Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A visual analysis approach for data imputation via multi-party tabular data correlation strategies*

Haiyang ZHU†1,2, Dongming HAN1, Jiacheng PAN1, Yating WEI3, Yingchaojie FENG1, Luoxuan WENG1, Ketian MAO1, Yuankai XING2, Jianshu LV2, Qiucheng WAN2, Wei CHEN†‡1

1*The State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310058, China*
2*Wuchan Zhongda Digital Technology Co., Ltd., Hangzhou 310020, China*
3*Zhejiang Metals and Materials Co., Ltd., Hangzhou 310005, China*
†E-mail: hnsyzhy@zju.edu.cn; chenvis@zju.edu.cn

Received July 17, 2023; Revision accepted Oct. 29, 2023; Crosschecked Dec. 4, 2023; Published online Dec. 29, 2023

Abstract: Data imputation is an essential pre-processing task for data governance, aimed at filling in incomplete data. However, conventional data imputation methods can only partly alleviate data incompleteness using isolated tabular data, and they fail to achieve the best balance between accuracy and efficiency. In this paper, we present a novel visual analysis approach for data imputation. We develop a multi-party tabular data association strategy that uses intelligent algorithms to identify similar columns and establish column correlations across multiple tables. Then, we perform the initial imputation of incomplete data using correlated data entries from other tables. Additionally, we develop a visual analysis system to refine data imputation candidates. Our interactive system combines the multi-party data imputation approach with expert knowledge, allowing for a better understanding of the relational structure of the data. This significantly enhances the accuracy and efficiency of data imputation, thereby enhancing the quality of data governance and the intrinsic value of data assets. Experimental validation and user surveys demonstrate that this method supports users in verifying and judging the associated columns and similar rows using their domain knowledge.

Key words: Data governance; Data incompleteness; Data imputation; Data visualization; Interactive visual analysis

https://doi.org/10.1631/FITEE.2300480 **CLC number:** TP391.4

1 Introduction

Data imputation is an important data governance technique that aims to fill in missing values. This technique builds the foundation for data analysis modeling (Liu et al., 2010) and value mining. Data incompletion can reduce the overall data quality and impact the reliability of intelligent algorithms and model analysis results, sometimes even leading to completely opposite views and conclusions. For instance, when key information, such as the information related to upstream and downstream customers of a large supply chain enterprise, is missing, this can influence the interpretation of market trends, leading to business losses and commercial risks (Smith, 2003). In the investment and acquisition process, the missing or errors in critical business information can lead to extreme values in asset evaluation, thereby increasing significant investment risks. When information on key variables is missing,

‡ Corresponding author

* Project supported by the Key R&D "Pioneer" Tackling Plan Program of Zhejiang Province, China (No. 2023C01119), the "Ten Thousand Talents Plan" Science and Technology Innovation Leading Talent Program of Zhejiang Province, China (No. 2022R52044), the Major Standardization Pilot Projects for the Digital Economy (Digital Trade Sector) of Zhejiang Province, China (No. SJ-BZ/2023053), and the National Natural Science Foundation of China (No. 62132017)

ORCID: Haiyang ZHU, https://orcid.org/0000-0002-4782-5654; Wei CHEN, https://orcid.org/0000-0002-8365-4741

this can cause analysis conclusions to deviate, misleading researchers and medical staff in their evaluations of the effectiveness and safety of treatment methods, and even lead to legal and liability issues (Bernhard et al., 1998; Gupta and Soeny, 2021; Luo, 2022). Therefore, in the era of rapid development of big data analysis and application, the research on data imputation has garnered widespread attention (Little and Rubin, 2002; Scheffer, 2002; Yang et al., 2021; Enders, 2022; Miao et al., 2023).

Traditional data imputation methods (Kang, 2013) can be loosely categorized as statistical methods (Lajeunesse, 2013; Yin et al., 2014; Yi et al., 2016), machine learning based methods (Marlin, 2008; Emmanuel et al., 2021; Nijman et al., 2022), or deep learning based methods (Chai et al., 2020; Wang HN et al., 2020; Harlim et al., 2021; Kök and Özdemir, 2021), where deep learning is a specialized offshoot of machine learning. They typically assume that the data are singular and isolated, and use statistical or machine learning methods to infer missing values using other rows or other data entries in the same data table. While the aforementioned approaches can to some extent alleviate the issue of data incompleteness, the resulting inferred data values are intricate to trace back to their sources and are imbued with a lack of user trust. Moreover, achieving an optimal balance between accuracy and efficiency remains a challenge. Therefore, there is an urgent need to explore a novel approach for data imputation through visual analysis.

In real-world scenarios, and especially when the amount of data is large, the quality of the collection of practices used to ensure their optimal storage and management may decline over time, concomitant with the rise in the data volume. Many elements of data are stored in the form of files (e.g., Excel files and JSON files) (McCarthy and Graniero, 2006; Palocsay et al., 2010; Raubenheimer, 2017), as distinguished from the use of databases. They do not have explicit primary and foreign keys to build relational databases. Moreover, due to the lack of unified data standards, there are often “system silos” and “data islands,” which result in differences in data forms, fields, time, departments, and other aspects. The data often lack effective correlations with each other. However, there are certain complementary correlations among different data tables. The problem of data incompletion in an isolated data table can be addressed by introducing identical or similar fields from other data tables. Therefore, it is essential to establish correlations between data tables. This involves identifying corresponding data columns representing the same attributes and data rows representing the same entries across different tables. Specifically, the challenges faced are as follows:

1. In large-scale datasets, the abundance of data columns and rows in each table creates challenges in data retrieval. The complexity of data retrieval escalates exponentially with the increasing number of tables, columns, and rows. This poses difficulties in accessing relevant data to fill in the missing values.

2. Due to the heterogeneity of data distribution, types, formats, and structures across various data tables, data columns representing the same attribute can have variations, such as different column names or data distributions. This implies the need for the homogeneity issue of associated data to be effectively addressed, while simple statistical methods may overlook correct correlations.

3. Because of the complexity of data correlations, the workload involved in employing users’ background knowledge to verify data becomes substantial and challenging. Therefore, providing guidance for users through the data imputation process is necessary for improving the efficiency.

To tackle these challenges, we collaborated with data governance professionals from large-scale supply chain enterprises. Through user interviews and requirement surveys focusing on data gaps, we identified and refined the data imputation requirements for different scenarios. Consequently, we propose a multi-data-table imputation method, which establishes correlations among multiple data tables that lack explicit primary and foreign keys to build relational databases, and then imputes missing cells by searching similar rows and columns. We develop an interactive visual analysis system to facilitate the verification of data relations based on users’ domain knowledge. The main contributions of this paper are as follows:

1. We introduce a correlation strategy based on row–column similarity to identify similar data across multi-parity tables, thus facilitating a more precise imputation of missing data.

2. We formulate a multi-party tabular data imputation approach by inferring missing data

using analogous information from correlated tables, thereby effectively addressing the issue of missing data. We then develop a visual system to support interactive data imputation with our approach.

3. Quantitative and qualitative experiments, as well as user surveys, have demonstrated the effectiveness of our approach in supporting the imputation of missing data based on users' domain knowledge.

2 Related works

2.1 Methods for analyzing missing data

Missing data may contain missing, erroneous, misaligned, or otherwise problematic data, which can lead to completely erroneous conclusions in big data analysis tasks (Kim et al., 2003). Missing data can arise at various stages of data generation and processing (Kim et al., 2003). These situations can occur due to factors such as data transmission interruptions, script errors, memory migration, and human errors. Missing data can affect data quality and introduce uncertainty into standard analysis algorithms, thereby posing a common problem that data analysis professionals often need to address. To effectively analyze the data, it is necessary to complete the missing data. This is particularly important in data governance (Kandel et al., 2011; Furche et al., 2016). Typically, data governance professionals inject substitute values for missing data to reduce the negative impact of incomplete data on data analysis and thereby enable such analyses to function appropriately within general data analysis frameworks. We focus on developing a method to impute missing data across multiple data tables. Our approach, based on a row–column similarity correlation strategy, accurately identifies similarities among various tables, resulting in more precise information for data imputation. This method offers a superior and advanced alternative to simple substitution-based approaches, providing enhanced effectiveness in addressing missing data.

Multiple studies (Little and Rubin, 2002; Lajeunesse, 2013) have summarized data imputation methods for replacing or inferring missing values, aiming to make use of other data information as much as possible for the imputation of current data. For instance, hot-deck imputation involves finding substitute values within the available data information for imputation, while code-deck imputation uses other data sources (Githungo et al., 2016) or domain heuristics (Gülensoy et al., 2014) to search for substitute values. Interpolation methods infer missing values through weighted combinations of data points (Gao, 2006), including linear interpolation, regression interpolation, and adaptive interpolation. Data imputation for missing values is a critical factor in enhancing data quality, especially in specific scenarios such as non-uniform sampling in time-series data (Gschwandtner et al., 2012), where it may be necessary to enforce interpolation. However, when joining time series with different granularities, misalignment issues may arise, leading to conflicts with interpolation. Furthermore, leveraging sophisticated related methods allows the integration of data from different stages of data wrangling into a single dataset, enabling targeted computations (Rässler, 2004), and even using relevant techniques like machine learning for comprehensive missing value prediction (Ahuja et al., 2016). These aforementioned methods provide innovative research perspectives for this study, which involve extracting valuable data information from multiple data sources for data imputation purposes. Compared to traditional approaches, our method combines the interplay among multiple data tables, delivering a more accurate and dependable solution for data imputation. This approach can be applied to a broader spectrum of data types, particularly flourishing in intricate settings such as supply chain environments. With insights grounded in practical circumstances, the method offers heightened practicality and applicability.

