






Enhancing Clustering with Automated Quality Assessment of Spanning Tree Method in Real Data Analysis

Rimma S. Zaripova¹^a, Marat G. Nuriev¹^b, Roman R. Galyamov²^c, Albina Yu. Sharifullina²^d,
Oksana A. Ignatjeva³^e

Kazan State Power Engineering University, Kazan, Russia

Kazan National Research Technological University, Kazan, Russia

Kazan (Volga region) Federal University, Kazan, Russia

zarim@rambler.ru, marat_nul@mail.ru, Roman-grr@mail.ru, albina.040491@gmail.com, dip789@mail.ru

Keywords: clustering, spanning tree method, minimal spanning tree, Prim's algorithm, Davies-Bouldin Index, SV Index, real data analysis, automated quality assessment, parallel algorithms, data visualization.


Abstract: This paper presents the development of an automated system for assessing the quality of the spanning tree method in clustering real data tuples. The system is designed to enhance the efficiency and accuracy of data clustering, allowing researchers and data analysts to uncover meaningful patterns within large datasets. The software reads data from Excel files, constructs a minimal spanning tree using Prim's algorithm, clusters the tree, and evaluates clustering quality using the Davies-Bouldin Index and the SV Index. The program's user-friendly interface simplifies the processes of loading data, constructing index plots, and saving clustering results. Experiments conducted on different datasets demonstrate the program's accuracy in reading data, saving clustering variants, and calculating quality indices. The implementation of parallelized algorithms for constructing the minimal spanning tree significantly improves performance, enabling the processing of large datasets efficiently. This automated system provides a robust tool for the quality assessment of the spanning tree method in clustering real data tuples, with broad applicability across various domains such as medicine, biology, physics, and economics. Future enhancements may include additional clustering methods, improved visualization features, and a web-based version of the program to facilitate global collaboration among researchers and analysts.


1 INTRODUCTION


In today's world, where data volumes are continuously growing, the need for automated quality assessment of clustering methods is becoming increasingly important. One of the most effective clustering techniques is the spanning tree method, which is particularly useful for clustering real data tuples (Zhang, Pang, Zhang, Du, 2023; Muppoori, Mothukuri, 2021). Developing a software system for the automated quality assessment of the spanning tree method for clustering real data tuples is highly relevant. This system enables researchers and data


analysts to quickly and accurately evaluate the quality of clustering, thus allowing them to make more informed decisions based on the results obtained (Sun, Wang, Zhang, Hu, Sun, Lu, 2022).


Data clustering is a fundamental method of data analysis that reveals hidden patterns and structures in large datasets. Among various clustering methods, the spanning tree method stands out for its efficiency and effectiveness. However, to fully utilize the potential of the spanning tree method, it is essential to accurately assess its quality. This paper proposes the development of an automated system for the quality assessment of the spanning tree method using metrics such as the SV index and the Davies-Bouldin index

^a  <https://orcid.org/0000-0002-3548-1866>

^b  <https://orcid.org/0009-0003-0741-1734>

^c  <https://orcid.org/0009-0009-8736-7051>

^d  <https://orcid.org/0009-0008-5830-2702>

^e  <https://orcid.org/0000-0002-9887-2901>

(Rojas Thomas, Peñas, Mora, 2013; Vergani, Binaghi, 2018).

The challenge with clustering real data tuples lies in selecting the appropriate clustering method and evaluating the quality of the formed clusters. Real data tuples consist of multiple numerical features, making it crucial to choose a method that can handle this complexity and ensure meaningful cluster formation (Zheng, Wang, Xu, Pan, Zhang, 2023). The primary goal of this study is to automate the quality assessment of the spanning tree method for clustering real data tuples and to determine the optimal number of clusters. To achieve this goal, it is necessary to implement a program for loading real data tuples from an Excel file, develop the spanning tree method for clustering the data into the optimal number of clusters, create algorithms for evaluating the clustering quality using the Davies-Bouldin index and the SV index (Saha, Bandyopadhyay, 2009; Viktorov, Gibadullin, 2025), and save the selected clustering model and evaluation results in an Excel file.

