A Strategy Utilizing an LLM and Augmented Reality for Handling the Missing Data: A Case Study Using Unity, Vuforia and ChatGPT

Enio Vicente de Limas[1], Rita de Fátima Rodrigues Guimarães[2], Bianchi Serique Meiguins[3], Leonardo Chaves Dutra da Rocha[4], Diego Roberto Colombo Dias[5], and Marcelo de Paiva Guimarães[6(✉)]

[1] Centro Universitário Campo Limpo Paulista (Unifaccamp), Campo Limpo Paulista, SP, Brazil
[2] Instituto Federal de São Paulo (IFSP), Jundiaí, SP, Brazil
[3] Federal University of Pará (UFPA), Belém, PA, Brazil
bianchi@ufpa.br
[4] Universidade Federal de São João del-Rei (UFSJ), São João del-Rei, MG, Brazil
lcrocha@ufsj.edu.brr
[5] Federal University of Espirito do Santo (UFES), Vitório, ES, Brazil
diego.dias@ufes.br
[6] Federal University of São Paulo (Unifesp), Osasco, SP, Brazil
marcelo.paiva@unifesp.br

Abstract. The presence of missing data in databases is common, which can make difficult or even prevent their analysis. Therefore, it is necessary to address them, for example, by filling in values or removing records or columns. This paper aims to present a strategy to facilitate the handling of missing data. To achieve this goal, the strategy uses an LLM (Large Language Models) for generating code and conducting data analysis, and 3D visualizations with Augmented Reality to understand the data missing. This strategy has been implemented in a case study, where users can visually explore, measure, and handle the data missing. The results indicate that the strategy can facilitate the analysis and handling of missing data; however, an increase in the complexity of visualizations and handling of data missing was observed as the scale of the datasets increased.

Keywords: Missing data · Visualization · Augmented reality · LLM · ChatGPT

1 Introduction

Data analysis is a crucial part of many fields, from science to business. However, a common challenge faced is the presence of missing data, which can compromise the quality and reliability of the conclusion obtained [9]. Missing data refers to

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025, LNCS 15898, pp. 273–287, 2026.
https://doi.org/10.1007/978-3-031-97657-5_17

information that should have been provided but, for some reason, was not. This can occur due to intentional human factors (e.g., not knowing how to respond, refusing to respond, and neglect) or unintentional factors (e.g., distraction, technical problems, and forgetfulness) [4,19]. Regardless of the cause, missing data can introduce biases and reduce the statistical power of the analyses.

Selecting the appropriate imputation method to fill in missing data is a challenge, as each method has its own assumptions and limitations. Simple methods, such as mean or median imputation, may be insufficient to capture the variability of the data. On the other hand, more complex methods, such as multiple imputation or machine learning-based algorithms, require greater computational power and a deep understanding of the data context. There are many factors involved in each method of handling missing data that need to be understood before selecting one of them. Otherwise, new problems may arise, such as overfitting when utilizing interpolation over hardly dispersed information [16].

Understanding missing data in a database presents a challenge that requires analysis to identify patterns and underlying causes of the missing data. The absence can be classified as random (MCAR), where the lack of data is not related to other values, or non-random (MAR or MNAR), where the absence is associated with other observed values or the missing values themselves [7,9].

Visualization is a promising tool for identifying patterns and underlying causes of missing data. By representing data graphically, it is possible to identify trends, anomalies, and gaps that might not be apparent in raw data. Through various graphical representations, such as heat maps, missing value matrices, and bar charts, one can understand where and how missing data occurs. For example, heat maps can depict concentrations of missing values across different features and instances, revealing if data absence is random or clustered. Bar charts can compare the frequency of missing data across different categories, indicating if certain groups are more prone to missing data. 3D charts can be useful for understanding missing values because they provide an enhanced visual representation of data patterns and anomalies. By mapping missing data onto the third axis, such as color intensity or size of data points, 3D charts can effectively highlight the distribution and clustering of missing values within the dataset. This enhanced visualization capacity enables a better grasp of the spatial distribution of missing data and helps identify any potential patterns or trends associated with their absence, facilitating the choice of imputation method to be used.

The capability of Large Language Models (LLMs) to extract statistical information about patterns and characteristics of missing data further enhances their potential as valuable tools for extracting information from databases. Additionally, LLMs can assist in generating imputation strategies or recommending appropriate methods for handling missing data based on statistical principles and machine learning algorithms. Moreover, LLMs can generate initial programming code, automate tasks, and improve productivity [10,22,25].