2.2 Methods for visualizing missing data

Wong and Varga (2012) metaphorically referred to missing data in the field of visualization as "black holes," representing the "dark territories" in the cognitive workspace that remain undisclosed. They argued that it remains unclear how visualization can assist in completing missing data to support users' intuitive cognitive construction. However, it is evident that humans can autonomously make judgments and inferences regarding missing data. Existing visualization systems (Sun et al., 2022; Zhang et al., 2022; Wu ZL et al., 2023) provide support for data quality analysis, including data pre-processing (Bernard et al., 2019), temporal variations in data quality (Bors et al., 2015), and highlighting missing values, incorrect values, or estimated values (Fernstad and Glen, 2014; Bögl et al., 2015; Bors et al., 2017). For instance, the visualization tool Visplause (Arbesser et al., 2017) facilitates quality investigation of time-series data and assists data wrangling professionals in inferring potential causes of missing data. Specifically, in the context of uncertainty visualizations, the presence of missing data represents a notable scenario, as outlined by Bonneau et al. (2014). It involves sampling uncertainties, visualization uncertainties, and modeling uncertainties. Eaton et al. (2005) contended that certain factors, including limited sample size, methodological flaws, and missing data for precise estimation, can collectively give rise to heightened uncertainty surrounding specific data values. This implies that missing data comprise merely one scenario contributing to uncertainty (Griethe and Schumann, 2006). Uncertainties originating from data collection can lead to perceptions of inadequate, excessive, or unreliable information. In datasets characterized by insufficient information due to missing or incomplete instances, various abovementioned visualization methods face significant challenges. Missing data will bring the risk of missing out valuable insights for visual components, and create disagreements about the understanding of the data (Kamal et al., 2021). Our interactive visual analysis system allows users to verify and assess columns and similar rows in the relevant multiple data tables. Users can observe the current progress of completion, as well as the differences and changes over time, and trace the origins of the data. This enhances user comprehensibility and trust, effectively bolstering the data imputation task.

Many visualization systems, such as XGobi (Swayne and Buja, 1998), MANET (Unwin et al., 1996), VIS+AI (Wang XM et al., 2023), and VIM (Templ et al., 2012), provide visual components regarding the distribution of missing data to assist data wrangling professionals in gaining a better understanding of missing data patterns and comparing different imputation methods. Furthermore, specific domain-oriented or data-type-specific visualization analysis systems can automatically handle missing data based on tasks and requirements (Chen et al., 2021). However, it is regrettable that some visualization systems offer little or even no visual representation of imputation data, potentially missing out on the advantages that data visualization techniques can bring. For example, the visualization system proposed by Turkay et al. (2012) replaces missing values with feature averages, while visualization systems in meteorology (Djurcilov and Pang, 1999) and psychology (Gülensoy et al., 2014) employ heuristic methods for imputing missing data based on domain knowledge. Additionally, some visualization systems use visual saliency to draw analysts' attention to missing values. For instance, TimeSearcher (Buono et al., 2005) employs vibrant color coding to indicate missing values, while Restorer (Twiddy et al., 1994) reduces the salience of missing spatial data by using grayscale and represents imputed values using brightness. Therefore, incorporating visual representation of imputed data into visualization systems is crucial. This can help data wrangling professionals gain a better understanding of the data, reduce the potential for misunderstandings and erroneous judgments, and improve the accuracy of data quality and analytical outcomes. Our system provides visual and interactive tools for visualizing the sources of data imputation. This enables users to verify and assess relevant columns and similar rows based on their background knowledge, effectively enhancing the data imputation task.

3 Design requirements

To better understand the functional requirements of users for the data imputation visualization system, we employ various research methods. First, in-depth discussions and research interviews are conducted with three data governance professionals (E_1–E_3) from a large supply chain company. Then, based on specific digital application scenarios, the actual pain points and difficulties are discussed and extracted, and the key technical problems that need to be addressed are identified. These professionals are responsible for collection, development, and analysis of multi-dimensional data from the company's headquarters and more than 400 subsidiary companies. During the construction of the company's supply chain big data center and the advancement of multi-dimensional data collection, random missing data issues frequently occur due to the heterogeneous nature of the digital systems used by member companies. These issues pose significant risks to subsequent data analysis and visualization. To effectively solve this problem, data governance professionals need to collaborate with data operators, business operators, and master data management personnel from different departments and organizations to conduct data auditing and revision. However, due to the large time span of data records and the time cost associated with cross-organizational and cross-departmental communication processes, this process requires a considerable amount of time and effort. Considering that data discrepancies can pose substantial risks to data analysis and even impact the company's operational management and decision-making analysis, it becomes necessary to cautiously and carefully complete or delete data manually to ensure accuracy and consistency. However, due to the inefficiency of traditional data imputation methods, there is an urgent need for data governance professionals to develop an automated algorithm for data imputation that can be intuitively understood. After discussions with data governance professionals, the following functional requirements are summarized and refined:

R1: missing data overview. The system should provide an overview of missing data to display their distribution across different data tables. By providing an overview, users can gain a preliminary understanding of the completeness and reliability of the datasets, allowing them to make an initial assessment of data quality.

R2: intelligent algorithm control. The system should support user-defined parameters to control intelligent imputation strategies, providing parameters that are easy to understand and perceive. Users can adjust these parameters to achieve the desired data imputation effect.

R3: display of imputation content. The system should visually present data imputation results, assisting users in perceiving the basis for completing missing content and indicating the level of certainty in the imputation, and thus supporting users' judgment and investigation.

R4: traceability of imputation basis. The system should have the capability to trace and determine, as well as to allow users to interactively modify, the imputation basis. Additionally, the system should provide clear identification of data sources and markers for imputation results to enable users to understand and evaluate the accuracy of the imputation.

R5: intuitive and familiar visualization and interaction design. This study focuses on the analysis of tabular missing data, targeting users who may lack experience in data visualization. Therefore, the system should provide visual components and interactive means that are easy to understand and use, ensuring user-friendliness for the target audience.

R6: human–machine collaboration. The system workflow should be presented to the users in a visual form, facilitating their understanding of the data imputation process. Users should be able to interactively participate in the workflow, exploring and analyzing the visualizations, and investigating and controlling different steps. User interactions should be fed back into the entire process for iterative updates and analysis, integrating the users' prior knowledge with the automated algorithms.

4 System overview

As shown in Fig. 1, the interactive visual analysis system for missing data imputation discussed in this study consists of three main components: datasets, visual interface, and construction engine.

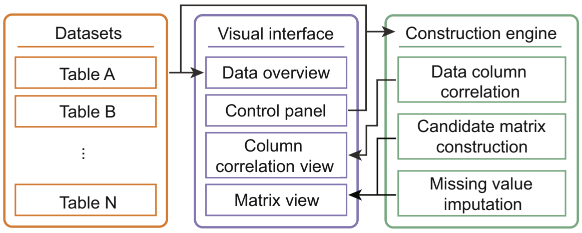


Fig. 1 Overview of our system

Given a set of tabular data, we assume that each data table has a unique primary key v, with no missing primary key information. If primary key values are identical across different data tables, this is an indication of a unique entity. Based on the data overview in the visual interface, users can select the tables that need to be used as data sources for the completion of missing data. Within the control panel, users determine the data imputation strategy, methods, and parameters used in the construction engine. The construction engine contains our data imputation method. It involves identifying similar content in other multi-dimensional tables to complete missing entries. Specifically, the method searches for similar rows and columns in existing tables and uses the discovered content to complete the missing values instead of inferring new values for insertion. As shown in Fig. 2, the proposed data imputation method consists of three main steps: data column correlation, candidate matrix construction, and missing value imputation.

The first step involves the identification of similar columns for correlation in multi-dimensional data tables. It includes column value type determination, column value similarity calculation, column name similarity calculation, and column correlation set determination. The second step involves searching for data similar to the current missing data in other data tables to construct a candidate matrix. It encompasses primary key identification, candidate row construction, candidate column construction, and candidate matrix construction. The third step combines intelligent algorithms and users' prior knowledge for data imputation. Once the missing information is determined, the engine will re-correlate data columns and perform iterative calculations.

After the construction engine execution is completed, the column correlation view displays the associated columns identified by the construction engine in multi-dimensional data tables, allowing users to understand, analyze, and interactively revise them based on prior knowledge. In the matrix view, users can observe the candidate matrix computed by the construction engine and select different strategies for intelligent imputation. Users can also interactively select multiple recommended values from the construction engine for imputation based on their own experience.

