Arules package. *International Journal of Data Mining. Modelling and Management*, *15*(2), 203–221.

Leprince, J., Miller, C., & Zeiler, W. (2021). Data mining cubes for buildings, a generic framework for multidimensional analytics of building performance data. *Energy and Building*, *248*, 111195. DOI: 10.1016/j.enbuild.2021.111195

Li, T., Zhang, Y., Liu, H., Xue, G., & Liu, L. (2022). Fast compressive spectral clustering for large-scale sparse graph. *IEEE Transactions on Big Data*, *8*(1), 193–202. DOI: 10.1109/TBDATA.2019.2931532

Li, Y. (2024). Optimizing the omni channel marketing strategy of green clothing by integrating artificial intelligence and big data. In *3rd ACM International Symposium on Intelligent Unmanned Systems and Artificial Intelligence* (pp. 227–230). Association for Computing Machinery. DOI: 10.1145/3669721.3669726

Liang, L., Cui, H., Arabameri, A., Arora, A., & Danesh, A. S. (2023). Landslide susceptibility mapping: Application of novel hybridization of rotation forests (RF) and Java decision trees (J48). *Soft Computing*, *27*(22), 17387–17402. DOI: 10.1007/s00500-023-08951-x

Liu, C., Chu, W. W., Sabb, F., Parker, D. S., & Bilder, R. (2014). Path knowledge discovery: Multilevel text mining as a methodology for phenomics. In Chu, W. W. (Ed.), *Data mining and knowledge discovery for big data* (pp. 153–192). Springer. DOI: 10.1007/978-3-642-40837-3_5

Liu, D. (2024). Design of data mining system for sports training biochemical indicators based on artificial intelligence and association rules. *International Journal of Data Mining and Bioinformatics*, *28*(3/4), 236–256. DOI: 10.1504/IJDMB.2024.139449

Luo, J., & Xu, Z. (2024). Research on the construction of supply chains of digital twin aquatic products using artificial intelligence. In *2024 ACM International Conference on Big Data and Digital Management* (pp. 19–24). Association for Computing Machinery. DOI: 10.1145/3696500.3696504

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Le Cam, L. M., & Neyman, J. (Eds.), *5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). University of California Press.

Martin, B., & Davis, K. C. (2021). Multi-temperate logical data warehouse design for large-scale healthcare data. *Big Data Research, 25*, Article 100255.

Masmoudi, M., Ben Abdallah Ben Lamine, S., Karray, M. H., Archimede, B., & Baazaoui Zghal, H. (2024). Semantic data integration and querying: A survey and challenges. *ACM Computing Surveys*, *56*(8), 1–35. DOI: 10.1145/3653317

Masum, M., Shahriar, H., Haddad, H., Hossain Faruk, M. J., Valero, M., Khan, M. A., Rahman, M. A., Adnan, M. I., Cuzzocrea, A., & Wu, F. (2021). Bayesian hyperparameter optimization for deep neural network-based network intrusion detection. In Chen, Y., Ludwig, H., Tu, Y., Fayyad, U., Zhu, X., Hu, X., Byna, S., Liu, X., Zhang, J., Pan, S., Papalexakis, V., Wang, J., Cuzzocrea, A., & Ordonez, C. (Eds.), *2021 IEEE International Conference on Big Data* (pp. 5413–5419). IEEE. DOI: 10.1109/BigData52589.2021.9671576

Mokkadem, A., Pelletier, M., & Raimbault, L. (2024). SufRec, an algorithm for mining association rules: Recursivity and task parallelism. *Expert Systems with Applications*, *236*, 121321. DOI: 10.1016/j.eswa.2023.121321

Nemati, H. R., Steiger, D. M., Iyer, L. S., & Herschel, R. T. (2002). Knowledge warehouse: An architectural integration of knowledge management, decision support, artificial intelligence and data warehousing. *Decision Support Systems*, *33*(2), 143–161. DOI: 10.1016/S0167-9236(01)00141-5

Ng, R. T., & Han, J. (1994). Efficient and effective clustering methods for spatial data mining. In Bocca, J. B., Jarke, M., & Zaniolo, C. (Eds.), *20th International Conference on Very Large Data Bases* (pp. 144–155). Morgan Kaufman Publishers.

