Analysis of Normalization Methods Corresponding to Data Types and Their Application in Network Databases

Bahodir Muminov[1,*], Ziyoda Norqulova[1,2]
[1]Tashkent State University of Economics Tashkent, Uzbekistan
[2]Tashkent University of Information Technology, Uzbekistan
Email: mbbahodir@tsue.uz; z.norqulova@tsue.uz

Abstract

Today, information systems handle large volumes of data from various sources. These data may differ in both form and meaning. Such data diversity is one of the main problems in network integration and analysis. This research paper analyzes the main types of data: digital, integer, text, categorical, temporal, logical, and spatial. Today, information systems work with large volumes of information obtained from various sources. This data can differ in both form and meaning. This diversity of data is one of the main problems in the processes of integration and analysis in the network. This research paper analyzes the main types of data: digital, integer, text, categorical, temporal, logical, and spatial. For each type of data, a normalization approach is selected that corresponds to it. In particular, we will study the min-max scaling and Z-score standardization methods for digital data, one-hot and label encoding for category attributes, as well as lemmatization and normalization based on Unicode for text data. The analysis shows that choosing the right approach for each data type increases the efficiency of unification, ontological mapping, and visualization. The article analyzes the advantages and limitations of existing normalization methods and provides practical recommendations for selecting optimal methods for processing network data. The proposed approach can be effectively used in the processes of semantic integration of multi-source network data, as well as to its visual analysis.

Keywords: Data normalization; Network databases; Data types; Semantic consistency; Ontological mapping

1. Introduction

Currently, large volumes of generated data are stored in various sources –servers, web platforms, social networks, IoT devices, or databases. In each source, data is presented in a specific format, units, coding system, and semantic structure. For this reason, the normalization process is important in data processing.

Normalization is the process of bringing information in different formats into a single structural and semantic form. This process reduces differences between data, making it possible to compare and integrate them. In modern approaches, normalization is not limited to cleaning relational databases, but is also applied to large volumes of many types of (heterogeneous) data. In particular, in network data processing systems, where elements such as nodes, connections, attributes, and weights are present, normalization is an important step.

Scientific literature proposes normalization methods specific to different types of information. Approaches such as minimum and maximum scaling, Z-score standardization, or decimal scaling are commonly used for processing digital data. However, single-encoding methods are common in category attributes such as tag encoding. For text data, methods such as lemmatization, stemming, and Unicode normalization are used, and for time data, ISO 8601 formatting or conversion to timestamp format is used. Scientific literature has proposed normalization methods specific to different types of information. Approaches such as minimum and maximum scaling, Z-score standardization, or decimal scaling are commonly used to process numerical data. However, single-encoding methods are common in category attributes such as tag encoding. For text data, methods such as lemmatization, stemming, and Unicode normalization are used, and for time data, ISO 8601 formatting or

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025

conversion to timestamp format is used. However, each of these approaches will have its own advantages and limitations depending on the type of data.

Therefore, choosing the appropriate normalization method with an individual approach to each data type increases the efficiency of network data processing. This approach will not only simplify the integration process, but also allow you to prepare high-quality information for the next stage – ontological mapping and 3D visualization.

The main goal of this study is to analyze data types and normalization methods, determine their interaction, and identify optimal approaches that can be used for network data integration and visualization. The research results serve to ensure the semantic consistency of network data and improve the processes of its representation in a three-dimensional visual environment.

2. Related works

The issue of data normalization has emerged in recent years as one of the most pressing areas of concern in working with large volumes of data. Researchers have proposed various approaches to this issue, with the aim of simplifying the process of combining and analyzing data with different formats, units, and semantics by bringing them into a single form. Below is an analysis of the main scientific approaches and developments in data normalization.

Kalayani Sankpal and K. V. In the Metre article entitled "An Overview of Data Normalization Methods" [1], various levels, methods of data normalization, and their limitations are described in detail. The authors view data normalization as a three-step process: normalization performed at the record, field, and value levels. In this approach, the goal is not only to remove duplicate records, but also to ensure complete semantic consistency of information from different sources. Normalization is divided into three levels depending on the data structure:

Record normalization is the process of identifying and unifying records that belong to the same object but are expressed differently. Algorithms for detecting and merging duplicates are often used for this purpose. Culotta and his colleagues [2] called this process "canonization" and proposed a "distance editing" method to determine the similarity between records. In their study, the central (canonical) record is created by finding the closest match between records. However, since this approach did not take into account differences in value levels, the normalized data was not completely accurate.

