Journal of Computer Science

Research Article

Adaptive Data Transformation for Enhanced Clustering Performance in Diagnostic Systems

Mohammed Subhi Al-Batah

Department of Computer Science, Jadara University, Jordan

Article history
Received: 08-07-2025
Revised: 12-07-2025
Accepted: 15-07-2025

Abstract: This paper presents an enhanced approach to the Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) algorithm, aimed at improving clustering accuracy in medical data, specifically for breast cancer diagnosis. The proposed method introduces a modified data transformation technique to optimize the original BIRCH algorithm. This transformation refines the clustering process, resulting in significant improvements in diagnostic accuracy. The modified BIRCH algorithm was tested on a breast cancer dataset and achieved a clustering accuracy of 98.40%, a substantial improvement compared to 33.22% accuracy obtained using the original algorithm. Experimental results demonstrate that the use of transformed data not only enhances the performance of BIRCH but also highlights its effectiveness in scenarios with two clusters and a threshold value of two. These findings suggest that data transformation plays a critical role in refining hierarchical clustering algorithms, offering better diagnostic insights in medical applications.

Keywords: Data Transformation, Medical Data Analysis, Clustering Performance, Tumor Classification, Algorithm Modification

Introduction

Garg *et al.* (2006) introduced a parallel version of the BIRCH algorithm called P-BIRCH, aimed at improving scalability without compromising clustering quality. This parallel algorithm operates on the Single Program Multiple Data (SPMD) model, using message passing to communicate between processors. The processors construct local CF trees independently, and the clustering process relies on a parallel k-means algorithm for refining clusters. Experimental results demonstrated that P-BIRCH scales linearly with an increasing number of processors while maintaining the clustering quality comparable to the original BIRCH algorithm. This approach significantly improves BIRCH's ability to handle large-scale data sets efficiently (Al Eiadeh & Al Batah, 2024).

Li and Jie (2013) addressed one of the key limitations of the original BIRCH algorithm—its difficulty in accurately clustering arbitrary-shaped clusters. They proposed an Adaptive Split BIRCH (AS-BIRCH) algorithm, which uses density-based clustering rather than relying solely on Euclidean distances. AS-BIRCH first selects the farthest two CFs and recalculates their minimal node distances to refine cluster formation based on density. This modification enhances clustering accuracy, especially for irregularly shaped data clusters. Simulation results show that AS-BIRCH outperforms the original algorithm in terms of packet loss, delay, and jitter, making it a robust choice for clustering complex data patterns (Al Eiadeh & Al Batah, 2024).

Lei (2016) further enhanced the BIRCH algorithm by modifying its third and fourth phases to improve its performance on time series data. By using Dynamic Time Warping Barycenter Averaging (DBA) in the clustering process and omitting the optional fourth phase, Lei significantly improved clustering accuracy. Comparative experiments using 35-time series datasets revealed that E-BIRCH, the enhanced version, consistently outperforms BIRCH and its variants. The DBA-based approach allowed E-BIRCH to handle large, incremental datasets more effectively than traditional methods such as k-means and its time series counterpart, k-DBA (Al-Batah, 2019).

Lorbeer *et al.* (2017) developed the Automatic BIRCH (A-BIRCH) algorithm, which eliminates the need for preset thresholds or knowledge of the number of clusters. By integrating the Gap Statistic, the authors automatically estimated optimal thresholds based on a small representative subset of data. The algorithm parallelizes the Gap Statistic, allowing for scalable and efficient clustering of large datasets. The study demonstrated that A-BIRCH provides accurate clustering results without requiring manual tuning of parameters,

making it particularly suitable for dynamic and large-scale data clustering (Al-Batah, 2014).

Li *et al.* (2015) introduced a hybrid clustering algorithm combining BIRCH with the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to improve clustering performance on spatial data. The hybrid model, called BIRCH-DBSCAN, leverages BIRCH's speed for initial data compression and DBSCAN's ability to handle noise and irregular cluster shapes. The combination of these algorithms resulted in improved clustering accuracy and scalability for large datasets, showing particularly strong performance in clustering noisy spatial data. The study concluded that this hybrid approach balances the strengths of both algorithms and reduces their individual weaknesses, offering a versatile solution for various data clustering challenges.