This paper aims to present a strategy to handle data missing; and, a case study illustrating our strategy. This strategy consists of a cycle with two subcycles: (1) Data Analysis and (2) Data Treatment. An LLM is utilized to support

the generation of source code for the 3D charts, extract information from the database, and develop methods for handling missing data. Additionally, Augmented reality (AR) is used to enhance understanding of missing data through 3D charts overlaid on the real world and to assist in selecting the appropriate methods to apply. The cycle is restarted and executed until the user is satisfied with the handling of the missing data.

The remainder of this paper is organized as follows. Section 2 discusses prior work related to this research. Section 3 describes the strategy. Section 4 shows a case study involving our strategy. Finally, Sect. 5 presents our conclusions.

2 Related Work

Visualization is a powerful tool for analyzing and sharing information. Humans process visual data rapidly, and visualization enables them to interpret it from various perspectives. This leads to a deeper understanding of problems and the identification of trends [6].

Costa et al. [24] developed an AR application that uses markers for creating, manipulating, and interacting with visualizations. However, they encountered difficulties in selecting virtual objects and filtering information. To address these issues, they suggested implementing menus to assist with these tasks. Miranda et al. [13] proposed a mobile AR application for data visualization. They used markers to interact with the visualizations. They also noted limitations in interacting with the visualization content, such as the restricted number of markers that can be recognized simultaneously and challenges in selection methods.

Padwal et al. [15] also explore the use of AR for data visualization. They created a prototype application that demonstrates various visualization techniques using AR. Tong et al. [23] investigated the interaction with printed data visualizations in AR. The participants of their study considered the interactions using AR intuitive and engaging, despite some physical limitations and properties of paper as a medium. Chandra, Jamiy, and Reza [17] evaluate various conventional visualization techniques and their application in creating visualizations with AR. They also explore the potential of integrating big data techniques with AR. They argue that most applications are still in the early stages, due to limitations such as hardware constraints, usability issues, and integration challenges. They emphasize that enhancing the user experience is the main benefit of visualizations with augmented reality.

Sjöbergh and Tanaka [21] explore the use of data visualization to handle missing data within real big data projects. They emphasize the importance of understanding the nature of missing data before selecting a treatment method. Furthermore, they highlight that while certain visualizations are suitable for specific types of missing data, others may not be as effective. Hence, a diverse range of visualization methods is essential for comprehensively addressing the nuances of missing data.

Addressing missing data is of critical importance in the field of machine learning, as it directly affects model performance and accuracy. Effectively managing

missing data is essential to ensure the quality, reliability, and validity of insights derived from the data.

The machine learning field has extensively studied the impact of missing data and methods to handle it [2,3,11]. Proper processing of missing data is crucial to ensure the quality, reliability, and accuracy of the insights derived from the data. The strategy proposed in this paper uses an LLM for code generation and data analysis, combined with 3D visualizations using AR to better understand the characteristics of missing data. This approach facilitates the selection of appropriate methods for handling missing data while maintaining flexibility to incorporate additional visualizations and techniques as needed.

3 Strategy

Our strategy aims to provide resources for exploratory data analysis and thus facilitate the selection of a method to handle missing data. To achieve this, it seeks to identify, locate, and measure the missing data. Additionally, it aims to facilitate the selection of methods for handling other data anomalies, such as outliers and data inconsistencies.

The objective of the Data Analysis sub-cycle is to provide visualizations through 3D charts to identify, assess, locate, and manage anomalies in a specific dataset. To achieve this, it uses AR technology. One of the strengths of visualization with AR is the wide visual space available for presenting information [13]. The sub-cycle is initiated when an AR marker is detected. Each marker corresponds to a query previously constructed in the dataset. Subsequently, the information resulting from the query is presented as 3D charts overlaid on the real environment. The displayed information includes, for example, the total amount of data, the total amount of missing data, and the percentage of missing data relative to the total amount of data. This allows for precise measurement and location of missing data. The process of creating new charts can be assisted by a LLM.

The Data Treatment sub-cycle aims to manipulate the data through the selection of methods for handling missing data (e.g., record elimination, mean values, and regression [1,5,9,12,18,20]) in an AR menu. An intermediate task of this sub-cycle involves analyzing the characteristics of each column from the database to choose the method that will be applied. Each method is designed for specific missing data characteristics (e.g., type and amount of missing data). Panda and Adhikari [16] discuss the techniques that are frequently used (e.g., missing data ignoring, missing data imputation, missing data model) to handle missing data and, in addition, propose the use of correlation between attributes. Specific analysis should be performed before choosing the method because it potentially impacts the analyses and treatment results [14]. If the available methods are not sufficient, an LLM can be used to support the creation of the method source code. Using an LLM tool can significantly enhance analyst productivity. One key advantage is its ability to provide quick information and answers related to missing data and coding. Instead of manually searching forums or documents,

analysts can ask the LLM tool questions and receive real-time responses. This not only saves time but also allows analysts to focus on higher value-added tasks, thereby increasing the overall efficiency of the missing data handling treatment.