Ngo, V. M., Le-Khac, N. A., & Kechadi, M. T. (2019). Designing and implementing data warehouse for agricultural big data. In Chen, K., Seshadri, S., & Zhang, L.-J. (Eds.), *8th International Congress on Big Data* (pp. 1–17). Springer-Verlag. DOI: 10.1007/978-3-030-23551-2_1

Ngo, V. M., Le-Khac, N. A., & Kechadi, M. T. (2020). Data warehouse and decision support on integrated crop big data. *International Journal of Business Process Integration and Management*, *10*(1), 17–28. DOI: 10.1504/IJBPIM.2020.113115

Nisar, K., & Shaheen, M. (2023). Determining context of association rules by using machine learning. *Journal of Experimental & Theoretical Artificial Intelligence*, *35*(1), 59–76. DOI: 10.1080/0952813X.2021.1955980

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. In Piatetsky-Shapiro, G., & Frawley, W. J. (Eds.), *Knowledge Discovery in Databases* (pp. 229–248). AAAI Press.

Qi, S., Wang, J., Miao, M., Zhang, M., & Chen, X. (2023). TinyEnc: Enabling compressed and encrypted big data stores with rich query support. *IEEE Transactions on Dependable and Secure Computing*, *20*(1), 176–192. DOI: 10.1109/TDSC.2021.3129332

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers.

Russom, P. (2011). Big data analytics. *TDWI Best Practices Report*. TWDI. https://tdwi.org/research/2011/09/best-practices-report-q4-big-data-analytics.aspx

Santos, M. Y., Martinho, B., & Costa, C. (2017). Modelling and implementing big data warehouses for decision support. *Journal of Management Analytics*, *4*(2), 111–129. DOI: 10.1080/23270012.2017.1304292

Sautot, L., Bimonte, S., & Journaux, L. (2021). A semi-automatic design methodology for (big) data warehouse transforming facts into dimensions. *IEEE Transactions on Knowledge and Data Engineering*, *33*(1), 28–42. DOI: 10.1109/TKDE.2019.2925621

Shahid, A., Nguyen, T. A. N., & Kechadi, M. T. (2021). Big data warehouse for healthcare-sensitive data applications. *Sensors (Basel)*, *21*(7), 2353. DOI: 10.3390/s21072353

Shakhovska, N., Kaminskyy, R., Zasoba, E., & Tsiutsiura, M. (2018). Association rules mining in big data. *International Journal of Computing*, *17*, 25–32. DOI: 10.47839/ijc.17.1.946

Song, J., Guo, C., Wang, Z., Zhang, Y., Yu, G., & Pierson, J. M. (2015). HaoLap: A Hadoop based OLAP system for big data. *Journal of Systems and Software*, *102*, 167–181. DOI: 10.1016/j.jss.2014.09.024

Song, Y., & Ge, C. (2011). Research on the integration architecture of OLAM and OLAP. In *IEEE ICEICE 2011* (pp. 1656–1659). IEEE. DOI: 10.1109/ICEICE.2011.5778086

Sun, Z. (2024). Big Data 4.0 = Meta4 (Big Data) = The Era of Big Intelligence. In Li, Y., Nishi, H., & Pang, P. (Eds.), *7th ACM International Conference on Software Engineering and Information Management* (pp. 14–22). Association for Computing Machinery. DOI: 10.1145/3647722.3647725

Tardío, R., Maté, A., & Trujillo, J. (2020). A new big data benchmark for OLAP cube design using data pre-aggregation techniques. *Applied Sciences (Basel, Switzerland)*, *10*(23), 8674. DOI: 10.3390/app10238674

Tian, Y., Lu, C., Liu, C., Zhang, J., & Huang, Q. (2024). DLCSS: A data life cycle security system for industry data warehouse platforms. In *4th ACM International Conference on Artificial Intelligence, Automation and Algorithms* (pp. 195–199). Association for Computing Machinery. DOI: 10.1145/3700523.3700558