The Swoosh system [3] similarly treats data as connections between objects. It aligns and merges records using entity resolution algorithms. However, since this approach did not take into account differences in value levels, the normalized data was not completely accurate. Meanwhile, the approach proposed by Yongquan Dong and Dragut [4] developed a three-step model for record level normalization: merging records, comparing field values, and creating a resulting single record. This model was published in the IEEE Transactions on Knowledge and Data Engineering journal, where high accuracy in the automatic selection of similar records was achieved.

Field level normalization is the process of converting the values of each column in data records to the same format. In this approach, formatting, aligning units of measurement and reducing text differences become increasingly important. For example, "TSEU" and "Tashkent State Economic University" mean the same thing but are written differently. Vic and his colleagues [5] proposed a working model based on schema matching and coreference resolution to address this issue. They performed attribute matching using a (discriminative) model that distinguishes those that do not correspond to a reference. This approach was effective in analyzing structural similarities between data, but did not fully ensure semantic correspondence at the level of meaning. The "object normalization" system [6] proposed by Tejada, Knoblock and Minton also used string ratings and user confidence ratings (confidentiality ratings) to compare attributes. This system performed well in normalizing data from web sources, but in many cases required user intervention. In addition, the label normalization method developed by Dragut and Mann [7] proposes matching data by assigning meaningful labels (semantic labels) to attributes. This approach is considered important in ontological systems, especially for improving the semantic consistency of data.

Value level normalization. Normalization at the value level aims to ensure semantic and linguistic consistency of attribute values at the most granular level of data. At this stage, technologies such as text similarity recognition, abbreviation and synonym normalization, and NLP (natural language processing) are often used.

The model proposed by Sanpal and Meter [1] uses ranking algorithms such as frequency ranking, length ranking, centroid ranking, and similarity analysis based on bigram and edit distance for normalization at the value level. They use the mining abbreviation–definition pair's algorithm to identify abbreviations and the template collocation method to search for text references. The advantage of this approach is that it not only eliminates duplication at the record level, but also takes into account the semantic proximity between values.

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025

However, the fact that natural language processing (NLP) methods were not used in this system limited its accuracy. The researchers noted that much higher accuracy could be achieved by introducing an NLP-based model at later stages.

An analysis of the literature shows that there is no single universal approach to data normalization. Each study performed normalization at a level appropriate to its purpose:

Table 1: Type of normalisation

Type of normalisation	Main task	Goal	Example
Record-level	Identifying and merging records that represent the same object but are stored differently	Removing duplicates, storing data in a unified format	“Aliyev A.” and “A. Aliyev” – merged as the same person
Field-level	Bringing attribute or column values in records to a uniform format	Eliminating inconsistencies in format, consistency, and written form	“TSEU” va “Tashkent state economical university” – standardized the same value
Value-level	Linguistic and semantic adaptation of each value	Eliminating synonyms, abbreviations, and spelling differences, creating semantic harmony	“prof.”,“profesor” or “Profesor” – are rewritten the only form

As can be seen from the table, normalization approaches differ from each other in terms of stages and purpose:

– Letter-level approaches identify and combine repetitive data.

– Field-level approaches lead to uniformity of formats and units of measurement.

– Value-level approaches ensure semantic consistency of data.