Figure 1 depicts an overview of the strategy, which consists of the following steps:

- Step 1: the first sub-cycle, Data Analysis, begins when the device's camera detects an AR marker. Then, a query to the dataset is performed, and a 3D chart is added to the environment. This allows the initial analysis of the data (for example, the amount of missing data per column);
- Step 2: the user can follow two paths:
  - Step 2.1: create and analyze a new 3D chart assisted by an LLM; associates the new chart with a new marker; (The visualization is not satisfactory) and return to Step 1; or
  - Step 2.2: initialize the Data Treatment sub-cycle, and go to Step 3 (The visualization provided a satisfactory understanding);
- Step 3: using the AR menu from the Data Treatment sub-cycle, the user selects one of the available methods for handling the missing data in the dataset
- Step 4: the user can follow two paths:
  - Step 4.1: apply a method for handling missing data, and go to Step 1 (There is an appropriate method in the menu to be used); finish or
  - Step 4.2: generate the code for data missing assisted by a LLM; add the new method to the AR menu, (There is not an appropriate method in the menu to be used); and go to Step 4;

The sub-cycles are executed until the conditions for data visualization and/or handling of missing data are satisfied. Therefore, the strategy aims to improve the efficiency of data visualization and handling of missing data, using available technologies.

4 Case Study

An application was developed to illustrate the implementation of the strategy, called MDT-AR (Missing Data Treatment with the help of Augmented Reality). It was created using Unity and Vuforia. Unity is a game engine equipped with a complete set of integrated tools based on Microsoft C# for developing 2D and 3D games. Vuforia is a SDK (software development kit) for mobile devices that enables the creation of AR applications. It detects and tracks physical objects from real-world camera images, then overlays virtual elements such as graphics, animations, videos, and texts. This creates a fusion between the real world and virtual elements through the device's camera. The C# programming language enables interactions with these virtual objects.

MDT-AR consists of two modules, one for each sub-cycle (Data analysis and Data treatment). The following sub-sections detail each of the modules.

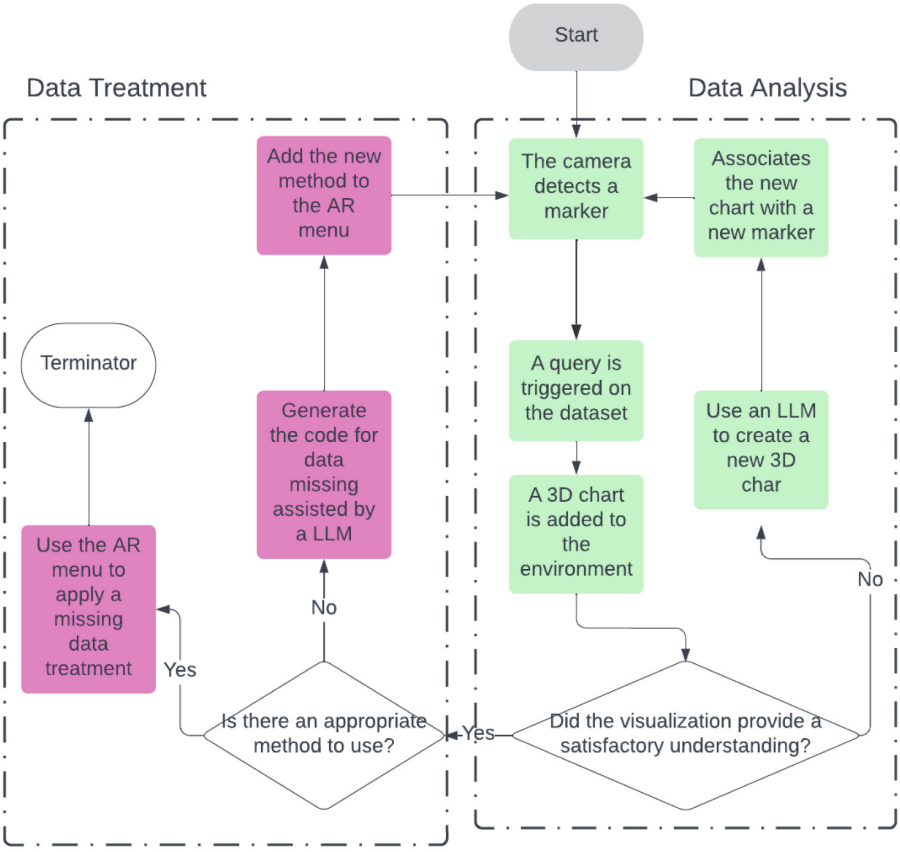

Fig. 1. Strategy overview

4.1 Data Analysis Module

The main objective of this module is to present 3D charts to enable the identification, assessment, location, and measurement of missing data and other anomalies in a specific dataset. When an AR marker is detected, a predefined SQL query is triggered in the database. Then, a 3D chart displays the query results. The graph can show, for example, the total amount of data, the quantity of missing data, and the percentage of missing data. To perform different queries, you need to change the marker or create new ones.