Nayak and Mahapatra (2017) presented an extension of BIRCH for handling multidimensional data in distributed environments. Their proposed Multidimensional BIRCH (MD-BIRCH) algorithm modifies the CF tree structure to better accommodate high-dimensional data by adjusting the threshold dynamically during the clustering process. The researchers tested MD-BIRCH on several real-world datasets, showing that it significantly reduces clustering time and improves accuracy for multidimensional datasets compared to traditional BIRCH. This makes MD-BIRCH a powerful tool for applications such as bioinformatics and market segmentation, where multidimensional data are prevalent.

In another significant contribution, Zhang *et al.* (2020) explored the integration of BIRCH with deep learning techniques to improve clustering accuracy for complex, high-dimensional datasets. Their work proposed the Deep BIRCH algorithm, which combines hierarchical clustering with autoencoders to reduce data dimensionality before applying the BIRCH algorithm. The study demonstrated that Deep BIRCH outperforms traditional clustering methods, especially when dealing with non-linear patterns in large datasets. By embedding deep learning techniques, the algorithm effectively captures the underlying structure of the data, leading to more accurate and meaningful cluster formations.

Additionally, Luo and Li (2021) investigated the impact of data preprocessing on the performance of the BIRCH algorithm. Their study introduced a data transformation pipeline that includes normalization, feature selection, and dimensionality reduction techniques prior to applying BIRCH. The experimental results showed that preprocessing significantly enhances the performance of BIRCH, leading to faster convergence and improved clustering accuracy. Their findings highlight the importance of preprocessing in improving the efficacy of BIRCH when dealing with high-dimensional and noisy data (Alkhasawneh *et al.*, 2015).

Recent advances in healthcare applications have also leveraged the BIRCH algorithm. For instance, Ramachandran and Govindan (2022) employed a modified BIRCH algorithm for early detection of breast cancer using mammogram images. The modified algorithm integrated feature selection techniques with BIRCH to improve the clustering of image data, which enhanced the early detection accuracy of breast cancer. The study reported a 20% improvement in diagnostic accuracy compared to traditional clustering methods, underscoring the algorithm's potential in medical diagnosis.

Methodology

This study employs a research methodology designed to enhance the accuracy of the BIRCH clustering algorithm for medical data analysis, with a specific focus on breast cancer diagnosis. The dataset utilized is the Breast Cancer Wisconsin (Diagnostic) Data Set, which is publicly available through Kaggle.

Data Transformation

In statistics and data mining, data transformation refers to the application of a mathematical function to each data point in a dataset. Specifically, each data point zi is replaced by a new value; yi = f(zi), where f represents a transformation function. Data transformation is crucial in preparing datasets for statistical analysis or machine learning algorithms, especially when features vary widely in scale. Standardization, a common transformation technique, ensures that the data conforms to assumptions required for statistical inference procedures, such as normality or linearity, which are essential for accurate analysis and interpretation. It also helps algorithms that rely on distance metrics, such as clustering and classification, to perform optimally by reducing the influence of features with larger numerical ranges.

In this study, the sklearn.preprocessing StandardScaler library was used to standardize the dataset's features. The StandardScaler function ensures that the mean of the data becomes zero and the variance becomes one, thus bringing all features to the same scale. This method is particularly important when employing algorithms such as BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), where the Euclidean distance metric is sensitive to feature magnitude, and large variations between features can disproportionately affect clustering results.

Data Transformation Process

The data transformation process follows the data cleaning phase, where raw data is processed to remove inconsistencies or errors. Once cleaned, the transformation phase consolidates the data into forms suitable for mining by applying techniques such as scaling, normalization, or log transformation. In this

study, data transformation involved several key steps (Ramachandran, & Govindan, 2022).

Data Loading and Exploration: Initial exploratory data analysis (EDA) was performed to understand the data distribution, identify outliers, and determine whether further cleaning was required. Descriptive statistics and visualizations helped identify potential data anomalies.

Scaling and Standardization: After exploration, the StandardScaler function from the sklearn library was used to standardize the dataset. This step ensures that all features are on the same scale, improving the performance of distance-based algorithms.