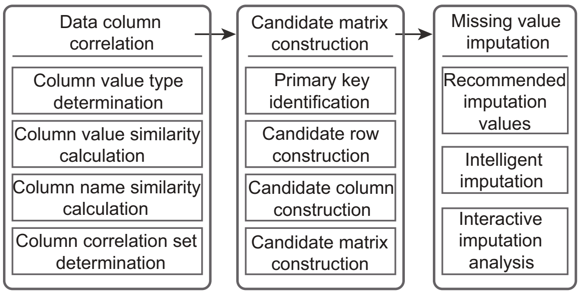

Data column correlation
Column value type determination
Column value similarity calculation
Column name similarity calculation
Column correlation set determination
Candidate matrix construction
Primary key identification
Candidate row construction
Candidate column construction
Candidate matrix construction
Missing value imputation
Recommended imputation values
Intelligent imputation
Interactive imputation analysis

Fig. 2 Overview of our method

5 Method

Before introducing our data imputation method, we first define the mathematical symbols used in this study (Table 1).

5.1 Data column correlation

The purpose of data column correlation is to compute a column correlation set C^{R} for a given set of data tables, where each correlation group $\left(c_i^j, c_p^q\right) \in C^{\mathrm{R}}$ represents the relations between two data columns $\left(c_i^j, c_p^q\right)$ from different tables.

1. Column value type determination. Our method first assesses the data type of each data column's values cv in all data tables, encompassing four types (identifier, categorical, numerical, and textual). The detection process is presented in Algorithm 1. When the unique ratio is smaller than ϵ, Algorithm 1 regards the data column as categorical. Here, ϵ is empirically set to 0.1. However, when it is difficult to determine the data type, users' prior knowledge is needed to make a judgment. For instance, it is hard to determine the type of a column value like "10002."

2. Column value similarity calculation. Our method computes similarities among column values

Table 1 Definition of symbols

Notation	Description
$T_i \in T$	The i^{th} data table in dataset T
$T_i = (C_i, R_i)$	All columns (C_i) and all rows (R_i) of data table T_i
$c_i^j \in C_i$	The j^{th} column of data table T_i
$\left(\mathrm{cn}_i^j, \mathrm{cv}_i^j\right) = c_i^j$	Column c_i^j includes the column name $\left(\mathrm{cn}_i^j\right)$ and the value list $\left(\mathrm{cv}_i^j\right)$
$\mathrm{cv}_i^j(v)$	Each value in value list cv_i^j of data column c_i^j
$r_i^j \in R_i$	The j^{th} row of data table T_i
C^{R}	Set of column correlations
$\left(c_i^j, c_p^q\right) \in C^{\mathrm{R}}$	Correlations between the j^{th} column c_i^j in data table T_i and the q^{th} column c_p^q in data table T_p
$m\left(c_i^j, c_p^q\right) \in M_{\mathrm{c}}$	The similarity between data columns c_i^j and c_p^q is stored in the similarity matrix M_{c}
$m\left(\mathrm{cv}_i^j, \mathrm{cv}_p^q\right) \in M_{\mathrm{cv}}$	The column value similarity between cv_i^j and cv_p^q is stored in the matrix M_{cv}
$m\left(\mathrm{cn}_i^j, \mathrm{cn}_p^q\right) \in M_{\mathrm{cn}}$	The column value similarity between cn_i^j and cn_p^q is stored in the matrix M_{cn}

cv that belong to different data tables but share the same data type, resulting in a matrix of column value similarities M_{cv}. The process for similarity calculation is presented in Algorithm 2. Each element $M[a,b]$ represents the similarity between c_i^j (in the a^{th} column) and c_p^q (in the b^{th} column) across all data columns. Our approach employs different similarity measures for various data types. We calculate the similarity for identifier type and categorical type data using the Jaccard index. The similarity of textual type data is calculated by word mover's distance (WMD) (Kusner et al., 2015), which is commonly used in natural language processing (NLP) (Huang et al., 2016; Wu LF et al., 2018). The similarity of numerical type data can be measured by distribution similarity metrics. Our method uses earth mover's distance (EMD) by default to ensure the consistency with the WMD of textual-type data, while also supporting other metrics, such as the Kullback–Leibler (KL) divergence.

3. Column name similarity calculation. To construct the column name similarity matrix M_{cn}, the textual data similarity calculation method discussed previously is employed for the column names. The process for similarity calculation is presented in Algorithm 2. Matrix M_{cn} can be obtained when c=cn.

Algorithm 1 Column value type determination

```
function INFERCOLUMNTYPE(cv)
    uniqueValues ← getUniqueValues(cv)
    uniqueRatio ← count(uniqueValues)/count(cv)
    if uniqueRatio=1 then
        return "Identifier Type"
    else if uniqueRatio< $\epsilon$ then
        return "Categorical Type"
    else if allNumerical(uniqueValues) then
        return "Numerical Type"
    else if allText(uniqueValues) then
        return "Textual Type"
    end if
end function
```

Algorithm 2 Column value/name similarity calculation

```
function COMPUTECOLUMNSIMILARITY($c$, tables)
    $M$ ← empty matrix
    for $T_i, T_p \in$ tables do
        for $c_i^j \in T_i$, $c_p^q \in T_p$ do
            $a$ ← getIndexInAllColumns($c_i^j$)
            $b$ ← getIndexInAllColumns($c_p^q$)
            $M[a,b]$ ← computeSimilarity($c_i^j$, $c_p^q$)
        end for
    end for
    return $M$
end function
```

4. Column correlation set determination. For data columns from different data tables, our method combines the column value similarity matrix M_{cv} and the column name similarity matrix M_{cn} to calculate the overall similarity matrix M_{c}. Therefore,

$$M_{\mathrm{c}}[a,b]=M_{\mathrm{cv}}[a,b]+M_{\mathrm{cn}}[a,b], \tag{1}$$

where a and b are both positive integers satisfying $1\leq a\leq N$ and $1\leq b\leq N$ with N being the number of data columns. The process of determining the column correlation set is two-parted: (1) it first finds pairs of data columns, whose similarities exceed the threshold; (2) it builds a disjoint-set data structure to facilitate subsequent searches for the most similar columns. We provide three strategies for selecting correlated data columns:

Strategy 1 C^{R} is directly obtained using column correlation set determination, based on the user-defined or default threshold. For example, it may result in the following situation:

$$\left(c_i^1, c_j^7, c_j^8\right)\in C^{\mathrm{R}}. \tag{2}$$

That is, the similarity between data column c_i^1 in data table T_i and data columns c_j^7 and c_j^8 in data table T_j is greater than the threshold.

Strategy 2 The disjoint-set rule is modified to correlate the two most similar columns $\left(c_i^j, c_p^q, i\neq p\right)$ from different data tables T, followed by adding them to the column correlation set. The similarity must be greater than the user-defined or default threshold to ensure validity. This makes c_i^j and c_p^q correlate uniquely to each other in the corresponding tables T_i and T_p. Under this strategy, Eq. (2) would become

$$\left(c_i^1, c_j^7\right)\in C^{\mathrm{R}}. \tag{3}$$

That is, the similarity between data column c_i^1 in data table T_i and data column c_j^7 in data table T_j is greater than the similarity of either of them with data column c_j^8, which leads to the correlation of (c_i^1, c_j^7) in the final column correlation set.

Strategy 3 The disjoint-set rule is partially adjusted. While ensuring that the similarity is greater than the user-defined or default threshold, we begin by identifying the data column c_i^j $(i\neq 1)$ most similar to the first data column c_1^1. This results in a column correlation set $\left(c_1^1, c_i^j\right)$. Subsequently, we identify the column with the highest similarity to all columns in the existing column correlation set; therefore,

$$c_p^q \in C, \quad p \neq i \neq 1. \tag{4}$$

That is, we identify the column with the maximum value for $\text{avg}\Big(\text{sim}\left(c_1^1, c_p^q\right) + \text{sim}\left(c_i^j, c_p^q\right)\Big)$. For instance, suppose that the current set of column correlations is $\left(c_1^1, c_j^3\right)$, and there exist columns $\left(c_j^1, c_j^3\right)$ in data table T_j that have a similarity greater than the user-defined or default threshold. In such a case, we compare the average similarities $\text{avg}\Big(\text{sim}\left(c_1^1, c_j^1\right) + \text{sim}\left(c_i^3, c_j^1\right)\Big)$ and $\text{avg}\Big(\text{sim}\left(c_1^1, c_j^3\right) + \text{sim}\left(c_i^3, c_j^3\right)\Big)$. Finally, we add the column with the higher average similarity to the existing set of column correlations.

5.2 Candidate matrix construction

The purpose of constructing the candidate matrix is to integrate multiple data tables based on the correlated column information between them, to support interactive imputation of missing values for users. Ideally, the candidate matrix constructed using the method proposed in the present study should be unique, meaning that each missing data point can be imputed using the corresponding values from rows and columns in other data tables. The process of candidate matrix construction is described in Algorithm 3.

However, the intelligent algorithm has two limitations. First, if there are multiple highly similar correlated columns, the algorithm will select only the most similar one, neglecting other columns with high similarity. This contradicts with the original intention of finding missing data from multi-dimensional data tables. Therefore, all highly similar columns should be displayed, and user knowledge and experience should be used to understand and select them. Second, for columns with low similarity, the algorithm does not establish any correlation, resulting in certain missing elements of data having no values for imputation. Additionally, there are many strategies and parameters that need to be manually adjusted and controlled in the column correlation step, which requires users to have a preliminary understanding and comprehension of the data imputation results. The proposed method generates multiple candidate matrices and uses subsequent visual interactions to assist users in exploration, analysis, understanding, and decision-making. The matrix construction process is illustrated in Fig. 3.