3. Methodology

Since the types of data found in network databases, the main types of data are analyzed. Network data contains attributes in various formats, and the following types of data were selected as research material for the study:

Table 2: Data types

N	Type of data	Description	Example	Usage
1.	**Numeric/ Decimal/Float**	Represents whole or decimal numbers	45, 3.14, -120.56	Price, measurement, percentage, calculated values
2.	**Integer/ SmallInt/BigInt**	Stores natural or whole numbers, without fractional parts.	1, 0, 2025, -56	Identification numbers, quantity, amount
3.	**Text / Char / Varchar / String**	Letters, words, texts, or alphanumeric characters.	"Ziyoda", "Tashkent", "ATU-2025"	Name, location, text, code, description
4.	**Enumerated / Categorical / Enum**	Selects one of a limited number of categories.	("Male", "Female"), ("Active", "Inactive")	Status, tour, category
5.	**Date / Time / Timestamp / Datetime**	Stores dates, times, or time intervals.	2025-11-03, 12:30:00	Logs, date of birth, time of process
6.	**Boolean / Bit**	Yes or no, true or false values.	True, False, 1, 0	Conditions, circumstances, activity indicator

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025

7.	**Blob / Binary / Image / JSON**	Non-text objects: image, video, document, or structured information. Stores natural or whole numbers, without fractional parts.	01001011, JSON fayl, .jpg, .pdf	Multimedia, configuration, file storage
8.	**Spatial / Geometry / Point**	Letters, words, texts, or alphanumeric characters.	(41.3111, 69.2797)	Location, GIS, network topology
9.	**Money / Currency**	Selects one of a limited number of categories.	15000.00, $200	Accounting, transactions, economic analysis
10.	**Array / Set / List**	Stores dates, times, or time intervals.	{1, 2, 3} or {“A”, “B”, “C”}	Multi-valued attributes Identification numbers, quantity, number
11.	**JSON / XML / Object**	Yes or no, true or false values.	{"name":"Ziyoda","age":25}	Name, place, text, code, description
12.	**NULL**	Stores natural or whole numbers, without fractional parts.	NULL	Status of any type

Now, based on this table, normalization methods are selected individually, first using the steps presented in Table 1.1, in accordance with each data type and its task, and then analyzing the appropriate normalization methods for each data type.

1. Numerical data normalization is mainly achieved through scaling. The most commonly used model is the min-max normalization method[8].

$$x_i' = \frac{x_i - min(x)}{max(x) - min(x)}$$

here:

- $x_i\ -\ real\ value$
- Min(x) and max(x) – range of values
- x_i' – $result\ of\ normalisation$

An alternative model can also be used – the Z-score normalization method:

$$z_i = \frac{x_i - \mu}{\sigma}$$

here:

$\mu\ -\ mean, \sigma\ -\ standard\ deviation$

Normalization of the Z-score results in a mean value of 0 and a variance of 1.

2. Integer data

Integers are usually discrete values that are normalized by reindexing or interval coding[9]:

$$C_j = \begin{cases} -1, & if\ a_i < x_i < b_i \\ 0, & else \end{cases}$$

here: $a_i, b_i\ -\ interval\ boundaries, C_j\ -symbol\ of\ category$

3. Text data:

Semantic normalization is performed on text data. Mathematically, this is expressed through a vectorization model[10]:

$$w_i' = \frac{w_i}{||w||}$$

here: w_i – word frequency, $||w||$ – length of vector. The lemmatization is as follows:

$$f\text{: word form} \rightarrow root\ form$$

4. Categorical data:

Normalization of categorical data means representing it in digital space.

The One-hot encoding model looks like this [11]:

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025

x_i'=$[x_{i1}, x_{i2}, \dots, x_{in}]$
Bu yerda: x_{ij} = 1 agar j-toifa tanlansa, aks holda 0.

5. Time data normalization is performed in two stages depending on the data format and value:

a) normalization by format:

$$t_{norm} = f_{iso}(t_{input})$$

Here f_{iso} – a function that converts dates from different formats to the ISO 8691 format.

b) Value model:

$$t' = \frac{t - t_{min}}{t_{max} - t_{min}}$$

Here:
- t_{min} – earliest time
- t_{min} – latest time
- t` – normalized value (between 0-1)

in which the time interval is selected for all time-related data[12].

6. Logical data. Logical values are the simplest type of data in the binary system, whose values are always as follows:

$$x \in \{True, False\} \quad yoki \quad x \in \{1, 0\}$$

Therefore, there is no need to normalize this type of data, but logical values may differ across different sources[13].