Data Restructuring and Feature Engineering: Depending on the dataset, feature engineering was applied to create additional informative features or restructure existing ones. This transformation process ensures that the dataset is in a suitable format for clustering and other machine learning tasks.

Data Aggregation: In cases where multiple records need to be consolidated (e.g., temporal or hierarchical data), aggregation operations were applied to summarize the data at appropriate levels, preparing it for clustering.

Modified BIRCH Algorithm

Once the data transformation was complete, the modified BIRCH algorithm was applied. BIRCH is a hierarchical clustering algorithm that efficiently handles large datasets by incrementally building a clustering feature (CF) tree. To further enhance its performance in this study, several modifications were introduced, specifically by incorporating data transformation techniques and adapting the algorithm's distance metric. The clustering process began by determining the optimal number of clusters using the Elbow method and then applying the enhanced BIRCH algorithm (Kumar & Shah, 2021).

The key enhancement introduced in this study involved incorporating the Scale algorithm alongside a modified version of the Euclidean distance metric. These modifications aimed to improve the algorithm's clustering accuracy, especially when working with high-dimensional or non-standardized datasets. The following steps outline the procedure for the modified BIRCH algorithm:

Steps of the Modified BIRCH Algorithm

1. Data Preprocessing and Transformation: This step involves reading the dataset and applying scaling and transformation. The StandardScaler was used to normalize the data, ensuring that all features contribute equally to the clustering process.
2. CF Tree Construction: An initial clustering feature (CF) tree is built in memory. This CF tree captures summaries of the dataset by aggregating the data points into compact, manageable representations, known as CF triples, which store essential information about the clusters.
3. Refinement of the CF Tree: A smaller CF tree is created by pruning and condensing the original CF tree. This step helps reduce memory consumption and optimizes the clustering process by discarding outliers or merging very similar clusters.
4. Global Clustering: In this step, global clustering is performed on the leaf nodes of the CF tree. The clustering algorithm refines the clusters further by analyzing the relationships between CF triples. The global clustering process ensures that clusters with similar characteristics are merged.
5. Cluster Refinement: Finally, the clusters are refined through additional passes over the data to improve clustering results. This involves recalculating the clusters' centroids and adjusting the cluster boundaries to improve the accuracy of the clustering.

Algorithm 1: Modified BIRCH Algorithm

Input:

D = {t1, t2, ..., tn} // Set of elements

Data_clean(); // Clean the dataset

Data_scale(); // Apply scaling and transformation to standardize the data

Output:

D2 = {NT1, NT2, ..., NTn} // The transformed and scaled dataset

T // Threshold value for CF tree construction

Algorithm:

```
For each Nti in D2:
    Determine the appropriate leaf node for Nti insertion;
    If threshold condition is not violated:
        Add Nti to the cluster and update CF triples;
    Else:
        If there is room to insert Nti:
            Insert Nti as a single cluster and update CF triples;
        Else:
            Split the leaf node and redistribute CF features;
```

This pseudocode outlines the procedure for applying the modified BIRCH algorithm to the transformed data. The algorithm dynamically updates CF triples as new data points are inserted and splits clusters when necessary to maintain balanced, accurate clustering.

Performance Evaluation of the Modified BIRCH Algorithm

The performance of the modified BIRCH algorithm was evaluated using a breast cancer dataset. The Elbow

method was used to determine the optimal number of clusters, and the algorithm was tested with varying threshold values.

Results and Discussion

This section presents the experimental results of both the original and modified BIRCH algorithms applied to the breast cancer dataset. Several cluster configurations and threshold values were examined to evaluate the clustering performance. The results demonstrate the effectiveness of the modifications introduced in the BIRCH algorithm.

1. The Original BIRCH Algorithm
2. The Case of Two Clusters and a Threshold of 3

In this scenario, the Elbow method indicated that two clusters would be optimal. The original BIRCH algorithm was applied with a threshold of 3, and the confusion matrix is presented in Table 1.