Algorithm 3 Candidate matrix construction

```
function FILLMISSINGDATA(missingData, tables, columnCorrelations)
    for all $m_i \in$ missingData do
        primaryKey ← findPrimaryKey($m_i$)
        candidateRows ← searchCandidateRows(primaryKey, tables)
        candidateColumns ← mergeCorrelationsColumns(candidateRows, columnCorrelations)
        candidateMatrices ← buildCandidateMatrices(candidateRows, candidateColumns)
        filledData$_i$ ← interactivelyFillMissingData($m_i$, candidateMatrices)
        updateTables(filledData$_i$, tables)
    end for
    return updated tables
end function
```

1. Primary key identification. As shown in Fig. 3a, we locate the primary key v in the row containing missing data. By locating the primary key in all rows with missing data, we can obtain the set

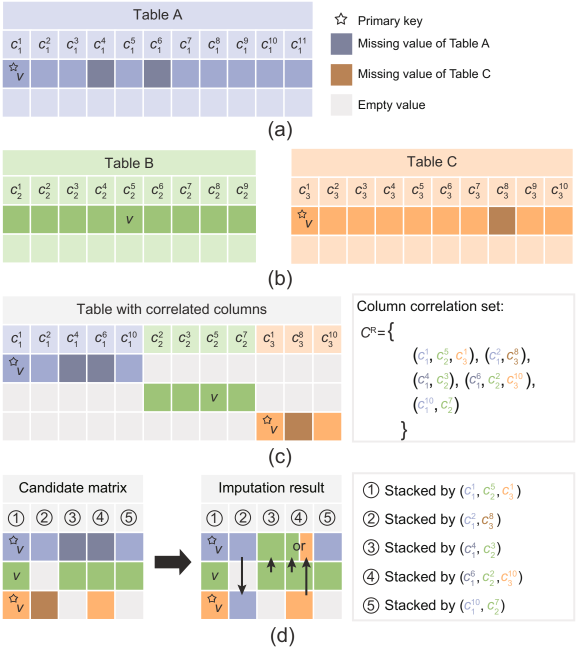


Table A
c_1^1 c_1^2 c_1^3 c_1^4 c_1^5 c_1^6 c_1^7 c_1^8 c_1^9 c_1^{10} c_1^{11}
v
☆ Primary key
Missing value of Table A
Missing value of Table C
Empty value
(a)
Table B
c_2^1 c_2^2 c_2^3 c_2^4 c_2^5 c_2^6 c_2^7 c_2^8 c_2^9
v
Table C
c_3^1 c_3^2 c_3^3 c_3^4 c_3^5 c_3^6 c_3^7 c_3^8 c_3^9 c_3^{10}
v
(b)
Table with correlated columns
c_1^1 c_1^2 c_1^4 c_1^6 c_1^{10} c_2^2 c_2^3 c_2^5 c_2^7 c_3^1 c_3^8 c_3^{10}
v
v
v
Column correlation set:
$C^R=\{ (c_1^1, c_2^5, c_3^1), (c_1^2, c_3^8), (c_1^4, c_2^3), (c_1^6, c_2^2, c_3^{10}), (c_1^{10}, c_2^7) \}$
(c)
Candidate matrix
① ② ③ ④ ⑤
v
v
v
Imputation result
① ② ③ ④ ⑤
or
v
v
v
① Stacked by (c_1^1, c_2^5, c_3^1)
② Stacked by (c_1^2, c_3^8)
③ Stacked by (c_1^4, c_2^3)
④ Stacked by (c_1^6, c_2^2, c_3^{10})
⑤ Stacked by (c_1^{10}, c_2^7)
(d)

Fig. 3 Candidate matrix construction process: (a) primary key identification; (b) candidate row construction; (c) candidate column construction; (d) candidate matrix construction and data imputation with recommended content

of primary key values K. Typically, the primary key is a column of identifiers, such as names, addresses, and IDs.

2. Candidate row construction. For each key value $v_i \in K$, we search for data rows in different data tables that contain v_i and retrieve the corresponding row data to construct candidate rows. As shown in Fig. 3b, we search for data rows containing v in different data tables, which are highlighted, to form candidate rows.

3. Candidate column construction. Based on the column correlation set C^{R}, we extract and merge the correlated columns in the candidate rows. As shown in Fig. 3c, the columns contained in C^{R} are the candidate columns. Our proposed method extracts these columns from their original data tables and combines them to create a new data table.

4. Candidate matrix construction. As shown in Fig. 3d, our method further merges the new data table generated in step 3. The column correlation set C^{R} contains multiple groups of correlated columns. For each group of correlated columns, the method stacks the candidate row information of the correlated columns into a single column. For example, column ② is created by stacking the row information of columns c_1^2 and c_3^8. To ensure the completeness of the matrix, the row information from Data Table B is set to empty in that column. By iterating through each group of correlated columns in C^{R}, the construction of the candidate matrix is completed. Particularly, when building the candidate matrix based on candidate rows and candidate columns, there can be multiple possibilities due to the existence of multiple column correlations in C^{R}.

5.3 Missing value imputation

1. Recommended imputation values. The resulting candidate matrix is the recommended outcome for filling in missing data. Each column in the candidate matrix is formed by stacking the candidate row information from several correlated columns. This means that the row information within each column exhibits high similarity, allowing for a mutual imputation of missing information, i.e., the recommended value for imputation. As shown in Fig. 3d, in column ④, there are two recommended imputation values for filling in missing data from Data Table A. The method supports intelligent imputation or manual selection, and it can automatically update the column correlation set C^{R} and rebuild a new candidate matrix based on the new correlations.

2. Intelligent imputation. For multiple candidate matrices, this method provides an intelligent imputation strategy. It chooses the matrix with the highest overall similarity in candidate columns as the imputation-recommended value. Users can determine whether to keep the automatically recommended values through interactive analysis.

3. Interactive imputation and analysis. Users can determine the imputation of missing content based on the source of the candidate matrix construction. This means that users can establish more accurate column correlation information and decide whether to update the entire engine. If users choose to update, they can use the determined column correlation information as the initial default value. They can update the column similarity matrix using Algorithm 2 and further recalculate the column correlation set C^{R}, followed by subsequent matrix construction.

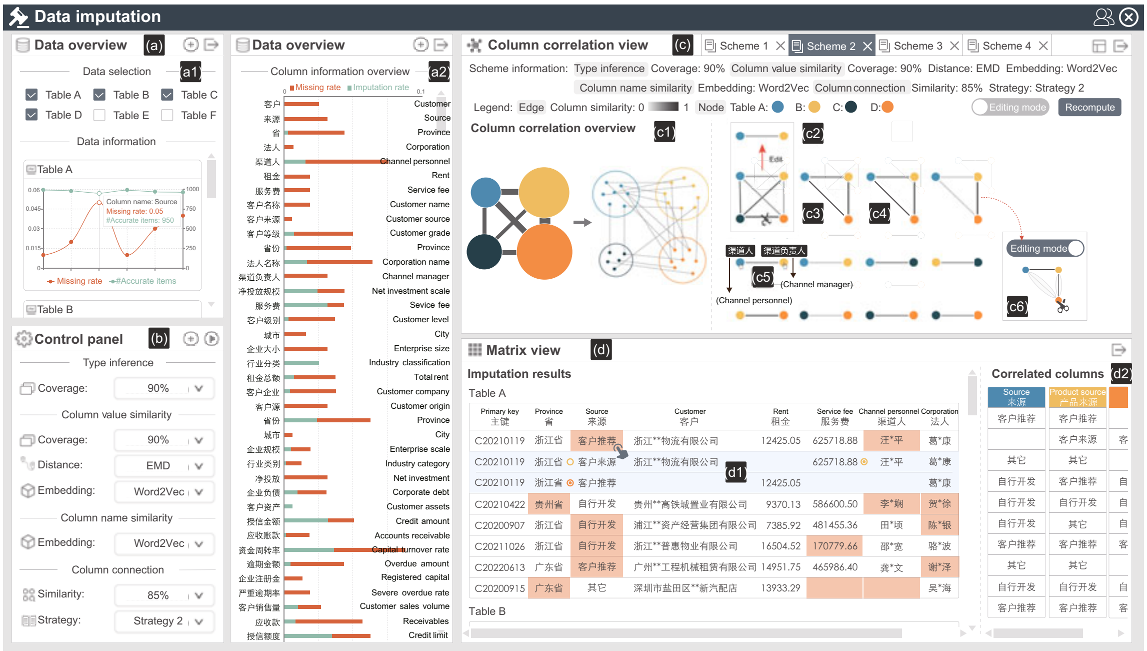
6 Visual interface

The data imputation visual analysis system consists of four main components: data overview, control panel, column correlation view, and matrix view (Fig. 4).