The development of an automated system for quality assessment of the spanning tree method has multiple advantages. It provides researchers and data analysts with a powerful tool to automatically evaluate clustering quality using various metrics, make quick and accurate decisions regarding the optimal number of clusters, and apply the method across various fields such as medicine, biology, physics, and economics (Uteyev, Gibadullin, 2024; Gizatullin, Gizatullin, Nuriev, 2024). In practice, this system can be used in various domains, including classifying diseases and identifying hidden signs in medicine, analyzing experimental data and uncovering patterns in system behaviors in physics, and segmenting markets and determining consumer preferences in economics (Gibadullin, Baimukhametova, Perukhin, 2019).

The methodology involves several steps. First, a program is developed to load real data tuples from an Excel file, considering specific constraints to ensure compatibility with the application. Second, the spanning tree method for clustering real data tuples into any possible number of clusters is implemented using Prim's algorithm for constructing the minimal spanning tree (Chopade, More, 2017). Third, the SEMST method is applied for clustering the minimal spanning tree into clusters (Biswas, Paul, Bhattacharya, 2015). Fourth, the clustering quality is evaluated using the Davies-Bouldin index and the SV index, and the results are displayed in a DataGridView component for easy analysis. Finally, the chosen clustering model and evaluation results are

saved in an Excel file for further reference (Ying Bai, 2010).

The expected outcome of this study is a robust software system that automates the quality assessment of the spanning tree method for clustering real data tuples. This system will significantly reduce the time required for analyzing clustering quality, thereby enhancing the efficiency and effectiveness of data analysis processes across various domains. The development of an automated system for the quality assessment of the spanning tree method for clustering real data tuples has substantial theoretical and practical significance. It provides a reliable tool for researchers and data analysts, enabling them to make well-informed decisions based on accurate clustering quality evaluations. The potential application of this system spans multiple fields, making it a versatile and valuable contribution to the data analysis domain (Liu, Shi, 2023; Gibadullin, Vershinin, Minyazev, 2018).

2 THEORETICAL FOUNDATIONS

In the realm of data analysis, clustering stands out as a crucial method that enables the grouping of data objects into clusters based on their similarities or differences. This process is particularly challenging when dealing with real data tuples, which consist of multiple numerical features. The complexity of real data tuples necessitates the selection of an appropriate clustering method and the accurate evaluation of the resulting clusters' quality.

The spanning tree method is a widely recognized approach for clustering real data tuples. This method involves constructing a minimal spanning tree (MST) that connects all the vertices of a graph without forming any cycles and ensures the minimum possible sum of edge weights. The concept of using MST for clustering was first introduced by C. T. Zah in 1971. The spanning tree method is advantageous for its efficiency and effectiveness in revealing hidden patterns within large datasets. However, its full potential can only be realized through accurate quality assessment.

To address the challenges associated with clustering real data tuples, this study proposes the development of an automated system for the quality assessment of the spanning tree method, utilizing metrics such as the SV index and the Davies-Bouldin index. The primary objective is to automate the assessment process to determine the optimal number of clusters, thereby ensuring meaningful and accurate cluster formation.

The core issue with clustering real data tuples lies in selecting a suitable method and accurately evaluating the quality of the clusters formed. Real data tuples, characterized by multiple numerical features, present a significant challenge in terms of complexity and variability. An effective clustering method must be capable of handling this complexity while ensuring that the resulting clusters accurately reflect the structure of the original data.

Internal measures of clustering quality are essential for determining how well a clustering algorithm groups similar objects without external information. These measures provide insights into the compactness and separation of clusters, which are critical for assessing clustering performance. Some of the widely used internal quality assessment indices include the Silhouette Index, the Davies-Bouldin Index (DBI), and the Dunn Index.