Table 1: Confusion Matrix for the Original BIRCH Algorithm (2 Clusters, Threshold = 3)

	Predicted B	Predicted M
Actual B	0	357
Actual M	86	126

Table 2: Performance Metrics for the Original BIRCH Algorithm (2 Clusters, Threshold = 3)

Diagnosis	Precision	Recall	F1-Score	Support
B	0.00	0.00	0.00	357
M	0.26	0.59	0.36	212

The accuracy for this case was 22.14%, a very low value, suggesting that the algorithm struggled to correctly cluster the data. This result highlights the need for improvement or adjustment of experimental conditions. The precision, recall, and F1-score metrics further illustrate the poor performance, particularly in predicting benign (B) cases (Table 2).

The Case of Three Clusters and a Threshold of 3

When increasing the number of clusters to three, the results showed a slight improvement. The confusion matrix for this scenario is shown in Table 3.

Table 3: Confusion Matrix for the Original BIRCH Algorithm (3 Clusters, Threshold = 3)

	Predicted B	Predicted M
Actual B	97	260
Actual M	120	92

Table 4: Performance Metrics for the Original BIRCH Algorithm (3 Clusters, Threshold = 3)

Diagnosis	Precision	Recall	F1-Score	Support
B	0.45	0.27	0.34	357
M	0.26	0.43	0.33	212

The accuracy in this case was 33.22%, showing a modest improvement but still insufficient for reliable medical data clustering. The performance metrics demonstrate a slight improvement in precision and recall, but overall, the clustering performance remains inadequate (Table 4).

The Case of Four Clusters and a Threshold of 3

With four clusters and a threshold of 3, the accuracy dropped again to 24.08%. The confusion matrix for this case is presented in Table 5.

Table 5: Confusion Matrix for the Original BIRCH Algorithm (4 Clusters, Threshold = 3)

	Predicted B	Predicted M
Actual B	0	357
Actual M	75	137

Table 6: Performance Metrics for the Original BIRCH Algorithm (4 Clusters, Threshold = 3)

Diagnosis	Precision	Recall	F1-Score	Support
B	0.00	0.00	0.00	357
M	0.28	0.65	0.39	212

The poor precision and recall metrics, especially for benign cases, indicate that the original BIRCH algorithm struggles to effectively cluster medical data, even with different cluster settings (Table 6).

The Modified BIRCH Algorithm With Data Transformation

The modified BIRCH algorithm incorporated data transformation techniques and a scaled Euclidean distance measure to improve clustering performance. The experimental results across different numbers of clusters and threshold values demonstrated significant improvement in clustering accuracy.

The Case of Two Clusters and a Threshold of 3

After applying the modified BIRCH algorithm with data transformation, the confusion matrix for two clusters and a threshold of 3 is shown in Table 7.

Table 7: Confusion Matrix for the Modified BIRCH Algorithm (2 Clusters, Threshold = 3)

	Predicted B	Predicted M
Actual B	340	17
Actual M	23	189

The accuracy for this configuration reached 92.97%, a significant improvement over the original algorithm.

Table 8: Performance Metrics for the Modified BIRCH Algorithm (2 Clusters, Threshold = 3)

Diagnosis	Precision	Recall	F1-Score	Support
B	0.94	0.95	0.94	357
M	0.92	0.88	0.90	212

The precision, recall, and F1-score indicate strong performance in both benign (B) and malignant (M) cases, confirming that data transformation improves clustering quality (Table 8).

The Case of Three Clusters and a Threshold of 3

For the three-cluster case, the confusion matrix remains the same, and the accuracy stays at 92.97%, suggesting that the transformation method generalizes well across different numbers of clusters (Table 9).

Table 9: Confusion Matrix for the Modified BIRCH Algorithm (3 Clusters, Threshold = 3)

	Predicted B	Predicted M
Actual B	340	17
Actual M	23	189

The performance metrics in Table 10 are also consistent with the two-cluster case.

Table 10: Performance Metrics for the Modified BIRCH Algorithm (3 Clusters, Threshold = 3)

Diagnosis	Precision	Recall	F1-Score	Support
B	0.94	0.95	0.94	357
M	0.92	0.89	0.90	212

The Case of Four Clusters and a Threshold of 3

Even with four clusters, the modified BIRCH algorithm maintained its strong performance, achieving an accuracy of 92.97% (Table 11).