Figure 2 depicts an example of a chart generated showing missing data from a database. In this example, the *Description* column (green) has 917 instances, of which 105 are null data, representing 11.45% of the total data in that column. The *Desc_Title* column (black) has 917 instances, of which 207 are missing data, representing 21.92% of the total data in this column. The *Desc_Title_Formatted* column (blue) has 917 instances, of which 207 are missing data, representing

21.92% of the total data in this column. When the marker is rotated, the quality of the visualizations is improved according to the desired orientation.

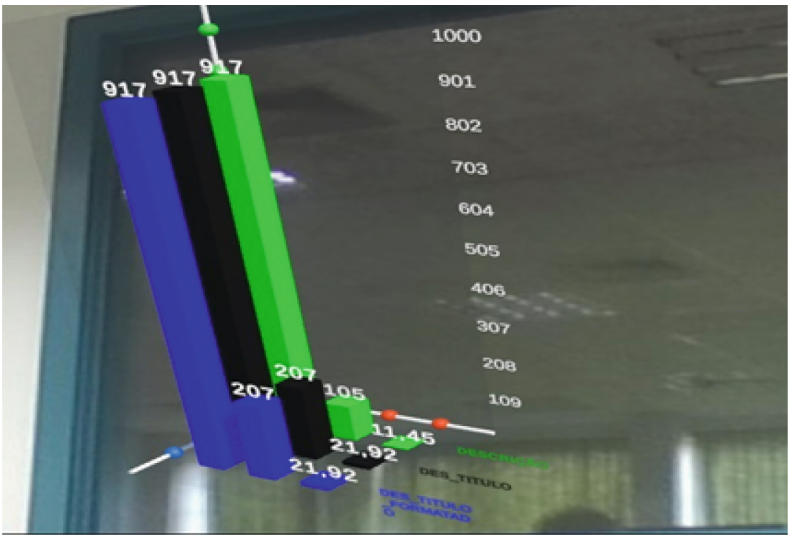

Fig. 2. 3D visual analysis of missing data

If the visualization has provided a satisfactory understanding of the data, you can then start the process of processing the missing data. Otherwise, you can visualize the data using another previously constructed chart or by requesting the LLM tool's assistance in creating a new chart. This new 3D chart will be linked to a new marker, and this sub-cycle will restart.

In this case study, ChatGPT 3.5 was used as an LLM to assist the MDT-AR. We used a data file in .csv format with 1000 records and 8 columns, 6 of which are categorical and 2 of which are numerical. Figure 3 illustrates the missing data highlighted in red.

Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
Douglas	Male	8/6/1993	12:42 PM	97308	6945	true	Marketing
Thomas	Male	3/31/1996	6:53 AM	61933	4.17	true	
Maria	Female	4/23/1993	11:17 AM	130590	11858	false	Finance
Jerry	Male	3/4/2005	1:00 PM	138705	9.34	true	Finance
Larry	Male	1/24/1998	4:47 PM	101004	1389	true	Client Services
Dennis	Male	4/18/1987	1:35 AM	115163	10125	false	Legal
Ruby	Female	8/17/1987	4:20 PM	65476	10012	true	Product
	Female	7/20/2015	10:43 AM	45906	11598		Finance
Angela	Female	11/22/2005	6:29 AM	95570	18523	true	Engineering
Frances	Female	8/8/2002	6:51 AM	139852	7524	true	Business Development
Louise	Female	8/12/1980	9:01 AM	63241	15132	true	

Fig. 3. Case study data source (Color figure online)

The interaction with the LLM to solve tasks is done through a *prompt*, which makes the model behave according to the context [8]. Various interactions with the *prompt* improve the accuracy and relevance of the answers, making the model more effective in generating relevant information [26]. In the case study, there were 12 interactions to analyze and apply the method of treating missing data.