The Silhouette Index evaluates clustering quality by considering both intra-cluster and inter-cluster distances. It aims to maximize the silhouette coefficients for all objects, thereby ensuring optimal clustering. The Davies-Bouldin Index assesses clustering quality by calculating the ratio of intra-cluster distances to inter-cluster distances. A lower DBI indicates better clustering quality, as it reflects more compact and well-separated clusters. The Dunn Index focuses on minimizing intra-cluster distances while maximizing inter-cluster distances, thus ensuring that clusters are both compact and distinct.

The spanning tree method is a powerful algorithm for solving clustering tasks using a minimal spanning tree. An MST is a subgraph that includes all vertices of the graph and is a tree, meaning it contains no cycles. The MST is constructed in such a way that the sum of the edge weights is minimized. This method can be applied to various optimization tasks, such as route planning, transportation system design, and energy expense optimization.

Several distance metrics can be utilized in the spanning tree method, each with its advantages and specific use cases. The Euclidean Distance is the most commonly used metric, measuring the straight-line distance between two points in multi-dimensional space. It is precise for measuring direct distances and is widely applicable in various domains. The Manhattan Distance (City-Block Distance) measures the distance between two points by moving along the axes at right angles. This metric is particularly useful in urban planning and transport routing tasks, where movements are constrained along grid-like patterns. The Chebyshev Distance, also known as the maximum metric, measures the maximum difference between the corresponding coordinates of points. It is

useful in scenarios where the maximum deviation in any one dimension is critical. The Bray-Curtis Distance measures dissimilarity between two points by considering the sum of the absolute differences divided by the sum of the absolute values of the coordinates. This metric is effective in ecological and environmental studies.

Each of these metrics has its strengths and is suitable for different types of data and clustering tasks. For instance, the Euclidean distance is preferred for its accuracy in measuring direct distances, while the Manhattan distance is more applicable in tasks involving movements along coordinate axes.

Several algorithms can be used to construct the minimal spanning tree, each with its own advantages and computational efficiencies. Prim's Algorithm is a widely used method for finding the MST in a weighted undirected graph. It starts from an arbitrary vertex and iteratively adds the minimum weight edge connected to the visited vertices until all vertices are included in the MST. The algorithm is effective for its simplicity and computational efficiency, making it suitable for a wide range of applications. Prim's Algorithm can be summarized as follows: an arbitrary starting vertex is selected and added to the set of visited vertices. The algorithm then finds all edges connected to the visited vertices and chooses the one with the minimum weight, adding the new vertex connected by the chosen edge to the set of visited vertices. This process is repeated until all vertices are visited, resulting in the MST.

The development of an automated system for the quality assessment of the spanning tree method can be facilitated by various development environments. Visual Studio is a popular choice, offering comprehensive tools for code editing, debugging, and interface design. JetBrains Rider is another effective environment, particularly for developing applications in C#, F#, and Visual Basic, known for its code assistance and debugging tools. Visual Studio Code is a lightweight, free, and cross-platform code editor that supports multiple programming languages, making it versatile for various development tasks.

Each development environment provides unique features and capabilities that can enhance the efficiency and effectiveness of the development process. Visual Studio offers a robust set of tools and integrations, making it ideal for large-scale projects. JetBrains Rider provides advanced code analysis and navigation features, which can significantly improve productivity. Visual Studio Code, with its extensive library of extensions and plugins, offers a

customizable development experience that can be tailored to specific project needs.

The theoretical foundations of the spanning tree method and the associated clustering quality assessment indices provide a comprehensive framework for developing an automated system for clustering real data tuples. By understanding the complexities and requirements of clustering, the selection of appropriate distance metrics, and the implementation of efficient algorithms, it is possible to create a robust system that enhances the accuracy and reliability of clustering outcomes. The development environments play a crucial role in facilitating the implementation and refinement of this system, ensuring that it meets the needs of researchers and data analysts across various domains.