1. Data overview (Figs. 4a1 and 4a2) helps users understand the characteristics of the data and identify missing data. It is divided into three parts: (1) data selection (Fig. 4a1) allows users to choose the data tables for analysis or import databases or local data using the "⊕" button; (2) data information visualizes the missing rates of different columns in each data table and the number of normal data entries using line charts; (3) column information overview (Fig. 4a2) uses a stacked histogram to present the missing rate of each column and the current imputation progress.

Interaction: In data selection, users can select different data tables for analysis and correlation. In data information, users can click on the points in the line chart to view specific information about the current column, and the corresponding column will be highlighted in other views.

2. Control panel (Fig. 4b) is designed for configuring the column value type determination, column value similarity calculation, and column name



Data selection: ☑ Table A ☑ Table B ☑ Table C ☑ Table D □ Table E □ Table F

Column information overview: 客户 Customer; 来源 Source; 省 Province; 法人 Corporation; 渠道人 Channel personnel; 租金 Rent; 服务费 Service fee; 客户名称 Customer name; 客户来源 Customer source; 客户等级 Customer grade; 省份 Province; 法人名称 Corporation name; 渠道负责人 Channel manager; 净投放规模 Net investment scale; 服务费 Sevice fee; 客户级别 Customer level; 城市 City; 企业大小 Enterprise size; 行业分类 Industry classification; 租金总额 Total rent; 客户企业 Customer company; 客户源 Customer origin; 省份 Province; 城市 City; 企业规模 Enterprise scale; 行业类别 Industry category; 净投放 Net investment; 企业负债 Corporate debt; 客户资产 Customer assets; 授信金额 Credit amount; 应收账款 Accounts receivable; 资金周转率 Capital turnover rate; 逾期金额 Overdue amount; 企业注册金 Registered capital; 严重逾期率 Severe overdue rate; 客户销售量 Customer sales volume; 应收款 Receivables; 授信额度 Credit limit

Control panel

Type inference
Coverage: 90%

Column value similarity
Coverage: 90%
Distance: EMD
Embedding: Word2Vec

Column name similarity
Embedding: Word2Vec

Column connection
Similarity: 85%
Strategy: Strategy 2

Scheme information: Type inference Coverage: 90% Column value similarity Coverage: 90% Distance: EMD Embedding: Word2Vec Column name similarity Embedding: Word2Vec Column connection Similarity: 85% Strategy: Strategy 2

Imputation results

Table A

Primary key 主键	Province 省	Source 来源	Customer 客户	Rent 租金	Service fee 服务费	Channel personnel 渠道人	Corporation 法人
C20210119	浙江省	客户推荐	浙江**物流有限公司	12425.05	625718.88	汪*平	葛*康
C20210119	浙江省	○ 客户来源	浙江**物流有限公司		625718.88 ◎	汪*平	葛*康
C20210119	浙江省	◉ 客户推荐		12425.05			葛*康
C20210422	贵州省	自行开发	贵州**高铁城置业有限公司	9370.13	586600.50	李*娴	贺*徐
C20200907	浙江省	自行开发	浦江**资产经营集团有限公司	7385.92	481455.36	田*顷	陈*银
C20211026	浙江省	自行开发	浙江**普惠物业有限公司	16504.52	170779.66	邵*宽	骆*波
C20220613	广东省	客户推荐	广州**工程机械租赁有限公司	14951.75	465986.40	龚*文	谢*泽
C20200915	广东省	其它	深圳市盐田区**新汽配店	13933.29			吴*海

Table B

Correlated columns

Source 来源	Product source 产品来源	
客户推荐	客户推荐	
	客户来源	客
其它	其它	
自行开发	客户推荐	自
自行开发	自行开发	自
自行开发	其它	自
客户推荐	客户推荐	客
其它	其它	
自行开发	自行开发	自
客户推荐	客户推荐	客

Fig. 4 Visual interface: (a) data overview, used for data quality investigation and offering a column information overview; (b) control panel, allowing users to adjust the methods, strategies, and relevant parameters for column correlations; (c) column correlation view, providing insights into data column correlations and enabling user-driven interactive modification and data relation reconstruction by the user; (d) matrix view, presenting candidate matrix information, recommended imputation values for missing data, and their sources. It also supports interactive selection of recommended imputation values for filling in missing data. References to color refer to the online version of this figure

similarity calculation, as well as for selecting similarity measures (such as EMD and Word2Vec), choosing strategies (e.g., Strategies 1–3), and adjusting relevant parameters (such as coverage rate and similarity) when processing column correlation.

Interaction: Users can select from a variety of algorithms and parameters provided by the system through dropdown menus. They can also import other files, models, or custom algorithms by clicking the "⊕" button in the top right corner. After clicking the "Run" button in the top right corner, the results under the current scheme will be displayed in the column correlation view. Users can set parameters to create different schemes and switch and compare results in the column correlation view.

3. Column correlation view (Fig. 4c) helps users understand the correlations between columns in different data tables. It displays results for different schemes and supports switching among them. It also enables interactive modification and reconstruction of data relations by the user. The top of Fig. 4c displays tabs for different schemes and shows the specific results corresponding to each scheme, providing users with references for adjusting schemes and parameters in the control panel. Column correlation overview (Fig. 4c1) represents the number of columns using circle sizes, and the width of the links indicates the number of column correlations between data tables. Node-link diagram (Fig. 4c2) supports analysis of the column correlations, where each node represents a column from a data table, and each link represents the correlation between two columns. The color of the circles represents different data tables, and the color of the links represents the similarity between the columns.

Interaction: In the column correlation view, users can click on the "Tabs" scheme to observe, compare, and delete results from different schemes. Clicking on a circle (depicted in Fig. 4c2) highlights the corresponding column details (depicted in Figs. 4a1 and 4a2). In the column correlation overview (Fig. 5a), users can click on circles to expand and observe the column correlation details between different data tables (Fig. 5b). When hovering over a circle, relevant edges are highlighted, and the column name is displayed. The editing mode

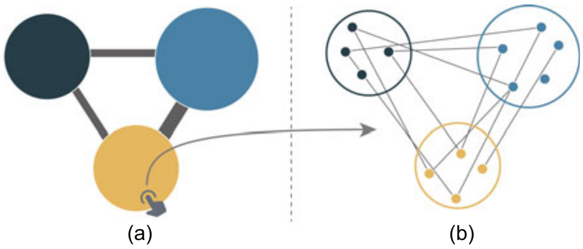


Fig. 5 Column correlation overview and detailed information: (a) column correlation overview before expansion; (b) column correlation detail information between different data tables after expansion

button (Fig. 4c6) allows users to modify column correlation information. Clicking the "Recompute" button triggers the update of results in various views.

4. Matrix view (Fig. 4d) displays recommended values for data imputation along with information in the candidate matrix and sources of the recommended imputation values. Users can select recommended imputation values to fill in missing cells (highlighted in red). Clicking on a missing cell shows the corresponding candidate matrix (Fig. 4d1). The color of the buttons representing the recommended imputation values indicates the source data table, allowing users to make informed choices. Additionally, the first 10 rows of the associated columns' data are displayed in Fig. 4d2. The color of the column headers represents the corresponding data table, enabling users to assess the similarity of the associated columns.

Interaction: When users click on a cell with a red background in the data imputation result table, the correlation column information will be displayed, as indicated in Fig. 4d2; at the same time, the corresponding columns are highlighted, as indicated in Figs. 4a1, 4a2, and 4c2. If users discover that the column correlations are not valid while examining the presented correlation column information in Fig. 4d2, they can click on the column header, and the corresponding column will be highlighted in Fig. 4c2. To make corrections to the column correlations, users can access the editing mode and make the necessary modifications.

7 Evaluation

7.1 Case study

E_1, an expert in data governance from a large supply chain company, had the task of completing four data tables with missing values. These tables contained information pertaining to the company's customers, with customer ID as the unique primary key. Each data table consisted of 2500 rows. After importing the tables into the system, E_1 observed from data overview that every column in the tables, except for the primary key, had varying degrees of missing values. On further examination of the column information overview, E_1 discovered that Data Table A had the highest missing value rates for columns such as source, channel personnel, and service fee. Consequently, E_1 decided to prioritize the imputation of Data Table A.

To configure the column connection strategy in the control panel (Fig. 4b), E_1 selected Strategy 2 and set the similarity threshold to 85%. These settings ensured comprehensive consideration of all potential correlation columns. It was also ensured that only one correlation column from a data table could be chosen for each given column, reducing visual clutter and cognitive load. Following the clicking of the "Run" button, the column correlation view displayed the calculated column correlations based on this scheme. By expanding the circles in the column correlation overview (Fig. 4c1), E_1 observed that Data Table A and several other data tables had multiple correlation columns (Fig. 4c1). Further analysis in the column correlation details revealed a total of 12 node-link diagrams of column correlations, each containing 2–4 nodes, representing distinct column correlations (Fig. 4c2). To validate the results preliminarily, E_1 clicked on the circles in one of the node-link diagrams of column correlations to highlight the corresponding column information in the data overview (Fig. 4a2). These columns represented customer, customer name, and customer company, which had identical meanings and aligned with prior knowledge (Fig. 4c3). As a result, E_1 began to develop trust in the automated algorithm and intended to use it further for imputing missing data.