In the initial interactions via the ChatGPT *prompt*, a code was requested to display the statistical description using the following text: "Generate information about statistics on the numerical data from a .csv file.". Listing 1.1 shows the result, which contains source code that executed returns the main statistical information of the data (count, mean, standard deviation, min, 25th percentile, median, 75th percentile, and max) and transposes it for easier viewing. The *generate_statistics_from_file* function has been sufficiently created, which receives a file path as input, reads the data into a DataFrame and returns a DataFrame with statistical summaries for the numerical columns. An example of how to use it was also created. The generated code could run without any modifications.

```
def generate_statistics_from_file(file_path):
    # Read the data from the file into a DataFrame
    df = pd.read_csv(file_path)
    # Select only numerical columns
    numerical_df = df.select_dtypes(include=['number'])
    # Generate basic statistical summary
    statistical_summary = numerical_df.describe().transpose()
    # Add median
    statistical_summary['median'] = numerical_df.median()
    # Add variance
    statistical_summary['variance'] = numerical_df.var()
    # Add standard deviation
    statistical_summary['std_dev'] = numerical_df.std()
    return statistical_summary

# Example usage
file_path = 'path_to_your_file.csv'
statistics = generate_statistics_from_file(file_path)
print(statistics)
```

Listing 1.1. Using ChatGPT to generate source code to retrieve statistics from a data source.

We also asked the following *prompt*: "Create a script using 3D bar charts in C# for Unity to visualize the total missing data for each column and the percentage of missing data.". Listing 1.2 shows the result, it creates 3D bars to visualize the missing data for each column and the percentage of missing data. The *VisualizeMissingData* method was created to calculate the total missing data and the percentage of missing data for each column, and to draw bars

accordingly. The *DrawBar* method draws a bar at the specified position, with the given height and color. The code needed adjustments for the desired dataset.

```
using System.Collections.Generic;
using UnityEngine;

public class MissingDataVisualization: MonoBehaviour
{
    ...
 private void VisualizeMissingData()
  {
   int numRows = data.GetLength(0);
   int numCols = data.GetLength(1);
   ...
   float percentage = (float)missingCount / numRows * 100f;
   // Draw bar for total missing data
   DrawBar(startX + j * barWidth, 0f, missingCount, Color.blue);
   // Draw bar for percentage of missing data
   DrawBar(startX + j * barWidth, 1f, percentage, Color.red);
 }

private void DrawBar(float x, float z, float height, Color color)
 {
  GameObject bar = new GameObject("Bar");
  bar.transform.parent = transform;
  LineRenderer lineRenderer = bar.AddComponent<LineRenderer>();
  ...
}
```

Listing 1.2. 3D chart to visualize missing data

While ChatGPT 3.5 can provide a useful starting point for 3D charts in Unity, they have significant limitations. Creating high-quality 3D graphics typically requires specialized knowledge and iterative adjustments. Moreover, ChatGPT 3.5 is not directly integrated with Unity. Therefore, the code generated cannot directly interact with the Unity environment or debugger in real-time. ChatGPT 3.5 can be used as an assisting tool but not as a comprehensive solution for generating code for 3D charts in Unity.

4.2 Data Treatment Module

Besides understanding the characteristics of missing data during the analysis stage, knowing its type is essential to apply the correct treatment method. In this manner, you can choose one of the available methods from the virtual menu from the case study: average, mode, or regression. It is also possible to find missing data values from a box plot.

If the user is satisfied with the applied data treatment, they can conclude the sub-cycle or request the application of another method. ChatGPT 3.5 can assist in creating new methods if needed. When the virtual menu is accessed again, a new method becomes available, prompting a restart of the sub-cycle. Figure 4 depicts the virtual menu with the available treatment methods. This boxplot displays the quantity of records(*regs*), showing the number of outliers above the upper whisker (*Outliers_GT*) and the number of outliers below the lower whisker (Outliers_*LT*).