3 PROGRAM DEVELOPMENT

The development of a software system for loading real data tuples from an Excel file into the program involved two primary tasks: establishing constraints for the Excel file and creating an algorithm for loading the data tuples into the program. These tasks were crucial to ensure the program's compatibility and functionality.

Setting constraints for the Excel file was essential because the program could not adapt to every possible file format. The constraints were as follows:

- 1) Data should start from cell A1 on the first sheet.
- 2) Columns must be labeled at the top.
- 3) Data should be organized sequentially from top to bottom.
- 4) Only the first sheet of the Excel file should contain data.
- 5) No extraneous information should be present on the sheet.

These constraints ensured that the program could reliably read the data from the Excel file without encountering format-related issues.

To develop the algorithm for loading real data tuples from an Excel file into the program using C#, the `Microsoft.Office.Interop.Excel` NuGet package was employed. This package is a key component of the .NET ecosystem for interacting with Excel files. The program used the following steps to load the data:

- 1) Open the Excel application:

```
Excel.Application ObjWorkExcel = new Excel.Application();
```
- 2) Open the Excel file using `OpenFileDialog` to specify the file path:

```
OpenFileDialog ofd = new OpenFileDialog();
ofd.DefaultExt = "*.xlsx";
```

```
ofd.Filter = "Microsoft Excel (*.xls*)|*.xls*";
ofd.Title = "Select Excel Document";
if (ofd.ShowDialog() != DialogResult.OK)
{
    MessageBox.Show("No file selected", "Data Loading...",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    return;
}
Excel.Workbook ObjWorkBook =
ObjWorkExcel.Workbooks.Open(ofd.FileName);
3) Access the first sheet of the Excel file:
Excel.Worksheet ObjWorkSheet =
(Excel.Worksheet)ObjWorkBook.Sheets[1];
4) Determine the range of used cells:
Excel.Range ShtRange = ObjWorkSheet.UsedRange;
int rowCount = ShtRange.Rows.Count;
int colCount = ShtRange.Columns.Count;
5) Read the data from the Excel file and store it in a suitable data structure.
```

The algorithm reads the data from the Excel file, checks for duplicates, and stores the unique data tuples for further processing.

The implementation of the spanning tree method for clustering real data tuples involves two main tasks: constructing the minimal spanning tree and clustering the MST into the desired number of clusters.

The first step is to calculate the edge weights between the data objects using the Euclidean metric. The edge weights are then used to construct the MST through Prim's Algorithm, which iteratively adds the vertex with the minimum edge weight to the spanning tree. The algorithm can be summarized as follows:

- 1) Initialize the MST with an arbitrary starting vertex.
- 2) Repeatedly add the smallest edge connecting a vertex in the MST to a vertex outside the MST.
- 3) Continue until all vertices are included in the MST.

A parallelized version of Prim's Algorithm was implemented to improve performance, using `Parallel.For` loops to calculate edge weights simultaneously. The code snippet below shows the parallelized weight calculation:

```
Parallel.For(0, length - 1, i =>
{
    int a = length - i;
    vershiny[i] = -1;
    weight[i] = new float[a - 1];
    for (int j = 1; j < a; j++)
    {
        for (int t = 0; t < CountColomum; t++)
        {
            weight[i][j - 1] += (float)Math.Pow(list[i, t]
                - list[j + i, t], 2);
```

```

    }
    weight[i][j - 1] = (float)Math.Pow(weight[i][j -
1], 0.5);
    }
    _excel_table[i] = new float[CountColomum];
    for (int t = 0; t < CountColomum; t++)
    {
        _excel_table[i][t] = list[i, t];
    }
});

```

The MST is clustered using the SEMST method, which involves sequentially removing edges with the maximum weights. This process divides the MST into clusters based on the desired number of clusters. The steps are as follows:

- 1) Sort all edges by weight in descending order.
- 2) Remove the top edges until the number of clusters is achieved.
- 3) Store each clustering configuration in an array of arrays, 'clustering_vershiny'.