Trautmann, L., & Lasch, R. (2020). Smart contracts in the context of procure-to-pay. In Golinska-Dawson, P., Tsai, K.-M., & Kosacka-Olejnik, M. (Eds.), *Smart and sustainable supply chain and logistics–trends, challenges, methods and best practices* (*Vol. 1*, pp. 3–23). Springer. DOI: 10.1007/978-3-030-61947-3_1

Tsai, C. W., Lai, C. F., Chao, H. C., & Vasilakos, A. V. (2015). Big data analytics: A survey. *Journal of Big Data*, *2*(1), 1–32. DOI: 10.1186/s40537-015-0030-3

Vitter, J. S., & Wang, M. (1999). Approximate computation of multidimensional aggregates of sparse data using wavelets. *SIGMOD Record*, *28*(2), 193–204. DOI: 10.1145/304181.304199

Von Winterfeldt, D., & Edwards, W. (1986). *Decision Analysis and Behavioral Research*. Cambridge University Press.

Wang, Y., & Song, W. (2019). Research on hierarchical mining algorithm of spatial big data set association rules. In G. Gui & L. Yun (Eds.), *Advanced hybrid information processing: Third EAI International Conference, ADHIP 2019* (pp. 200–208). Springer. DOI: 10.1007/978-3-030-36405-2_21

Wang, Z., Huang, J., Xu, L., Zeng, Q., Li, R., Li, S., & Zhou, L. (2022). Big data analysis and potential development research of regional logistics warehousing based on Baidu index and warehouse in cloud. In *5th ACM International Conference on Algorithms, Computing and Artificial Intelligence* (pp. 1–8). Association for Computing Machinery. DOI: 10.1145/3579654.3579776

Warnke, B., Martens, K., Winker, T., Groppe, S., Groppe, J., Adhiyaman, P., Srinivasan, S., & Krishnakumar, S. (2024). ReJOOSp: Reinforcement learning for join order optimization in SPARQL. *Big Data and Cognitive Computing*, *8*(7), 71. DOI: 10.3390/bdcc8070071

Witten, I. H., & Frank, E. (2005). *Data mining - Practical machine learning tools and techniques*. Morgan Kaufmann Publishers.

Wu, Z., Yin, W., Cao, J., Xu, G., & Cuzzocrea, A. (2013). community detection in multi-relational social networks. In Lin, X., Manolopoulos, Y., Srivistava, D., & Huang, G. (Eds.), *14th International Conference on Web Information Systems Engineering* (pp. 43–56). Springer.

Xin, D., Han, J., Cheng, H., & Li, X. (2006). Answering top-K queries with multi-dimensional selections: The ranking cube approach. In U. Dayal, K.-Y. Whang, D. Lomet, G. Alonso, G. Lohman, M. Kersten, S. K. Cha, & Y.-K. Kim (Eds.), *32nd ACM International Conference on Very Large Data Bases* (pp. 463–475). VLDB Endowment.

Xu, W. X., & Wang, R. J. (2006). A fast algorithm of mining multidimensional association rules frequently. In *2006 IEEE International Conference on Machine Learning and Cybernetics*, (1199–1203). IEEE. DOI: 10.1109/ICMLC.2006.258605

Yang, L., & Qi, F. (2024). Strategic transformation of e-commerce big data classification and mining algorithms based on artificial intelligence era. *Informatica (Slovenia), 48*(17).

Zhang, H., Chen, G., Ooi, B. C., Tan, K. L., & Zhang, M. (2015). In-memory big data management and processing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *27*(7), 1920–1948. DOI: 10.1109/TKDE.2015.2427795

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *SIGMOD Record*, *25*(2), 103–114. DOI: 10.1145/235968.233324

Zhao, Y., Deshpande, P., & Naughton, J. F. (1997). An array-based algorithm for simultaneous multidimensional aggregates. *SIGMOD Record*, *26*(2), 159–170. DOI: 10.1145/253262.253288

Ziarko, W. (1994). *Rough sets, fuzzy sets and knowledge discovery*. Springer. DOI: 10.1007/978-1-4471-3238-7