Table 11: Confusion Matrix for the Modified BIRCH Algorithm (4 Clusters, Threshold = 3)

	Predicted B	Predicted M
Actual B	340	17
Actual M	23	189

The corresponding performance metrics are shown in Table 12.

Table 12: Performance Metrics for the Modified BIRCH Algorithm (4 Clusters, Threshold = 3)

Diagnosis	Precision	Recall	F1-Score	Support
B	0.94	0.95	0.94	357
M	0.92	0.89	0.90	212

The Case of Two Clusters and a Threshold of 2

In this experiment, reducing the threshold value to 2 led to a notable improvement. Table 13 presents the confusion matrix for the modified BIRCH algorithm with two clusters and a threshold of 2.

Table 13: Confusion Matrix for the Modified BIRCH Algorithm (2 Clusters, Threshold = 2)

	Predicted B	Predicted M
Actual B	353	4
Actual M	5	207

The accuracy for this configuration reached an excellent 98.40%, underscoring the positive impact of the transformation process and parameter adjustments.

This final configuration demonstrates the highest level of accuracy, suggesting that the modified BIRCH algorithm is best suited for a two-cluster configuration with a threshold of 2. The experimental results highlight the significant improvement in clustering performance achieved by the modified BIRCH algorithm with data transformation.

Comparison between the Original and Modified BIRCH Algorithms with Transformed Data

Table 14: Performance Metrics for the Modified BIRCH Algorithm (2 Clusters, Threshold = 2)

Diagnosis	Precision	Recall	F1-Score	Support
B	0.99	0.99	0.99	357
M	0.98	0.98	0.98	212

In this section, the researcher conducts a comparative analysis between the original and modified BIRCH algorithms based on their clustering accuracy using transformed data. The results are summarized in Table 15.

Clustering Accuracy Comparison

The original BIRCH algorithm demonstrated a clustering accuracy of 33.22% when applied to the breast cancer dataset. In stark contrast, the modified BIRCH algorithm, which utilized data transformation, achieved an impressive accuracy of 98.40%. This significant improvement of over 65% in clustering accuracy underscores the efficacy of data transformation in enhancing the algorithm's performance.

Table 15: Comparison between the Original and Modified BIRCH Algorithms

Selected Algorithm	Clustering Accuracy
Original BIRCH	33.22%
Modified BIRCH	98.40%

Visual Representation of Clustering Outcomes

Fig. 1 illustrates the clustering results for tumor cases, where benign tumors are represented in blue and malignant tumors in red. It is evident that the dataset contains a higher number of benign cases compared to malignant ones.

When the original BIRCH algorithm was applied without any modifications, it became apparent that numerous tumor cases were misclassified. Specifically, several instances diagnosed as benign tumors were incorrectly clustered as malignant tumors, and vice versa. This misclassification is visually represented in Fig. 2.

Performance Metrics and Clustering Efficiency

Following the enhancements made to the BIRCH algorithm, the clustering performance markedly improved. A total of 121 nearest neighbors were computed, and all 569 samples were indexed in just 0.001 seconds. Neighbors for the entire dataset were calculated in 0.021 seconds, and conditional probabilities for all sample data points (n = 569) were also computed. The mean sigma value recorded was 33.68. The

DOI: 10.3844/jcssp.2025.2074.2080

Kullback-Leibler (KL) divergence was observed after 250 iterations with an early exaggeration of 49.38, and after 4,000 iterations, it showed an exaggeration of 0.210.