The developed system uses the Davies-Bouldin Index and the SV Index to evaluate clustering quality. The 'Qualite_control' method calculates these indices. The method initializes several arrays and variables necessary for the calculations, ensuring accurate quality assessments.

The final task involves displaying the clustering results and saving the chosen clustering model to an Excel file. The user selects the desired clustering configuration from a DataGridView, and the program saves the data in the specified format.

The development of this software system automates the quality assessment of the spanning tree method for clustering real data tuples, significantly enhancing the efficiency and accuracy of data analysis processes. The detailed implementation ensures compatibility with Excel files, efficient MST construction, effective clustering, and reliable quality assessment.

4 EXPERIMENTS AND RESULTS

Any program must be thoroughly tested for functionality after it has been written. In programming, unforeseen errors often occur during testing, which need to be corrected. Therefore, experiments were conducted on two datasets with tuples containing four and three values, respectively. The program is automated and can independently determine the number of values in a tuple, significantly simplifying its usage. The success of the experiments was evaluated based on several criteria:

- 1) The program reads all data from the Excel file.

2) The program correctly saves the chosen clustering variant.

3) The program accurately calculates the indices.

For the experiments, two sets of real-valued vectors were used. These datasets included one output parameter and several input parameters. In the first experiment, the output parameter was ZN_BLOOD, and the input parameters were S_AREA and ZN_WATER. The total number of data vectors was 392. However, the number of objects might decrease during program execution, as some data vectors may be identical, and the program does not account for duplicate objects.

In the second experiment, the number of input parameters increased by one, where S_AREA was split into two input parameters: HEIGHT and WEIGHT. The total number of vectors remained 392, similar to the first dataset.

During the experiments, the quality indices were analyzed. The values of the indices were normalized to a uniform scale using the formula:

$$x = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

For the Davies-Bouldin Index, the most optimal result was achieved with 10 clusters. The first decline in the index value often indicates the appropriate number of clusters, which was satisfactory for this dataset (Fig. 1). In this case, the SV index results were not necessary. The SV index in this program is primarily used to support the main quality index, Davies-Bouldin.

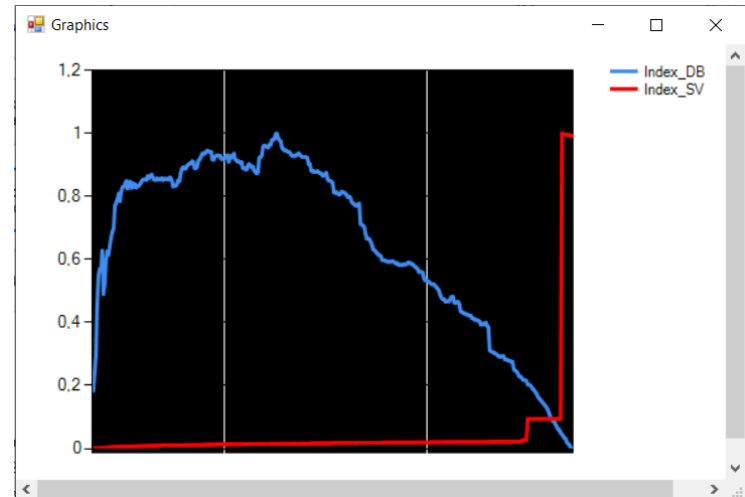


Fig. 1. Graph of index values for the 1st data set.

For the second dataset (Fig. 2), the index values also indicated that the optimal solution was the first decline in the Davies-Bouldin Index, which suggested four clusters.