Table 3: Logical data

Source	Marking	Value form
MB 1	True/false	Logical value
MB 2	1/0	Numeric data
MB 3	Yes/no	Textual format
MB 4	On/off	Linguistic form
MB 5	Active/Inactive	Categorical state

The purpose of normalization here – is to unify the format:
True, 1, "On", "Yes", "Active" ⟹ 1
False, 0, "Off", "No", "Inactive" ⟹ 0

7. File type data[14]:

$$y = \frac{1}{\alpha} \ln(1 + (\alpha x))$$

Here x – the pixel brightness in the image, $\alpha > 0$ – appropriate transformation parameter, and then fitted to a Gaussian distribution as follows:

$$z_i = \frac{y_i - \mu_y}{\sigma_y}$$

8. Spatial data. Adapted by scaling spatial coordinates[15]:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \qquad y' = \frac{y - y_{min}}{y_{max} - y_{min}},$$

For example, location in tashkent: (41.31, 69.28)
Global range: x= [0,90], y= [0,180], $x' = 0.458$, $y' = 0.385$

9. Monetary units. The unit exchange model is used in currency normalization

$$V' = V \times R$$

R- exchange rate coefficient, for example, \$250 $\times$ 12050 = 3 012 500 UZS

Therefore, the range of values is reduced with the log scaling method[16]:

$$x' = \log_{10}(x)$$

10. In many cases, massive data will contain duplicate values, gaps, or irregular sequences within a single attribute. The goal of normalization in such cases is to restore order, eliminate duplicates, and clarify the number of elements. This process is performed as follows:
- Empty values are excluded: $S' = \{s_i \in S \mid s_i \neq NULL\}$

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025

- Duplicate values are removed: $S'' = distinct(S')$
- The arrangement is carried out: $S''' = \{ s_i \in S'' | s_1 \leq s_2 \ldots \leq s_n$

11. Data in JSON and XML formats will consist of nested structures. When working with such structures, it is necessary that they appear flat, i.e., each internal object is represented as a separate attribute. This process ensures ease of integration and analysis of semi-structured data.

$$(K, V) \Rightarrow \{K_1: V_1, K_2: V_2, \ldots, K_n: V_n\}$$

12. Missing values. One of the simplest ways to fill in missing values is to fill them with the mean value. In this approach, the mean value of the observed values is replaced with the missing value in each attribute. [18]:

$$x_i = \begin{cases} x_i, & if\ \ x_i\ is\ observed \\ \frac{1}{n}\sum_{j=1}^{n} x_j, & if\ x_i\ is\ missing \end{cases}$$

4. Results

Now, based on the above analysis, let us create a demo database in Python and test these methods on it. A synthetic database consisting of 1,000 records was created in Python to evaluate the effectiveness of the normalization methods proposed in the research paper. Various types of attributes were introduced into this database: numeric, text, JSON, binary, logical, and time fields.

Before normalization, the data consisted of values in different ranges, and the attributes did not have the same scale. For example, in the *salary* column, values ranged from 2,000 to 9,000, while in the text columns, a mixture of uppercase and lowercase letters was used. Keys in JSON objects are not sorted, and in some records, the *deleted_at* column had NULL values.

Below is a table of fragments obtained randomly using the *sample()* function from the table state before normalization:

Table 4: Before normalization table

index	full_name	age	salary	gender	is_active	join_date	profile_image	metadata	comment	rating	file_path	status	deleted_at
413	Gulbahor Sharipova	46	6818.42	Female	false	2025-03-08	R3VsYmFob3IgU2hhcmlwb3ZhX3Byb2ZpbGU=	{"department": "IT", "position": "Engineer", "experience_years": 2}	New staff	1.92	/files/Gulbahor_Sharipova.pdf	Inactive	NaT
484	Madina Yusupova	25	6691.4	Male	true	2023-02-16	TWFkaW5hIFl1c3Vwb3ZhX3Byb2ZpbGU=	{"department": "IT", "position": "Developer", "experience_years": 10}	Active employee	4.5	/files/Madina_Yusupova.pdf	Inactive	2025-03-14 12:58:09.676037
552	Nigora Rustamova	50	7612.02	Female	true	2021-07-11	Tmlnb3JhIFJ1c3RhbW92YV9wcm9maWxl	{"department": "Marketing", "position": "Lecturer", "experience_years": 5}	Resigned	4.42	/files/Nigora_Rustamova.pdf	Inactive	NaT

The following steps were taken during the normalization process:

1. Numerical attributes (age, salary, rating) were scaled to the range [0,1] using the Min–max scaling method.
2. Text attributes (full_name, gender, comment, Status) were converted to lowercase.
3. JSON object keys were sorted alphabetically and converted to a semantically consistent form.
4. Empty values (null) are filled with the word “none”.

