UNIVERSITAS UTPADAKA SWASTIKA
Jl. KS Tubun No.11 Tangerang 15112 Banten, Telp. (021) 5589161-62 Fax. (021) 5589163
JISI Jurnal Ilmiah Sistem Informasi
Vol. 3 No. 01 (2025)
ISSN: 3046-711X

A Comparative Review of Clustering and Classification Algorithms for Big Data Analytics

Lukas Umbu Zogara[1], *Leny Tritanto Ningrum[2]

[1]Utpadaka Swastika University, Tangerang, Indonesia 15112
[2]Universitas Binaniaga Indonesia, Bogor, Indonesia, 16153
[1]lukasumbuzogara68@gmail.com, [2]lenytrinie@unbin.ac.id
*corresponding author: lenytrinie@unbin.ac.id

Accepted: May 5, 2024 | Published: May 8, 2025

ABSTRACT

These days, there's so much data being created all the time. It's honestly getting hard to keep up. That's where data mining comes in. Basically, people use it to make sense of all this huge amount of information, and there are two main ways to do it: clustering and classification. I found that there are a bunch of algorithms for both, like K-Means, DBSCAN, and Hierarchical Clustering for clustering, and then there's Decision Tree, Naïve Bayes, SVM, and Random Forest for classification. Each of these has its own strengths and weaknesses depending on the data you're working with. The point of this paper was really to see how these algorithms perform and to give people an idea of which one might work best depending on the situation. What we found is that no algorithm is perfect for everything. So, choosing the right one really comes down to understanding the data and figuring out what you're trying to get out of it.

KEYWORDS: Big Data, Data Mining, Clustering, Classification, Machine Learning

1. Introduction

In this day and age, we can't escape data. It's coming from everywhere, right? Social media posts, transactions, devices we carry around, sensors on machines all of it. The volume of data is growing so fast that managing it is starting to feel like a challenge.

This is where Big Data comes in. It helps us understand the sheer scale of it all. Big Data is about a few things: the amount of data, how quickly it flows, how different it is, whether it's trustworthy, and what we can actually use it for.

But just collecting data doesn't do much on its own. You need to make sense of it, and that's where tools like data mining come in. Data mining helps us dig through massive datasets, uncovering patterns using a mix of statistics and machine learning. There are two key techniques in data mining: clustering and classification.

Clustering is all about grouping data points that are similar. The catch is, we don't always know how those groups should be defined upfront. This is where algorithms like K-Means, DBSCAN, and Hierarchical Clustering come into play. They work great in many situations, but they aren't perfect. For instance, K-Means can be finicky about where it starts, Hierarchical Clustering doesn't handle large datasets well, and DBSCAN doesn't work well when the data is too complex [1], [5], [6].

Classification is a little different. It's used when we already know the labels of data, and we want to predict the labels of new data. Methods like Decision Trees, Naïve Bayes, SVM, and Random Forest are often used for classification tasks. They're useful for things like risk prediction or pattern recognition, but each comes with its own set of weaknesses. Some might overfit, others might make assumptions that don't hold true, and some can be hard to interpret [2], [3], [9], [10].

A lot of times, combining techniques can give better results. For example, some studies have found that combining K-Means with Hierarchical Clustering yields better results than using either one by itself. In this paper, we'll dive into how these methods work, what

their strengths and weaknesses are, and when you might want to use them based on the kind of data you're dealing with.

2. Review of Literature

The rapid evolution of data complexity and volume has prompted significant research in clustering and classification algorithms tailored for Big Data environments. Traditional algorithms such as K-Means and Hierarchical Clustering have been widely studied, but face challenges in scalability and adaptability when applied to large-scale datasets [1], [6].

Recent developments have introduced advanced methods such as deep clustering, which integrates representation learning and clustering in a unified framework. Ren et al. [11] provide a comprehensive survey on deep clustering approaches, categorizing them into unsupervised, semi-supervised, and multi-view types. Similarly, Zhou et al. [12] examine the landscape of deep clustering and emphasize the potential of hybrid models to improve clustering outcomes in high-dimensional spaces.

In addition, the use of Big Data platforms like Apache Spark has been explored to enhance the scalability of clustering algorithms. Saeed et al. [13] reviewed Spark-based clustering techniques and highlighted their effectiveness in reducing computational time while maintaining accuracy.

In the domain of classification, deep learning has emerged as a dominant paradigm. Minaee et al. [14] reviewed over 150 deep learning models for text classification, demonstrating their superiority over traditional methods in tasks such as sentiment analysis and intent classification. However, the complexity and resource requirements of deep learning models remain a barrier in many practical scenarios.

Hybrid classification systems that combine traditional machine learning models with modern deep learning frameworks are gaining popularity. Banait et al. [15] discuss the significance of hybrid clustering-classification models in enhancing accuracy for large datasets. These findings align with earlier studies that suggest no single algorithm performs best across all tasks and that ensemble or hybrid techniques can provide balanced solutions [1], [10].