In the matrix view (Fig. 4d), E_1 focused on the imputation results of Data Table A. It was observed that all 104 missing values in the source column had been successfully completed. Among these, 48 values were originated from the customer source column in Data Table B, and the remaining 56 values were originated from the customer origin column in Data Table D (Fig. 4c4). By clicking on a cell with a completed value (red background) in the source column in Data Table A (Fig. 4d1), E_1 compared it with its corresponding correlation column information and confirmed their similarity in light of prior knowledge. Continuing the analysis, E_1 discovered that the channel personnel column was correlated with only the channel manager column in Data Table B (Fig. 4c5) by hovering over the nodes. However, 21 missing values in this column could not be completed due to the absence of corresponding primary keys in Data Table B. This highlighted the necessity for additional data sources to enhance the imputation process.

Regarding the column named service fee, it was found to have correlations with the service fee column in Data Table B, the total rent column in Data Table C, and the net investment column in Data Table D (Fig. 4c2). However, on careful examination of the suggested imputation results, E_1 observed that values in the service fee column deviated significantly from the expected range. Consequently, E_1 manually re-selected the imputation results. On analyzing the information provided by the correlated columns, E_1 recognized that the relationships of the total rent and net investment columns with the service fee column were incorrect. These two columns actually represented entirely different concepts at the business level. It appeared that the automated algorithm focused solely on numerical similarity while disregarding the actual context. Consequently, E_1 decided to eliminate these correlations between the columns. E_1 clicked on the corresponding column headers in the column correlation view to highlight them and entered the column correlation editing mode. E_1 removed the correlations between the columns in the node-link diagram. After rerunning the modified scheme, E_1 was satisfied with the updated imputation results for Data Table A.

Following a similar operating procedure, E_1 proceeded to complete the remaining three data tables. E_1 believed that the system can automatically associate data columns that may be related, allowing users to select imputation values based on their expertise and also to adjust column correlations, which greatly improved the efficiency of data imputation.

7.2 Quantitative experiment

This experiment evaluates the accuracy of our imputation algorithm by comparison with state-of-the-art data imputation algorithms using their available implementations.

1. Datasets. To verify the effectiveness of these algorithms in real-world production environments, we used supply chain datasets from a large supply chain enterprise. The datasets are composed of six tables containing supply-chain customer information (e.g., name, unified social credit code, date of incorporation, type, registration authority, and status), including numerical, categorical, and textual data. The datasets have a total of 125 columns and 9987 rows, with 1693 missing data items. Several data governance professionals from the company imputed the missing items using their experience, and we refer to these imputations as the ground truth.

2. Algorithms. Because there are few imputation algorithms that work on textual data or have public implementations, we focused mainly on numerical data imputation algorithms, including KNNimpute (Troyanskaya et al., 2001), MissForest (Stekhoven and Bühlmann, 2012), IterativeImputer (Pedregosa et al., 2011), Soft-Impute (Mazumder et al., 2010), multivariate imputation by chained equations (MICE) (Azur et al., 2011), and SimpleFill (Rubinsteyn and Feldman, 2016). All algorithms are configured by default.

3. Results. We used the normalized mean squared error to report the results, which are indicated in Fig. 6. To compare the efficacies of different algorithms in imputing categorical data, we transformed the categorical data into non-negative integers. Table 2 presents the proportion of correctly imputed data by different methods of imputation of different data types, where "correctly imputed" means that the imputed value is exactly the same as the ground truth.

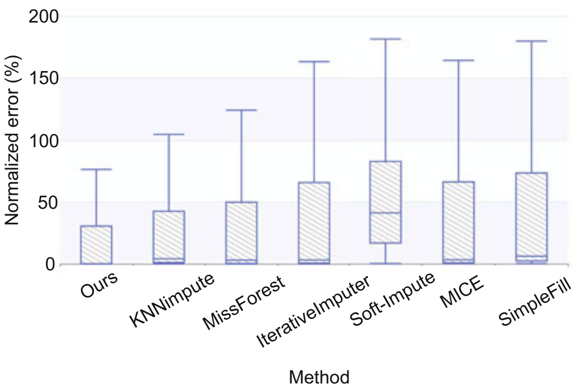


Fig. 6 Normalized error distribution of results by different imputation methods

7.3 User studies

1. Participants and data. To further evaluate the effectiveness of the method and system, we conducted a user study. The survey participants consisted of 3 data governance professionals from a large supply chain company, as well as 7 graduate students and 10 undergraduate students majoring in computer science and technology from a university. We used the data from the previous case study.

2. Tasks and procedure. Participants were initially provided with a detailed explanation of the system, including the meanings of different views and their interactive functionalities. We then presented some foundational questions and their corresponding answers. After becoming familiar with the system, participants were asked to complete a series of tasks (Table 3) designed for each view.

After completing the imputation task, participants were invited to freely explore the system. Subsequently, they rated their system experience using a 5-point Likert scale (Table 4).

3. Passing rates. Each participant spent approximately 15–20 min on task imputation, and the average passing rates for all tasks are shown in Fig. 7. The ratings for the system are presented in Fig. 8.

4. Results. Data overview: During user interactions, we observed that the data overview feature was not widely used by users as a comprehensive tool, contrary to our initial expectations. For instance, in tasks involving data selection, users seldomly engaged with this functionality. Instead, users tended to treat data information and column information as navigation aids rather than focusing on gaining insights from the visualizations. This discrepancy between our initial design intent and real-world usage patterns underscores the importance of further exploring user needs regarding overviews and navigation. In our forthcoming research, we plan to delve deeper into understanding user expectations for overviews and navigation, aiming to devise more user-centric visual interactive methods. By conducting more comprehensive requirement analyses, we expect to better create visualizations that effectively cater to users' practical usage scenarios.

Column correlation view and matrix view: In our study, we observed that the column correlation view and matrix view features received relatively low scores from users in terms of task completion and feedback, especially among non-expert users. This suggests that these visualization techniques might

Table 2 Proportion of correctly imputed data by different methods

Method	Proportion of correctly imputed data (%)		
	Numerical	Categorical	Textual
Ours	**91.92**	**95.95**	82.39
KNNimpute	23.45	53.15	N/A
MissForest	37.80	71.56	N/A
IterativeImputer	30.41	55.05	N/A
Soft-Impute	1.83	36.03	N/A
MICE	31.25	55.34	N/A
SimpleFill	10.15	44.45	N/A

Bold values indicate the optimal results; N/A indicates that the algorithm does not support imputation for the corresponding data type

Table 3 Evaluation tasks of the user study

No.	Task	Requirement(s)
T1	Find the data table with the highest missing rate	R1
T2	Find the data field with the highest missing rate	R1
T3	Combine different data tables	R2
T4	Use different column correlation schemes	R2, R6
T5	Describe the recommended imputation results	R3
T6	Explain the basis of the imputation results	R4
T7	Edit column correlations	R3, R6
T8	Complete missing data values	R3, R6

Table 4 The questionnaire of the user study

No.	Question
Q1	Is the system interface intuitive and easy to use?
Q2	Is the system interface easy to learn?
Q3	Is the interface of the data overview intuitive and easy to use?
Q4	Does the interface of the data overview help you gain insight into the data?
Q5	Is the interface of the control panel intuitive and easy to use?
Q6	Does the interface of the control panel help you adjust strategies, methods, and parameters?
Q7	Is the interface of the column correlation view intuitive and easy to use?
Q8	Is the interaction of the column correlation view intuitive and easy to use?
Q9	Does the column correlation view help you analyze row connections?
Q10	Is the interface of the matrix view intuitive and easy to use?
Q11	Is the interaction of the matrix view intuitive and easy to use?
Q12	Does the matrix view help you handle missing values?

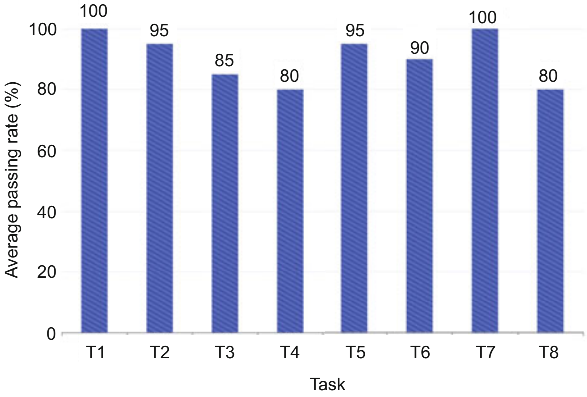


Fig. 7 Results of tasks

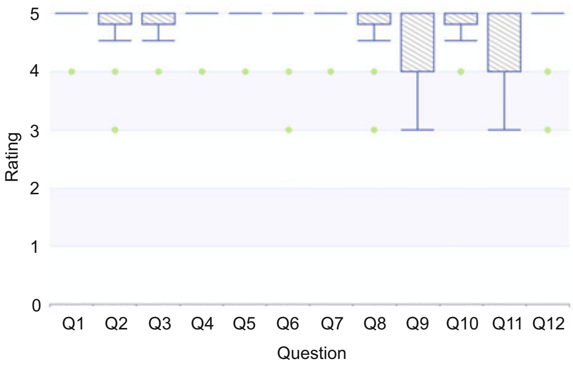


Fig. 8 Results of the questionnaire

not be immediately intuitive for users and might require a significant learning curve. Despite these visualization techniques are quite common in various contexts, our study indicates that they may demand additional time for users to comprehend and explore. Nevertheless, it is encouraging to note that when users observed improvements in the results, their attitudes toward these visualizations tended to become more positive. Therefore, in our future design endeavors, we need to strike a balance between the learning curve and efficiency of the visualization. We can achieve this by offering clearer, more intuitive explanations and guidance, helping users better grasp these advanced visualization tools.