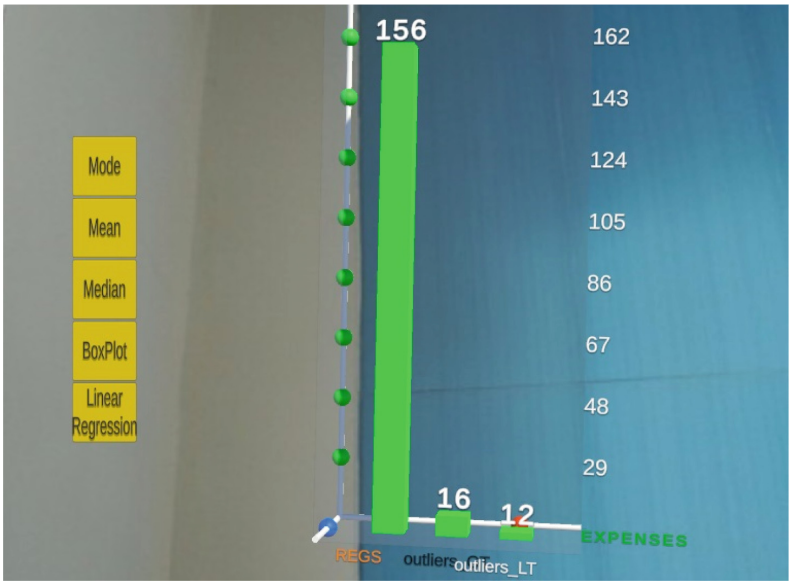

Fig. 4. Methods for data treatment

In cases where the available methods do not meet the requirements, ChatGPT 3.5 can be utilized to assist in creating a customized treatment method. We request the following task via the *prompt*: "Create a C# for unity to replace the categorical columns with missing data using mode imputation". Listing 1.3 shows the result, it creates a script to replace the data. The *ImputeMode* method is responsible for replacing the data. This code needed adjustments for the desired dataset.

```
public class ModeImputation : MonoBehaviour
{
    void Start()
    {
        ...
        ImputeMode(data);
        ...
    }
    void ImputeMode(List<Dictionary<string, object>> data)
    {
        // Get all columns
        ...
        foreach (var column in columns)
        {
    // Filter out rows with missing values in the current column
            ...
                // Calculate the mode (most frequent value)
                var mode = values
                    .GroupBy(v => v)
                    .OrderByDescending(g => g.Count())
                    .First()
                    .Key;
                // Impute missing values with the mode
              foreach (var row in data.Where(row => row[column] ==null))
                {
                    row[column] = mode;
                }
        }
    }
```

Listing 1.3. Method to replace the categorical data

The use of ChatGPT 3.5 in the case study accelerated the data analysis and treatment by providing statistical calculations and generating code. However, the generated code always required adjustments before it could be effectively used. In total, 12 interactions were conducted via prompts. In the end, the missing data was successfully treated. The fields highlighted in green in Fig. 5 illustrate the treatment of missing data that was carried out.

Figure 6 depicts the 3D chart after the missing data treatment has been applied. It shows the total amount of data in each column, the total amount of missing data, and the percentage of missing data. Both the total amount and percentage of missing data are zero, indicating successful data processing. This step completes the strategy cycle in the MDT-AR.

Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
Douglas	Male	8/6/1993	12:42 PM	97308	6945.0	True	Marketing
Thomas	Male	3/31/1996	6:53 AM	61933	4.17	True	Client Services
Maria	Female	4/23/1993	11:17 AM	130590	11858.0	False	Finance
Jerry	Male	3/4/2005	1:00 PM	138705	9.34	True	Finance
Larry	Male	1/24/1998	4:47 PM	101004	1389.0	True	Client Services
Dennis	Male	4/18/1987	1:35 AM	115163	10125.0	False	Legal
Ruby	Female	8/17/1987	4:20 PM	65476	10012.0	True	Product
Marilyn	Female	7/20/2015	10:43 AM	45906	11598.0	True	Finance
Angela	Female	11/22/2005	6:29 AM	95570	18523.0	True	Engineering
Frances	Female	8/8/2002	6:51 AM	139852	7524.0	True	Business Development
Louise	Female	8/12/1980	9:01 AM	63241	15132.0	True	Client Services

Fig. 5. Dataset after treatment of missing data (Color figure online)