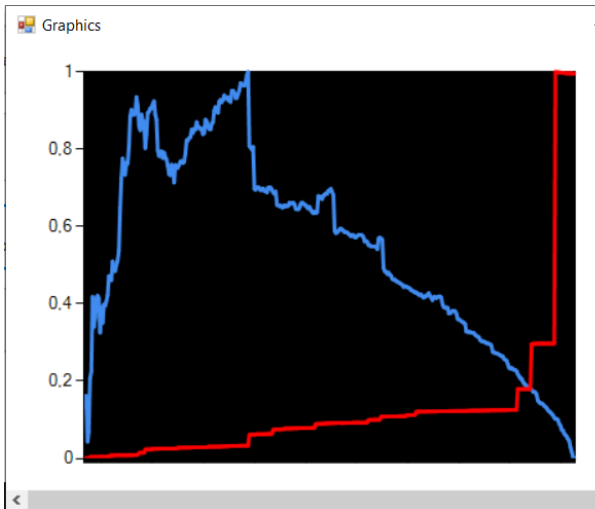


Fig. 2. Graph of index values for the 2nd data set.

These results demonstrate that the program successfully performs the intended tasks. It does not produce errors when used correctly, reads all data from the Excel file, correctly saves the chosen clustering variant, and accurately calculates the indices.

The program interface is straightforward. Upon launching the executable file, the main window of "Clustering" opens. This window contains several buttons and an empty table. The first step is to click the "Load Data" button, which opens the OpenFileDialog window for selecting the Excel file. Once the data is loaded into the table, users can proceed to construct the index plot by clicking the "Build Index Plot" button. The plot displays two lines: the blue line for the Davies-Bouldin Index and the red line for the SV index.

To save a clustering variant, the user selects the row corresponding to the desired number of clusters in the table and clicks the "Save Data" button. The selected clustering variant is saved in an Excel file, maintaining the same order as the original file, with clusters separated by spaces.

The developed program automates the quality assessment of the spanning tree method for clustering real data tuples, enhancing the efficiency and accuracy of data analysis processes. The implementation ensures compatibility with Excel files, efficient construction of the minimal spanning tree, effective clustering, and reliable quality assessment. The experiments conducted confirmed the program's functionality and accuracy in evaluating clustering quality using the Davies-Bouldin and SV indices.

5 CONCLUSION

To conclude, the creation of an automated system for assessing the quality of the spanning tree method in clustering real data tuples addresses a critical need in the field of data analysis. This system significantly enhances researchers' and data analysts' capabilities to perform efficient and accurate clustering, thereby uncovering meaningful patterns and structures within extensive datasets.

The central aim of this research was to automate the quality assessment of the spanning tree method to determine the optimal number of clusters for real data tuples. This goal was accomplished by developing software that can read data from an Excel file, construct a minimal spanning tree using Prim's algorithm, cluster the tree, and evaluate the clustering quality using the Davies-Bouldin Index and the SV Index. Additionally, the program offers features for visualizing clustering results and saving selected clustering models.

Experiments validated the program's effectiveness in handling real datasets. It was tested on two datasets with different configurations, demonstrating its accuracy in reading data, saving clustering variants, and calculating quality indices. Normalizing the index values ensured consistent and reliable quality assessments.

A significant outcome of this research is the development of a user-friendly interface that simplifies data clustering and quality assessment processes. Users can load data, construct index plots, and save clustering results effortlessly, making the program accessible to individuals with varying levels of technical expertise.

The program's flexibility and adaptability to different data types and clustering tasks are enhanced by its support for multiple distance metrics and clustering algorithms. Its capability to handle multiple input parameters and duplicate data vectors increases its applicability in various fields such as medicine, biology, physics, and economics.

The implementation of parallelized algorithms for constructing the minimal spanning tree has markedly improved the program's performance, allowing it to process large datasets efficiently. This improvement ensures the program's utility for real-time data analysis, which is vital in many practical applications.