In summary, participants positively assessed the usability and user-friendliness of our system, highlighting its visual design, ease of use, and convenience. The system effectively fulfilled user requirements and successfully achieved the desired outcomes. Participants successfully completed tasks, obtaining the required data effortlessly. The task design aligned with expectations for data imputation, demonstrating the system's effectiveness. Furthermore, participants independently discovered additional features such as linkage and interactivity, enhancing data analysis efficiency and providing valuable insights. Finally, the ratings provided for the system by the participants indicated their levels of satisfaction with its visualization, interactivity, usability, and reliability. Our user study has provided us with insights into user behavior and feedback, steering our design and enhancement strategies toward better addressing user needs. Our system delivers a favorable user experience and serves as a valuable tool for data exploration and analysis. Leveraging these insights, we look forward to continually refining our visualization system to make it more adept at accommodating users' real-world usage scenarios.

8 Discussion

1. Human–machine collaboration. We leverage the similarities between column headers and values to identify correlation columns and construct candidate matrices for the intelligent imputation of missing data. We then engage data governance experts to refine the final results based on their prior knowledge. Our experts acknowledged that this approach enables the combination of machine intelligence and human expertise. Fully automated algorithms typically enhance overall efficiency but sacrifice some effectiveness by ignoring the semantic meanings of certain columns. By allowing experts to trace the process of filling missing value and interactively adjust column correlation schemes, the accuracy and reliability of data imputation can be significantly improved. However, our experts expressed concerns about incorrect or unfilled missing values arising consequent to inadequate data sources, necessitating additional effort for verification. We believe that incorporating supplementary data imputation methods, such as prediction, to address such cases holds promise for future research.

2. Visual interface. The visual analysis system received favorable feedback from our experts regarding its ability to facilitate the exploration and imputation of missing data, from an overview to a detailed level. Particularly noteworthy was the node-link diagrams, which effectively represented column correlations and simplified the analysis and updating of various schemes. Two experts highlighted the importance of focusing on correlations with higher

similarities, which are visually indicated by links with deeper colors. Furthermore, the system supports iterative editing of column correlations and re-computation for improved results through intuitive interactions, eliminating the need for time-consuming manual checks. These features significantly reduced the workload of the experts. Therefore, we demonstrate the practicality and effectiveness of employing straightforward visual designs to enhance data imputation.

3. Limitations. Two limitations characterize the present research. First, the proposed algorithm currently relies on the presence of a single primary key for each table to ensure simplicity, which may not always be feasible in real-world scenarios. However, it is possible to identify candidate rows based on similarities between rows, rather than solely relying on primary keys. Consequently, our approach can be easily extended to accommodate a wider range of diverse and complex scenarios. The other limitation pertains to scalability, seeing that the algorithm and system were evaluated using only four tables, encompassing a small number of rows and columns. Although the algorithm theoretically should function adequately when handling more tables, rows, and columns, the visual analysis system might encounter performance issues due to excessive nodes and links, resulting in visual clutter and increased cognitive load for users. Therefore, further research is necessary to address these concerns.

9 Conclusions

This paper presents a data imputation method based on a multi-party table data association strategy and constructs an efficient interactive data imputation visual analysis system to assist data governance professionals in addressing the accuracy and effectiveness issues prevailing in relation to missing values. The system uses existing data information from multi-dimensional data tables for imputation. It builds a candidate matrix from the multi-dimensional data tables, associates column information from different data tables, and leverages the background knowledge of data governance professionals to interactively analyze, comprehend, and determine recommended imputation values for missing data in the candidate matrix. The proposed method is validated using a dataset from a large-scale supply chain enterprise and is compared with existing mainstream data imputation methods through comparative experiments and user surveys. The effectiveness and practicality of the method are quantitatively evaluated and validated.

Contributors

Haiyang ZHU conceptualized the main idea and led the research. Haiyang ZHU and Wei CHEN surveyed the relevant materials. All the authors had in-depth discussions; they drafted, revised, and finalized the paper.

Compliance with ethics guidelines

Haiyang ZHU, Dongming HAN, Jiacheng PAN, Yating WEI, Yingchaojie FENG, Luoxuan WENG, Ketian MAO, Yuankai XING, Jianshu LV, Qiucheng WAN, and Wei CHEN declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Ahuja S, Roth M, Gangadharaiah R, et al., 2016. Using machine learning to accelerate data wrangling. Proc IEEE 16th Int Conf on Data Mining Workshops, p.343-349. https://doi.org/10.1109/ICDMW.2016.0055

Arbesser C, Spechtenhauser F, Mühlbacher T, et al., 2017. Visplause: visual data quality assessment of many time series using plausibility checks. *IEEE Trans Visual Comput Graph*, 23(1):641-650. https://doi.org/10.1109/TVCG.2016.2598592

Azur MJ, Stuart EA, Frangakis C, et al., 2011. Multiple imputation by chained equations: what is it and how does it work? *Int J Methods Psych Res*, 20(1):40-49. https://doi.org/10.1002/mpr.329

Bernard J, Hutter M, Reinemuth H, et al., 2019. Visual-interactive preprocessing of multivariate time series data. *Comput Graph Forum*, 38(3):401-412. https://doi.org/10.1111/cgf.13698

Bernhard J, Cella DF, Coates AS, et al., 1998. Missing quality of life data in cancer clinical trials: serious problems and challenges. *Statist Med*, 17(5-7):517-532. https://doi.org/10.1002/(SICI)1097-0258(19980315/15)17:5/7<517::AID-SIM799>3.0.CO;2-S

Bögl M, Filzmoser P, Gschwandtner T, et al., 2015. Visually and statistically guided imputation of missing values in univariate seasonal time series. Proc IEEE Conf on Visual Analytics Science Technology, p.189-190. https://doi.org/10.1109/VAST.2015.7347672

Bonneau GP, Hege HC, Johnson CR, et al., 2014. Overview and state-of-the-art of uncertainty visualization. In: Hansen CD, Chen M, Johnson CR (Eds.), Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization. Springer, London, UK, p.3-27. https://doi.org/10.1007/978-1-4471-6497-5_1

Bors C, Gschwandtner T, Miksch S, 2015. QualityFlow: provenance generation from data quality. Proc EuroVIS Conf on Visualization Posters Track.

Bors C, Bögl M, Gschwandtner T, et al., 2017. Visual support for rastering of unequally spaced time series. Proc 10th Int Symp on Visual Information Communication and Interaction, p.53-57. https://doi.org/10.1145/3105971.3105984

Buono P, Aris A, Plaisant C, et al., 2005. Interactive pattern search in time series. Proc SPIE 5669, Visualization and Data Analysis, p.175-186. https://doi.org/10.1117/12.587537

Chai XT, Gu HM, Li F, et al., 2020. Deep learning for irregularly and regularly missing data reconstruction. *Sci Rep*, 10(1):3302. https://doi.org/10.1038/s41598-020-59801-x

Chen W, Zhang TY, Zhu HY, et al., 2021. Perspectives on cross-domain visual analysis of cyber-physical-social big data. *Front Inform Technol Electron Eng*, 22(12):1559-1564. https://doi.org/10.1631/FITEE.2100553

Djurcilov S, Pang A, 1999. Visualizing gridded datasets with large number of missing values. Proc Visualization, p.405-408. https://doi.org/10.1109/VISUAL.1999.809916

Eaton C, Plaisant C, Drizd T, 2005. Visualizing missing data: classification and empirical study. Proc IFIP Int Conf on Human–Computer Interaction, p.861-872.

Emmanuel T, Maupong T, Mpoeleng D, et al., 2021. A survey on missing data in machine learning. *J Big Data*, 8(1):140. https://doi.org/10.1186/s40537-021-00516-9

Enders CK, 2022. Applied Missing Data Analysis. Methodology in the Social Sciences Series (2nd Ed.). Guilford Press, New York, USA.

Fernstad SJ, Glen RC, 2014. Visual analysis of missing data—To see what isn't there. Proc IEEE Conf on Visual Analytics Science Technology, p.249-250. https://doi.org/10.1109/VAST.2014.7042514

Furche T, Gottlob G, Libkin L, et al., 2016. Data wrangling for big data: challenges and opportunities. Proc 19th Int Conf on Extending Database Technology, p.473-478. https://doi.org/10.5441/002/edbt.2016.44

Gao J, 2006. Adaptive interpolation algorithms for temporal-oriented datasets. Proc 13th Int Symp on Temporal Representation and Reasoning, p.145-151. https://doi.org/10.1109/TIME.2006.4

Githungo W, Otengi S, Wakhungu J, et al., 2016. Infilling monthly rain gauge data gaps with satellite estimates for ASAL of Kenya. *Hydrology*, 3(4):40. https://doi.org/10.3390/hydrology3040040

Griethe H, Schumann H, 2006. The visualization of uncertain data: methods and problems. Proc SimVis, p.143-156.

