Metaheuristic Optimization Review

♦ ISSN: 2692-4048 Vol: 04, No. 02, 2025 ♦ PP. 30–34

Article DOI: https://doi.org/10.54216/MOR.040206



REVIEW ARTICLE

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

Emails: mbbahodir@tsue.uz · z.norqulova@tsue.uz

Received: February 27, 2025 Revised: March 29, 2025 Accepted: May 24, 2025 ⋆ Corresponding author

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. For each type of data, a normalization approach is selected that corresponds to it. In particular, we study min–max scaling and Z-score standardization methods for digital data, one-hot and label encoding for categorical attributes, as well as lemmatization and Unicode-based normalization 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 in 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 heterogeneous data. 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 numerical data. Single-encoding methods are common in categorical attributes, while for text data, methods such as lemmatization, stemming, and Unicode normalization are used. For time data, ISO 8601 formatting or conversion to timestamp format is used. However, each of these approaches has 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 not only simplifies the integration process, but also prepares high-quality information for the next stage: ontological mapping

© 2025 The Author(s). Published by ASPG. This article is licensed under CC BY 4.0.

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 in working with large volumes of data. Researchers have proposed various approaches to simplify the process of combining and analyzing data with different formats, units, and semantics by bringing them into a single form.

Kalayani Sankpal and K. V. Metre, in "An Overview of Data Normalization Methods" [1], analyze various levels and methods of data normalization. Entity resolution and record canonicalization have also been studied in adaptive edit-distance models [2], generic entity-resolution frameworks [3], duplicate-record normalization from multiple sources [4], and unified schema matching and coreference models [5]. Object normalization systems compare attributes using string ratings and user confidence ratings [6], while meaningful label normalization improves semantic consistency in integrated query interfaces [7].

Value-level normalization 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 natural language processing are often used. Literature analysis shows that there is no single universal approach to data normalization. Each study performs normalization at a level appropriate to its purpose, as summarized in Table 1.

As can be seen from Table 1, normalization approaches differ in stage and purpose. Record-level approaches identify and combine repetitive data, field-level approaches lead to uniformity of formats and units of measurement, and value-level approaches ensure semantic consistency of data.

3. METHODOLOGY

Since the types of data found in network databases are diverse, the main types of data are analyzed. Network data contains attributes in various formats, and the data types in Table 2 were selected as research material.

Based on these data types, normalization methods are selected individually in accordance with each type and task.

3.1 Numerical Data

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

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where x_i is the real value, $\min(x)$ and $\max(x)$ represent the value range, and x_i' is the normalization result. An alternative model is Z-score normalization:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{2}$$

where μ is the mean and σ is the standard deviation. Z-score normalization produces a mean value of 0 and variance of 1.

3.2 Integer Data

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

$$C_j = \begin{cases} -1, & \text{if } a_i < x_i < b_i, \\ 0, & \text{else}, \end{cases} \tag{3}$$

where a_i and b_i are interval boundaries, and C_j is the category symbol.

3.3 Text Data

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

$$w_i' = \frac{w_i}{\|w\|} \tag{4}$$

where w_i is word frequency and $\|w\|$ is vector length. Lemmatization is expressed as:

$$f : \text{word form} \rightarrow \text{root form} \tag{5}$$

3.4 Categorical Data

Normalization of categorical data means representing it in digital space. The one-hot encoding model is [11]:

$$x_i' = [x_{i1}, x_{i2}, \ldots, x_{in}] \tag{6}$$

where $x_{ij} = 1$ if the j-th category is selected, and 0 otherwise.

3.5 Time Data

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

$$t_{\text{norm}} = f_{\text{iso}}(t_{\text{input}}) \tag{7}$$

where f_{iso} converts dates from different formats to ISO 8601 format. Value normalization is:

$$t' = \frac{t - t_{\min}}{t_{\max} - t_{\min}} \tag{8}$$

where $t_{\min}$ is the earliest time, $t_{\max}$ is the latest time, and t' is the normalized value between 0 and 1 [12].

3.6 Logical Data

Logical values are the simplest type of data in the binary system:

$$x \in \{\text{True}, \text{False}\} \quad \text{or} \quad x \in \{1, 0\} \tag{9}$$

There is no need to normalize this type of data, but logical values may differ across sources [13]. Therefore, the purpose of normalization is to unify the format, as shown in Table 3.

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 and storing data in a unified format	"Aliyev A." and "A. Aliyev" are 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" and "Tashkent State Economical University" are standardized as the same value
Value-level	Linguistic and semantic adaptation of each value	Eliminating synonyms, abbreviations, and spelling differences and creating semantic harmony	"prof.", "profesor", and "Profesor" are rewritten in one form

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, type, category
5	Date / Time / Timestamp / Datetime	Stores dates, times, or time intervals	2025-11-03, 12:30:00	Logs, date of birth, process time
6	Boolean / Bit	Yes/no or true/false values	True, False, 1, 0	Conditions, circumstances, activity indicator
7	Blob / Binary / Image / JSON	Non-text objects: image, video, document, or structured information	01001011, JSON file, .jpg, .pdf	Multimedia, configuration, file storage
8	Spatial / Geometry / Point	Spatial coordinates or geometric information	(41.3111, 69.2797)	Location, GIS, network topology
9	Money / Currency	Monetary values in different currencies	15000.00, $200	Accounting, transactions, economic analysis
10	Array / Set / List	Multi-valued or ordered attributes	{1, 2, 3}, {A, B, C}	Multi-valued attributes
11	JSON / XML / Object	Semi-structured nested information	{"name":"Ziyoda", "age":25}	Name, place, text, code, description
12	NULL	Missing or unavailable value	NULL	Status of any type

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

Thus, True, 1, "On", "Yes", and "Active" are mapped to 1, while False, 0, "Off", "No", and "Inactive" are mapped to 0.