In summary, the automated system developed in this research offers a robust and reliable tool for the quality assessment of the spanning tree method in clustering real data tuples. Its accuracy, efficiency, and user-friendly interface make it a valuable resource for researchers and data analysts across

various domains. The system's versatility and adaptability to different datasets and clustering tasks ensure its wide applicability and potential for future enhancements.

Future work could focus on expanding the program's capabilities to include additional clustering methods and quality assessment metrics. Improving the program's visualization features and integrating it with other data analysis tools could further enhance its usability and functionality. Developing a web-based version of the program could also increase its accessibility and facilitate collaboration among researchers and analysts worldwide.

REFERENCES

- Zhang, W., Pang, X., Zhang, J., Du, S., 2023. Real time operation data mining algorithm for power systems based on adaptive incremental clustering algorithm. 2023 3rd New Energy and Energy Storage System Control Summit Forum (NEESSC), Mianyang, China, pp. 191-194.
- Muppoori, N., Mothukuri, R., 2021. A Case Study on Clustering Application to Unlabelled Real-time Time-series Data Streams. 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 363-368.
- Sun, J., Wang, P., Zhang, C., Hu, C., Sun, Q., Lu, R., 2022. Data flow adaptive clustering method in data center. 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, pp. 1748-1752.
- Rojas Thomas, J. C., Peñas, M. S., Mora, M., 2013. New Version of Davies-Bouldin Index for Clustering Validation Based on Cylindrical Distance. 2013 32nd International Conference of the Chilean Computer Science Society (SCCC), Temuco, Chile. pp. 49-53.
- Vergani, A. A., Binaghi, E., 2018. A Soft Davies-Bouldin Separation Measure. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Rio de Janeiro, Brazil, 2018, pp. 1-8.
- Zheng, Y., Wang, F., Xu, W., Pan, M., Zhang, P., 2023. Semantic Communications with Explicit Semantic Base for Image Transmission. GLOBECOM 2023 - 2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, pp. 4497-4502.
- Saha, S., Bandyopadhyay, S., 2009. Performance Evaluation of Some Symmetry-Based Cluster Validity Indexes. in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 39, no. 4, pp. 420-425.
- Viktorov, I. V., Gibadullin, R. F., 2025. Machine Learning-Driven Web System for Automated Sequential-to-Parallel Code Transformation. 2025 International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russian Federation, pp. 229-234.
- Uteyev, G., Gibadullin, R. F., 2024. Development of the Decentralized Biometric Identity Verification System Using Blockchain Technology and Computer Vision. 2024 International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russian Federation, pp. 350-355.
- Gizatullin, Z., Gizatullin, R., Nuriev, M., 2024. Analysis of Signal Integrity and Distortion in Twisted Pair in a Multilayer Printed Circuit Board. 2024 International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russian Federation, pp. 356-360.
- Gibadullin, R. F., Baimukhametova, G. A., Perukhin, M. Y., 2019. Service-Oriented Distributed Energy Data Management Using Big Data Technologies. 2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russia, pp. 1-7.
- Chopade, S., More, P., 2017. Effective bug triage with Prim's algorithm for feature selection. 2017 International Conference on Signal Processing and Communication (ICSPC), Coimbatore, India, pp. 217-220.
- Biswas, P., Paul, A., Bhattacharya, P., 2015. Generating spanning tree of non-regular graphic sequences through a variant of Prim's algorithm. 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], Nagercoil, India, pp. 1-4.
- Ying Bai, 2010. Data Selection Query with Visual C#.NET. in Practical Database Programming With Visual C#.NET, IEEE, 2010, pp.235-438.
- Liu, Y., Shi, L., 2023. Domain-Oriented Transformation Method for Big Data Analysis Process Model. 2023 International Conference on Networking, Informatics and Computing (ICNETIC), Palermo, Italy, pp. 109-113.
- Gibadullin, R. F., Vershinin, I. S., Minyazev, R. S., 2018. Development of Load Balancer and Parallel Database Management Module. 2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Moscow, Russia, pp. 1-6.