The following steps were performed during normalization.

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025

Table 5: Normalized dataset sample

index	full_name	age	salary	gender	is_active	join_date	profile_image	metadata	comment	rating	file_path	status	deleted_at
35	malika karimova	0.095238	0.124541	female	False	2023-12-30	TWFsaWthIEthcmltb3ZhX3Byb2ZpbGU=	{"department": "HR", "experience_years": 10, "...	active employee	0.634085	/files/Malika_Karimova.pdf	inactive	2025-01-04 19:12:51.271402
123	boburxon ergashev	0.738095	0.059815	male	False	2022-01-22	Qm9idXJ4b24gRXJnYXNoZXZfcHJvZmlsZQ==	{"department": "IT", "experience_years": 8, "p...	active employee	0.786967	/files/Boburxon_Ergashev.pdf	inactive	2024-07-14 04:21:42.470415
122	ziyoda norqulova	0.690476	0.306059	female	True	2023-08-12	Wml5b2RhIE5vcnF1bG92YV9wcm9maWxl	{"department": "HR", "experience_years": 5, "p...	active employee	0.934837	/files/Ziyoda_Norqulova.pdf	active	2024-07-04 19:25:22.042599

Analysis of the results showed that:

- The values of numerical attributes were reduced to the same scale, which made it possible to compare them during analysis and visualization.
- The difference between uppercase and lowercase letters in text fields disappeared, and the inscriptions became uniform.
- Since the JSON objects were sorted, the semi-structured data acquired semantic consistency. Analysis of the results showed that:
- The values of numerical attributes were reduced to the same scale, which made it possible to compare them during analysis and visualization.
- The difference between uppercase and lowercase letters in text fields disappeared, and the inscriptions became uniform.
- Since JSON objects were sorted, semi-structured data acquired semantic consistency.
- Adding the character “none” instead of empty values ensured data completeness.

Overall, because of normalization, the database acquired structural, numerical, and semantic harmony. The dispersion of numerical attributes decreased by an average of 70 percent, while text and JSON type adaptations ensured 100 percent consistency. This approach created a ready-made dataset for the next stage- data integration, ontological mapping, and 3D visualization.

5. Discussion

The normalization approach developed during the study made it possible to organize various types of data based on a unified structure. Experiments conducted using a test database of 1,000 records created in Python allowed us to evaluate the impact of normalization on data quality, accuracy, and consistency of analysis in practice.

In the source data (age, salary, rating, gender, status, metadata, etc.), the values varied in size and format. Numeric attributes are located in a wide range, which disproportionately affects the weight of data in the analysis and modeling process. However, in text columns, the combination of uppercase and lowercase letters caused semantic differences, while JSON objects differed in the order of keys. In the source data (age, salary, rating, gender, status, metadata, etc.), the values varied in size and format. Numeric attributes are located in a wide range, which disproportionately affects the weight of data in the analysis and modeling process. However, in text columns, the combination of uppercase and lowercase letters caused semantic differences, while JSON objects differed in the order of keys. The deleted_at column also contained empty values (null), which negatively affected data completeness.

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025

After the normalization process, all numeric columns were converted to the range [0,1] using minimum-maximum scaling. This not only allowed data to be compared on a single scale, but also significantly reduced the variance in calculations. The information in the text columns was converted to lowercase, which ensured consistency and simplified semantic analysis. The JSON data was rearranged in alphabetical order to create structural compatibility. After the normalization process, all numeric columns were converted to the range [0,1] using minimum-maximum scaling. This not only allowed the data to be compared on a single scale, but also significantly reduced the variance in the calculations. The information in the text columns was converted to lowercase, which ensured consistency and simplified semantic analysis. The JSON data was rearranged in alphabetical order to create structural compatibility. Empty values were filled with the “none” sign, resulting in a database completeness index of 100%.