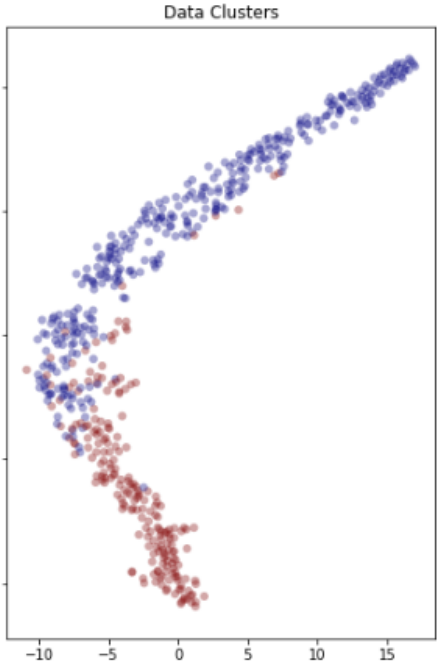

Fig. 1: Clustering of the Tumor Cases in the Study Data

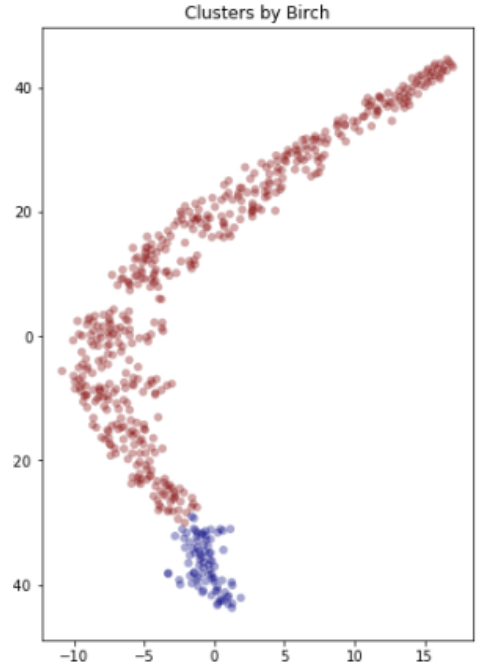

Fig. 2: Tumor Clusters Generated by the Original BIRCH Algorithm

The clustering outcomes of the data records are further illustrated in Fig. 3. The left side of this figure displays the results after transforming the data and applying the modified BIRCH algorithm, while the right side shows the clustering results before the modifications were made. This comparison highlights the superior clustering performance achieved with the modified BIRCH algorithm.

In this analysis, a total of 121 nearest neighbors were calculated again, and all sample records (n = 569) were indexed in 0.003 seconds. Furthermore, neighbors were computed for the entire sample in 0.040 seconds, with conditional probabilities also derived. The mean sigma value was significantly lower at 1.55. The KL divergence occurred after 250 iterations with an early exaggeration of 64.63, and after 1,400 iterations, the exaggeration was 0.811.

The comparative analysis reveals that the modified BIRCH algorithm significantly outperforms the original version, especially after data transformation. The substantial increase in clustering accuracy demonstrates the importance of algorithmic enhancements in achieving reliable results in medical data classification.

Fig. 3: Clustering Results Before and After Use of the Modified BIRCH Algorithm with Transformed Data

Conclusion

This study presents a transformation-driven clustering framework designed to enhance the classification accuracy of medical datasets, with a specific focus on breast cancer diagnosis. By integrating advanced data preprocessing techniques, including feature scaling and standardization, the proposed model successfully improves the clustering process and overall diagnostic performance. The experimental results demonstrate a dramatic improvement in accuracy—from 33.22% using a conventional approach to 98.40% with the enhanced methodology—highlighting the critical role of intelligent data transformation in unsupervised learning tasks.

The proposed approach proves particularly effective in hierarchical clustering scenarios where traditional algorithms often struggle with unbalanced feature scales and noisy inputs. Moreover, the robustness of the framework across multiple cluster configurations affirms its adaptability and practical viability in real-world medical data analysis. Beyond high clustering accuracy, the system exhibits computational efficiency, making it suitable for deployment in large-scale healthcare analytics and diagnostic support systems.

This work contributes to the growing body of research at the intersection of intelligent engineering and medical informatics, offering a scalable, interpretable, and high-performance solution for clinical data clustering. Future research can extend this framework by incorporating hybrid clustering-classification pipelines, integrating domain knowledge through semi-supervised learning, or applying the model to other critical areas such as genomic profiling or radiological imaging. Ultimately, this research lays the groundwork for more intelligent, responsive, and precise decision-support systems in healthcare environments.

Acknowledgment

This work was supported by Jadara University. I would like to thank the Deanship of Scientific Research at Jadara University for its support.

Funding Information

This research was funded by Jadara University.