Gschwandtner T, Gärtner J, Aigner W, et al., 2012. A taxonomy of dirty time-oriented data. Proc Int Conf on Availability, Reliability, and Security, p.58-72. https://doi.org/10.1007/978-3-642-32498-7_5

Gülensoy K, Gawrilow C, von Landesberger T, 2014. Visual exploration of dirty activity sensor and emotional state data from psychological experiments. Proc 14th Int Conf on Knowledge Technologies and Data-Driven Business, Article 19. https://doi.org/10.1145/2637748.2638432

Gupta M, Soeny K, 2021. Algorithms for rapid digitalization of prescriptions. *Visual Inform*, 5(3):54-69. https://doi.org/10.1016/j.visinf.2021.07.002

Harlim J, Jiang SW, Liang SW, et al., 2021. Machine learning for prediction with missing dynamics. *J Comput Phys*, 428:109922. https://doi.org/10.1016/j.jcp.2020.109922

Huang G, Guo C, Kusner MJ, et al., 2016. Supervised word mover's distance. Proc 30th Int Conf on Neural Information Processing Systems, p.4869-4877.

Kamal A, Dhakal P, Javaid AY, et al., 2021. Recent advances and challenges in uncertainty visualization: a survey. *J Visual*, 24(5):861-890. https://doi.org/10.1007/s12650-021-00755-1

Kandel S, Heer J, Plaisant C, et al., 2011. Research directions in data wrangling: visualizations and transformations for usable and credible data. *Inform Visual*, 10(4):271-288. https://doi.org/10.1177/1473871611415994

Kang H, 2013. The prevention and handling of the missing data. *Korean J Anesthesiol*, 64(5):402-406. https://doi.org/10.4097/kjae.2013.64.5.402

Kim W, Choi BJ, Hong EK, et al., 2003. A taxonomy of dirty data. *Data Min Knowl Discov*, 7(1):81-99. https://doi.org/10.1023/A:1021564703268

Kök İ, Özdemir S, 2021. DeepMDP: a novel deep-learning-based missing data prediction protocol for IoT. *IEEE Int Things J*, 8(1):232-243. https://doi.org/10.1109/JIOT.2020.3003922

Kusner M, Sun Y, Kolkin N, et al., 2015. From word embeddings to document distances. Proc 32nd Int Conf on Machine Learning, p.957-966.

Lajeunesse MJ, 2013. Recovering missing or partial data from studies: a survey of conversions and imputations for meta-analysis. In: Koricheva J, Gurevitch J, Mengersen K (Eds.), Handbook of Meta-Analysis in Ecology and Evolution. Princeton University Press, Princeton, USA, p.195-206. https://doi.org/10.1515/9781400846184-015

Little RJA, Rubin DB, 2002. Statistical Analysis with Missing Data (2nd Ed.). John Wiley & Sons, New York, USA. https://doi.org/10.1002/9781119013563

Liu YJ, Fang YJ, Zhu XM, 2010. Modeling of hydraulic turbine systems based on a Bayesian–Gaussian neural network driven by sliding window data. *J Zhejiang Univ Sci C (Comput & Electron)*, 11(1):56-62. https://doi.org/10.1631/jzus.C0910176

Luo Y, 2022. Evaluating the state of the art in missing data imputation for clinical data. *Brief Bioinform*, 23(1):bbab489. https://doi.org/10.1093/bib/bbab489

Marlin BM, 2008. Missing Data Problems in Machine Learning. PhD Thesis, University of Toronto, Toronto, Canada.

Mazumder R, Hastie T, Tibshirani R, 2010. Spectral regularization algorithms for learning large incomplete matrices. *J Mach Learn Res*, 11:2287-2322.

McCarthy JD, Graniero PA, 2006. A GIS-based borehole data management and 3D visualization system. *Comput Geosci*, 32(10):1699-1708. https://doi.org/10.1016/j.cageo.2006.03.006

Miao XY, Wu YY, Chen L, et al., 2023. An experimental survey of missing data imputation algorithms. *IEEE Trans Knowl Data Eng*, 35(7):6630-6650. https://doi.org/10.1109/TKDE.2022.3186498

Nijman SWJ, Leeuwenberg AM, Beekers I, et al., 2022. Missing data is poorly handled and reported in prediction model studies using machine learning: a literature review. *J Clin Epidemiol*, 142:218-229. https://doi.org/10.1016/j.jclinepi.2021.11.023

Palocsay SW, Markham IS, Markham SE, 2010. Utilizing and teaching data tools in Excel for exploratory analysis. *J Bus Res*, 63(2):191-206. https://doi.org/10.1016/j.jbusres.2009.03.008

Pedregosa F, Varoquaux G, Gramfort A, et al., 2011. Scikit-learn: machine learning in Python. *J Mach Learn Res*, 12:2825-2830.

Rässler S, 2004. Data fusion: identification problems, validity, and multiple imputation. *Austr J Stat*, 33(1-2):153-171.

Raubenheimer J, 2017. Excel-lence in data visualization?: the use of Microsoft Excel for data visualization and the analysis of big data. In: Prodromou T (Ed.), Data Visualization and Statistical Literacy for Open and Big Data. IGI Global Information Science Reference, Hershey, Pennsylvania, USA, p.153-193. https://doi.org/10.4018/978-1-5225-2512-7.ch007

Rubinsteyn A, Feldman S, 2016. Fancyimpute: an Imputation Library for Python (Version: 0.7.0). https://github.com/iskandr/fancyimpute

Scheffer J, 2002. Dealing with missing data. *Res Lett Inform Math Sci*, 3(1):153-160.

Smith DM, 2003. The cost of lost data. *J Contemp Bus Pract*, 6(3):1-9.

Stekhoven DJ, Bühlmann P, 2012. MissForest-non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112-118. https://doi.org/10.1093/bioinformatics/btr597

Sun YJ, Li J, Chen SM, et al., 2022. A learning-based approach for efficient visualization construction. *Visual Inform*, 6(1):14-25. https://doi.org/10.1016/j.visinf.2022.01.001

Swayne DF, Buja A, 1998. Missing data in interactive high-dimensional data visualization. *Comput Stat*, 13(1):15-26.

Templ M, Alfons A, Filzmoser P, 2012. Exploring incomplete data using visualization techniques. *Adv Data Anal Classif*, 6(1):29-47. https://doi.org/10.1007/s11634-011-0102-y

Troyanskaya O, Cantor M, Sherlock G, et al., 2001. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520-525. https://doi.org/10.1093/bioinformatics/17.6.520

Turkay C, Lundervold A, Lundervold AJ, et al., 2012. Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE Trans Visual Comput Graph*, 18(12):2621-2630. https://doi.org/10.1109/TVCG.2012.256

Twiddy R, Cavallo J, Shiri SM, 1994. Restorer: a visualization technique for handling missing data. Proc Visualization, p.212-216. https://doi.org/10.1109/VISUAL.1994.346317

Unwin A, Hawkins G, Hofmann H, et al., 1996. Interactive graphics for data sets with missing values—MANET. *J Comput Graph Stat*, 5(2):113-122. https://doi.org/10.1080/10618600.1996.10474700

Wang HN, Liu N, Zhang YY, et al., 2020. Deep reinforcement learning: a survey. *Front Inform Technol Electron Eng*, 21(12):1726-1744. https://doi.org/10.1631/FITEE.1900533

Wang XM, Wu ZL, Huang WQ, et al., 2023. VIS+AI: integrating visualization with artificial intelligence for efficient data analysis. *Front Comput Sci*, 17(6):176709. https://doi.org/10.1007/s11704-023-2691-y

Wong BLW, Varga M, 2012. Black holes, keyholes and brown worms: challenges in sense making. *Proc Human Factors Ergon Soc Annu Meet*, 56(1):287-291. https://doi.org/10.1177/1071181312561067

Wu LF, Yen IEH, Xu K, et al., 2018. Word mover's embedding: from Word2Vec to document embedding. Proc Conf on Empirical Methods in Natural Language Processing, p.4524-4534. https://doi.org/10.18653/v1/D18-1482

Wu ZL, Chen W, Ma YX, et al., 2023. Explainable data transformation recommendation for automatic visualization. *Front Inform Technol Electron Eng*, 24(10):1007-1027. https://doi.org/10.1631/FITEE.2200409

Yang Y, Zhuang YT, Pan YH, 2021. Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. *Front Inform Technol Electron Eng*, 22(12):1551-1558. https://doi.org/10.1631/FITEE.2100463

Yi XW, Zheng Y, Zhang JB, et al., 2016. ST-MVL: filling missing values in geo-sensory time series data. Proc 25th Int Joint Conf on Artificial Intelligence, p.2704-2710.

Yin S, Wang G, Yang X, 2014. Robust PLS approach for KPI-related prediction and diagnosis against outliers and missing data. *Int J Syst Sci*, 45(7):1375-1382. https://doi.org/10.1080/00207721.2014.886136

Zhang GF, Zhu ZH, Zhu SJ, et al., 2022. Towards a better understanding of the role of visualization in online learning: a review. *Visual Inform*, 6(4):22-33. https://doi.org/10.1016/j.visinf.2022.09.002