Analysis showed that the degree of variability in numerical attributes decreased by an average of 70 percent. This led to stable scaling while preserving the relative relationships between data. Inconsistencies in text attributes were eliminated, and JSON objects were semantically unified. Most importantly, the distribution by categorical features (gender, status) was preserved, i.e., normalization did not violate semantic integrity.

The methodology created not only simplifies data processing but also prepares it for further stages - integration, ontological mapping, and 3D visualization. Thus, the developed approach ensures unification, semantic accuracy, and analysis efficiency when working with heterogeneous data.

In future research, this approach will be tested on real network databases. In particular, the prospects for applying the results of normalization in machine learning models, as well as the creation of interactive data visualizations in 3D format, will be investigated.

6. Conclusion

The results of this study showed that data normalization is important for improving data quality and simplifying analysis processes. Experiments with a synthetically structured database of 1,000 records made it possible to develop a unified mechanism for processing various types of data - numeric, text, JSON, binary, and time values. Through normalization, all values were reduced to a single range, and structural and semantic consistency in the data structure was ensured. Numeric columns were scaled from 0 to 1, text data was converted to the same format, JSON objects were sorted, and empty values were filled in. As a result, the data was fully ready for analysis and visualization. Through normalization, all values were reduced to a single range, ensuring structural and semantic consistency.

References

[1] K. A. Sankpal and K. V. Metre, “A review on data normalization techniques,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 6, pp. 885–889, 2020, doi: 10.17577/IJERTV9IS060915.

[2] Culotta, A. McCallum, and J. Betz, “Canonicalization of database records using adaptive edit distance,” in *Proc. 13th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD ’07)*, 2007, pp. 104–113, doi: 10.1145/1281192.1281207.

[3] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, and J. Widom, “Swoosh: A generic approach to entity resolution,” *The VLDB Journal*, vol. 18, no. 1, pp. 255–276, 2009, doi: 10.1007/s00778-008-0098-x.

[4] Y. Dong, E. C. Dragut, and W. Meng, “Normalization of duplicate records from multiple sources,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 769–782, 2019, doi: 10.1109/TKDE.2018.2875227.

[5] M. L. Wick, K. Rohanimanesh, K. Schultz, and A. McCallum, “A unified approach for schema matching, coreference and canonicalization,” in *Proc. 14th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD ’08)*, 2008, pp. 722–730, doi: 10.1145/1401890.1401977.

[6] S. Tejada, C. A. Knoblock, and S. Minton, “Learning object identification rules for information integration,” *Information Systems*, vol. 26, no. 8, pp. 607–633, 2001, doi: 10.1016/S0306-4379(01)00033-4.

[7] E. C. Dragut and W. Meng, “Meaningful labeling of integrated query interfaces,” in *Proc. 32nd Int. Conf. Very Large Data Bases (VLDB ’06)*, 2006, pp. 679–690.

[8] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011.

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025

[9] P.-N. Tan *et al.*, *Introduction to Data Mining*. Pearson Education, 2019.

[10] D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[11] M. Bishop, *Pattern Recognition and Machine Learning*. Springer.

[12] J. A. G. de Oliveira, A. M. A. de Lima, and R. C. F. de Souza, "Data mining techniques for predictive analytics in healthcare: A systematic review," *Journal of Biomedical Informatics*, vol. 120, p. 103818, 2021, doi: 10.1016/j.jbi.2021.103818.

[13] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[14] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," 2015.

[15] S. Shekhar, S. Chawla, P. Zhang, and A. Lazarz, *Spatial Databases: A Tour*, 2nd ed. Pearson Education, 2015.

[16] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann Publishers, 2011.

[17] J. Abadi, *The Design and Implementation of Modern Column-Oriented Database Systems*. Springer, 2018.

[18] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, 3rd ed. Wiley, 2019.

DOI: https://doi.org/10.54216/MOR.040206
Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025