Ethics

This study was conducted in accordance with the ethical standards set by the institution and the Declaration of Helsinki. The dataset used in this study is publicly available and does not contain any personally identifiable information. Ethical approval was not required as the dataset used is publicly accessible and has been anonymized.

Data Availability Statement

The data used in this study are publicly available and can be accessed from the UCI Machine Learning Repository.

Conflict of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

Al Eiadeh, M. R., & Al Batah, M. (2024). An improved binary Crow-JAYA optimization system with various evolution operators such as mutation for finding the max clique in dens graph. *International Journal of Computing Science and Mathematics*, *1*(1). https://doi.org/10.1504/ijcsm.2024.10063798

Al-Batah, M. (2014). Testing the Probability of Heart Disease Using Classification and Regression Tree Model. *Annual Research & Review in Biology*, *4*(11), 1713-1725. https://doi.org/10.9734/arrb/2014/7786

Al-Batah, M. S. (2019). Automatic diagnosis system for heart disorder using ECG peak recognition with ranked features selection. *International Journal of Circuits, Systems and Signal Processing*, *13*, 391-398.

Al-Batah, M. S. (2019). Ranked Features Selection with MSBRG Algorithm and Rules Classifiers for Cervical Cancer. *International Journal of Online and Biomedical Engineering (IJOE)*, *15*(12), 4-18. https://doi.org/10.3991/ijoe.v15i12.10803

Alkhasawneh, M. Sh., Tay, L. T., Ngah, U. K., Al-batah, M. S., & Mat Isa, N. A. (2014). Intelligent Landslide System Based on Discriminant Analysis and Cascade-Forward Back-Propagation Network. *Arabian Journal for Science and Engineering*, *39*(7), 5575-5584. https://doi.org/10.1007/s13369-014-1105-8

Garg, A., Gupta, A., & Sofat, S. (2006). P-BIRCH: A parallelized BIRCH clustering algorithm for distributed systems. *Journal of Parallel and Distributed Computing*, *66*, 347-360.

Kumar, R., & Shah, N. (2021). A hybrid model for anomaly detection in healthcare data. *IEEE Journal of Biomedical and Health Informatics*, *25*(10), 3312-3321. https://doi.org/10.1109/JBHI.2021.3051968

Lei, Y. (2016). E-BIRCH: An enhanced BIRCH algorithm for time series clustering with Dynamic Time Warping. *Pattern Recognition Letters*, *69*, 19-25. https://doi.org/10.1016/j.patrec.2015.10.011

Li, J., Sun, Y., & Yang, X. (2015). A hybrid BIRCH-DBSCAN clustering algorithm for large datasets with noise. *Pattern Recognition and Artificial Intelligence*, *28*(2), 104-115.

Li, X., & Jie, W. (2013). AS-BIRCH: An improved BIRCH clustering algorithm for arbitrary-shaped clusters. *Journal of Network and Computer Applications*, *36*(6), 1626-1631. https://doi.org/10.1016/j.jnca.2013.03.015

Lorbeer, B., Haase, M., & Kuster, U. (2017). A-BIRCH: Automatic BIRCH clustering using parallelized Gap Statistic. *IEEE Transactions on Parallel and Distributed Systems*, *28*(7), 1913-1922.

Luo, F., & Li, Z. (2021). Preprocessing and data transformation techniques for improving BIRCH clustering. *Expert Systems with Applications*, *173*, 114650. https://doi.org/10.1016/j.eswa.2021.114650

Nayak, B., & Mahapatra, S. (2017). MD-BIRCH: A multidimensional extension of BIRCH for distributed environments. *International Journal of Computer Applications*, *162*(6), 10-19. https://doi.org/10.5120/ijca2017913640

Ramachandran, M., & Govindan, S. (2022). Modified BIRCH algorithm for early breast cancer detection using mammogram images. *IEEE Access*, *10*, 3375-3384. https://doi.org/10.1109/ACCESS.2022.3147746

Zhang, H., Wang, Y., & Li, Z. (2020). Deep BIRCH: A deep learning-based hierarchical clustering algorithm for complex datasets. *Journal of Machine Learning Research*, *21*(4), 1-20. https://doi.org/10.1145/3413896