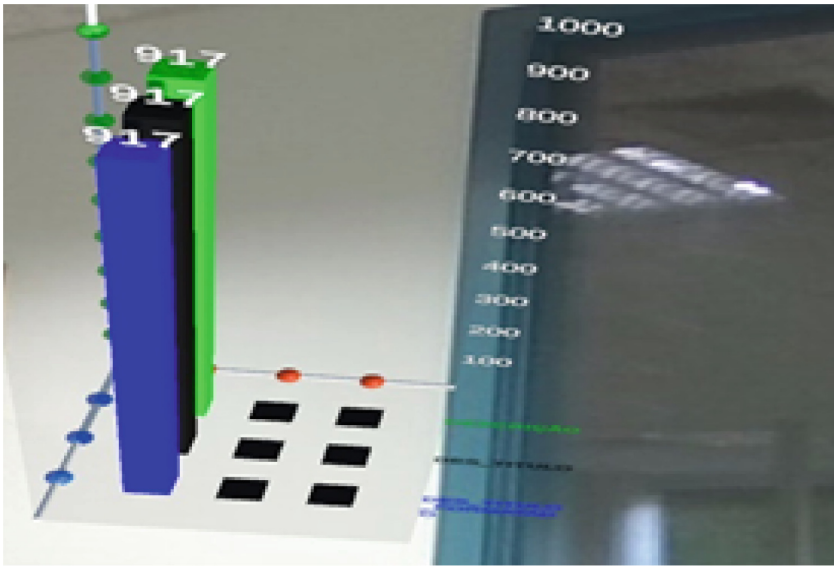

Fig. 6. Post-treatment data

The case study MDT-AR still faces some limitations, such as performance degradation in visualizations as the data volume increases, the absence of comparison features between different methods, the lack of precise graphics control, and it requires technical knowledge in both data analysis and software development. However, it remains a promising tool due to its potential to facilitate the identification of missing data patterns, enhance user engagement with attractive visualizations, and provide interactivity—allowing users to manipulate, rotate, and explore models from various angles and perspectives. Furthermore, it can be extended and modified to incorporate new data visualization and treatment methods. Although ChatGPT's code generation integration with Unity exists, the current version of MDT-AR does not utilize this feature, primarily due to the need for code modifications to meet specific visualization requirements. Instead, the code generated by ChatGPT is received via an HTTP response, parsed in

Unity, and the source code is extracted. Currently, the MDT-AR user must be a developer with proficiency in AR application development, expertise in dataset manipulation, experience interacting with LLMs, and knowledge of handling missing data.

5 Conclusion

In this paper, we present a strategy developed to address missing data in a dataset. The strategy includes two sub-cycles: the Data Analysis sub-cycle, and the Data Treatment sub-cycle, both utilizing augmented reality. Additionally, the strategy incorporates the use of an LLM as a support tool, assisting in data analysis and processing.

To demonstrate the implementation of our strategy, we developed an application called MDT-AR, designed to analyze and process the missing data in a database. This application was built using the Unity game engine, Vuforia, and the LLM ChatGPT 3.5.

The AR features for handling missing data provide users with an intuitive and immersive visualization experience, enabling a detailed understanding of the data. This technology offers a wide variety of ways to visualize the data, which motivates research. It allows users to manipulate, rotate, and explore models from various angles and perspectives, seamlessly integrating virtual objects with the real world.

While LLMs like ChatGPT 3.5 can serve as valuable tools for offering general guidance and code examples, they have notable limitations in assisting with the analysis and handling of missing data. Effective data analysis and handling typically necessitate a combination of expert knowledge, access to real data, utilization of advanced statistical techniques, and a deep understanding of the data context, all of which may be beyond the current capabilities of LLMs.

Future work involves adding new analytical tools, such as additional chart types, and implementing new methods for handling missing data. Additionally, we plan to conduct tests with larger data volumes and perform user evaluations to further refine and enhance the application.

Acknowledgment. The authors acknowledge partial funding from the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Funding Code 001, the National Council for Scientific and Technological Development (CNPq), and the Research Support Foundation of the State of Minas Gerais (FAPEMIG). We also thank the Research and Innovation Support Foundation of Espírito Santo (FAPES) for the resources provided through the PROAPEM project (368/2022 - P: 2022-NGKM5) and the PDPG project (129/2021 - P: 2021-GL60J), which were fundamental to the completion of this work.