3.7 File Type Data

For file type data, image brightness can be transformed as follows [14]:

$$y = \frac{1}{\alpha}\ln(1+\alpha x) \tag{10}$$

where x is pixel brightness and $\alpha > 0$ is the transformation parameter. It is then fitted to a Gaussian distribution:

$$z_i = \frac{y_i - \mu_y}{\sigma_y} \tag{11}$$

3.8 Spatial Data

Spatial coordinates are adapted by scaling [15]:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad y' = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \tag{12}$$

For example, a location in Tashkent is $(41.31, 69.28)$. With global ranges $x = [0, 90]$ and $y = [0, 180]$, the normalized coordinates are $x' = 0.458$ and $y' = 0.385$.

3.9 Monetary Units

The unit exchange model is used in currency normalization:

$$V' = V \times R \tag{13}$$

where R is the exchange-rate coefficient. For example, $\$250 \times 12050 = 3,012,500$ UZS. The range of values can then be reduced with log scaling [16]:

$$x' = \log_{10}(x) \tag{14}$$

3.10 Arrays, JSON/XML, and Missing Values

Massive data can contain duplicate values, gaps, or irregular sequences within a single attribute. The goal of normalization is to restore order, eliminate duplicates, and clarify the number of elements:

$$S' = \{s_i \in S \mid s_i \neq \text{NULL}\}, \tag{15}$$

$$S'' = \text{distinct}(S'), \tag{16}$$

$$S''' = \{s_i \in S'' \mid s_1 \leq s_2 \leq \cdots \leq s_n\}. \tag{17}$$

Data in JSON and XML formats consist of nested structures. Flattening is expressed as [17]:

$$(K, V) \Rightarrow \{K_1 : V_1, K_2 : V_2, \ldots, K_n : V_n\} \tag{18}$$

Table 4. Before normalization table

index	full_name	age	salary	gender	active	join	image	metadata	comment	rating	file	status	deleted
413	Gulbahor Sharipova	46	6818.42	Female	false	2025-03-08	R3Vs…	{department IT, position: Engineer, experience_years: 2}	New staff	1.92	files/Gulbahor_Sharipova.pdf	Inactive	NaT
484	Madina Yusupova	25	6691.40	Male	true	2023-02-16	TWFk…	{department: IT, position: Developer, experience_years: 10}	Active employee	4.50	files/Madina_Yusupova.pdf	Inactive	2025-03-14
552	Nigora Rustamova	50	7612.02	Female	true	2021-07-11	Tmln…	{department Marketing, position: Lecturer, experience_years: 5}	Resigned	4.42	files/Nigora_Rustamova.pdf	Inactive	NaT

Table 5. Normalized dataset sample

index	full_name	age	salary	gender	active	join	image	meta	comment	rating	file	status	deleted
35	malika karimova	0.095238	0.124541	female	False	2023-12-30	TWFs…	{dep HR, experience_ 10, … }	active employee	0.634085	files/Malika_Karimova.pdf	inactive	2025-01-04
123	boburxon ergashev	0.738095	0.059815	male	False	2022-01-22	Qm9i…	{depa IT, experience_years: 8, … }	active employee	0.786967	files/Boburxon_Ergashev.pdf	inactive	2024-07-14
122	ziyoda norqulova	0.690476	0.306059	female	True	2023-08-12	Wml5…	{dep HR, experience_ 5, … }	active employee	0.934837	files/Ziyoda_Norqulova.pdf	active	2024-07-04

One of the simplest ways to fill missing values is mean imputation [18]:

$$x_i = \begin{cases} x_i, & \text{if } x_i \text{ is observed,} \\ \frac{1}{n}\sum_{j=1}^{n} x_j, & \text{if } x_i \text{ is missing.} \end{cases} \tag{19}$$

4. RESULTS

Based on the above analysis, a demo database was created in Python and the methods were tested on it. A synthetic database consisting of 1,000 records was created to evaluate the effectiveness of the proposed normalization methods. 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 were not sorted, and in some records, the `deleted_at` column had NULL values. Table 4 shows fragments obtained randomly using the `sample()` function before normalization.

The following steps were taken during the normalization process: numerical attributes (age, salary, rating) were scaled to the range $[0, 1]$ using min–max scaling; text attributes (full name, gender, comment, status) were converted to lowercase; JSON object keys were sorted alphabetically and converted to a semantically consistent form; and empty values were filled with the word “none”. Table 5 presents the normalized dataset sample.

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, and others), the values varied in size and format. Numeric attributes were located in a wide range, which disproportionately affected the weight of data in the analysis and modeling process. 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, which negatively affected data completeness.

After the normalization process, all numeric columns were converted to the range $[0, 1]$ using minimum–maximum scaling. This allowed data to be compared on a single scale and significantly reduced variance in calculations. The information in text columns was converted to lowercase, ensuring consistency and simplifying semantic analysis. 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%. This led to stable scaling while preserving 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; normalization did not violate semantic integrity.

The methodology 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 normalization results in machine learning models and creating 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.

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.

[2] A. 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.

[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.

[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.

[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.

[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.

[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.

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

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

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[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.

[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.