References

1. Alabadla, M., et al.: Systematic review of using machine learning in imputing missing values. IEEE Access **10**, 44483–44502 (2022). https://doi.org/10.1109/ACCESS.2022.3160841
2. Cenitta, D., Arjunan, R.V.: Missing data imputation using machine learning algorithm for supervised learning. In: 2021 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–5 (2021). https://doi.org/10.1109/ICCCI50826.2021.9402558
3. Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., Tabona, O.: A survey on missing data in machine learning. J. Big Data **8**(1), 140 (2021). https://doi.org/10.1186/s40537-021-00516-9
4. Enders, C.K.: Missing data: an update on the state of the art. Psychol. Methods (2023). https://doi.org/10.1037/met0000563
5. Faizin, R.N., Riasetiawan, M., Ashari, A.: A review of missing sensor data imputation methods. In: 2019 5th International Conference on Science and Technology (ICST), vol. 1, pp. 1–6 (2019). https://doi.org/10.1109/ICST47872.2019.9166287
6. Hirve, S.A., Ch, P.R.: Improving big data analytics with interactive augmented reality. Int. J. Inf. Syst. Modeling Des. (IJISMD) **13**(7), 1–11 (2022). https://ideas.repec.org/a/igg/jismd0/v13y2022i7p1-11.html
7. Little, R., Rubin, D.: Statistical analysis with missing data. Wiley Series in Probability and Mathematical Statistics. Probability and Mathematical Statistics. Wiley (2002). http://books.google.com/books?id=aYPwAAAAMAAJ
8. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. ACM Comput. Surv. **55**(9) (2023). https://doi.org/10.1145/3560815
9. Liu, Q., Hauswirth, M.: A provenance meta learning framework for missing data handling methods selection. In: 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0349–0358 (2020). https://doi.org/10.1109/UEMCON51285.2020.9298089
10. Liu, Z., Tang, Y., Luo, X., Zhou, Y., Zhang, L.F.: No need to lift a finger anymore? Assessing the quality of code generation by chatgpt. IEEE Trans. Softw. Eng. 1–35 (2024). https://doi.org/10.1109/TSE.2024.3392499
11. Makaba, T., Dogo, E.: A comparison of strategies for missing values in data on machine learning classification algorithms. In: 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), pp. 1–7 (2019). https://doi.org/10.1109/IMITEC45504.2019.9015889
12. Miao, X., Wu, Y., Chen, L., Gao, Y., Yin, J.: An experimental survey of missing data imputation algorithms. IEEE Trans. Knowl. Data Eng. **35**(7), 6630–6650 (2023). https://doi.org/10.1109/TKDE.2022.3186498
13. Miranda, B.P., Queiroz, V.F., Araújo, T.D.O., Santos, C.G.R., Meiguins, B.S.: A low-cost multi-user augmented reality application for data visualization. Multimedia Tools Appl. **81**(11), 14773–14801 (2022). https://doi.org/10.1007/s11042-021-11141-2
14. Othman, L.B., Yahia, S.B.: A multiple criteria evaluation technique for missing values imputation. In: 2018 12th International Conference on Research Challenges in Information Science (RCIS), pp. 1–12 (2018). https://doi.org/10.1109/RCIS.2018.8406659
15. Padwal, P., Singh, Y., Singh, J., Pansambal, S.: Dvar: data visualization using augmented reality. In: 2021 2nd Global Conference for Advancement in Technology (GCAT), pp. 1–6 (2021). https://doi.org/10.1109/GCAT52182.2021.9587831

16. Panda, B.S., Kumar Adhikari, R.: A method for classification of missing values using data mining techniques. In: 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1–5 (2020). https://doi.org/10.1109/ICCSEA49143.2020.9132935
17. Ramaseri Chandra, A.N., El Jamiy, F., Reza, H.: Augmented reality for big data visualization: a review. In: 2019 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 1269–1274 (2019). https://doi.org/10.1109/CSCI49370.2019.00238
18. Santos, M.S., Pereira, R.C., Costa, A.F., Soares, J.P., Santos, J., Abreu, P.H.: Generating synthetic missing data: a review by missing mechanism. IEEE Access **7**, 11651–11667 (2019). https://doi.org/10.1109/ACCESS.2019.2891360
19. Schafer, J.L., Graham, J.W.: Missing data: our view of the state of the art. Psychol. Methods **7**(2), 147–177 (2002)
20. Sharifyanov, N., Latypova, V.: A method of filling missing values in data using data mining. In: 2023 IX International Conference on Information Technology and Nanotechnology (ITNT), pp. 1–5 (2023). https://doi.org/10.1109/ITNT57377.2023.10139280
21. Sjöbergh, J., Tanaka, Y.: Visualizing missing values. In: 2017 21st International Conference Information Visualisation (IV), pp. 242–249 (2017). https://api.semanticscholar.org/CorpusID:30613266
22. Su, H., Ai, J., Yu, D., Zhang, H.: An evaluation method for large language models' code generation capability. In: 2023 10th International Conference on Dependable Systems and Their Applications (DSA), pp. 831–838 (2023). https://doi.org/10.1109/DSA59317.2023.00118
23. Tong, W., et al.: Exploring interactions with printed data visualizations in augmented reality. IEEE Trans. Visual Comput. Graphics **29**(1), 418–428 (2023). https://doi.org/10.1109/TVCG.2022.3209386
24. Victor Costa, I., Favacho Queiroz, V., Pinto Miranda, B., Abreu de Freitas, A., Gustavo Resque dos Santos, C., Serique Meiguins, B.: A card-based interaction to design visualizations in augmented reality environments. In: 2019 23rd International Conference in Information Visualization, Part II, pp. 52–55 (2019). https://doi.org/10.1109/IV-2.2019.00019
25. Wang, J., Chen, Y.: A review on code generation with LLMs: application and evaluation. In: 2023 IEEE International Conference on Medical Artificial Intelligence (MedAI), pp. 284–289 (2023). https://doi.org/10.1109/MedAI59581.2023.00044
26. White, J., et al.: A prompt pattern catalog to enhance prompt engineering with chatgpt (2023)