This study employs a qualitative descriptive research method, focusing on analyzing and comparing clustering and classification algorithms based on secondary data from existing literature. The goal is to systematically evaluate each algorithm's behavior, strengths, and limitations by reviewing findings from scholarly sources, empirical case studies, and technical documentation referenced in [1]–[15]

3. Methodology

This research adopts a descriptive qualitative approach based on a systematic literature review. Information was gathered from peer-reviewed journals, technical documentation, and reputable online academic sources. The study focuses on eight algorithms grouped into two categories:

- ✓ Clustering Algorithms: K-Means, Hierarchical Clustering, DBSCAN
- ✓ Classification Algorithms: Decision Tree, Naïve Bayes, Support Vector Machines (SVM), Random Forest

Each algorithm was analyzed based on factors including computational efficiency, scalability, interpretability, resilience to noise, and adaptability to Big Data environments. The evaluation draws upon previous studies and expert documentation to ensure validity and objectivity [1]–[10].

4. Results and Discussion

The performance comparison of clustering and classification algorithms reveals varying strengths and weaknesses based on five critical evaluation parameters: computational efficiency, scalability, interpretability, robustness to noise/outliers, and suitability for

high-dimensional data. The following analysis presents detailed findings, supported by existing literature [1]–[15].

4.1 Clustering Algorithms

4.1.1 K-Means

K-Means is widely praised for its simplicity and computational efficiency, especially with large datasets. Its time complexity is generally O(nkt), where n is the number of data points, k the number of clusters, and t the number of iterations [6]. However, K-Means is sensitive to the initial placement of centroids and struggles with clusters of irregular shapes or varying densities [5]. Recent studies have attempted to address these limitations using initialization strategies like K-Means or integrating with Spark for scalability [13].

4.1.2 Hierarchical Clustering

Hierarchical clustering provides a tree-based (dendrogram) representation of data clusters, making it highly interpretable. It does not require the number of clusters to be specified beforehand. However, the method is computationally intensive (typically $O(n^2 \log n)$) and not scalable for large datasets [1], [7]. Despite its poor scalability, it remains useful for small to medium datasets in domains such as bioinformatics and social network analysis.

4.1.3 DBSCAN

DBSCAN is effective in identifying clusters of arbitrary shapes and is robust to noise and outliers [5], [8]. Unlike K-Means, DBSCAN does not require the number of clusters as an input. Its main drawback is poor performance with high-dimensional data and sensitivity to parameters like ε (radius) and MinPts (minimum points) [12]. New approaches, such as adaptive DBSCAN or integrating it with Spark, have shown promise for improving its scalability and performance [13].

4.2 Classification Algoritms

4.2.1 Decision Tree

Decision Trees are highly interpretable and capable of handling both categorical and numerical data [9]. Their decision rules are human-readable, which is a major advantage in domains requiring explainability (e.g., healthcare, law). However, they tend to overfit, especially with noisy data. Recent advancements include pruning techniques and ensemble methods such as Gradient Boosted Trees to improve accuracy and generalization [10].

4.2.2 Naive Bayes

Naïve Bayes is one of the fastest classifiers in terms of training and prediction, with linear time complexity. It works particularly well on text data and high-dimensional problems, provided the assumption of feature independence is reasonable [2]. Its simplicity makes it suitable for real-time systems, but in scenarios with correlated features, its performance can degrade [14].

4.2.3 Support Vector Machine (SVM)

SVM offers excellent performance for binary classification with a clear margin separation. It is robust to overfitting and handles high-dimensional data well, especially in text classification and bioinformatics [2], [14]. However, SVM can be computationally expensive for large datasets, and adapting it to multi-class problems typically requires techniques like one-vs-one or one-vs-all decomposition.

4.2.4 Random Forest

Random Forest combines multiple decision trees to improve prediction accuracy and reduce overfitting [10]. It is robust to noise and effective for large datasets. Although less interpretable than a single

Decision Tree, Random Forest provides feature importance measures that can guide variable selection [10], [15]. The model performs well across a variety of data types and is often used as a baseline in data science competitions.

4.2.5 Comparative Summary

Table 1. Comparison of Clustering and Classification Algorithms

Algorithm	Efficiency	Scalability	Interpretability	Noise Robustness	High-Dimensional Suitability
K-Means	High	High (with Spark)	Moderate	Low	Low
Hierarchical	Low	Low	High	Moderate	Moderate
DBSCAN	Moderate	Moderate (with tuning)	Moderate	High	Low
Decision Tree	High	High	High	Low	Moderate
Naïve Bayes	Very High	High	High	Low	High
SVM	Moderate	Moderate	Low	Moderate	High
Random Forest	Moderate	High	Moderate	High	High

4.2.6 Synthesis of Findings

The results demonstrate that no single algorithm performs optimally across all dimensions. For instance, while Naïve Bayes is fast and scales well, it assumes feature independence, which limits its accuracy in complex datasets. SVM and Random Forest are strong performers in high-dimensional settings, while DBSCAN excels in identifying non-linear patterns and noise resilience.

In clustering, the choice often depends on the shape and density of the data. K-Means works well when clusters are spherical and balanced, whereas DBSCAN handles irregular clusters and noise more effectively. For classification, Random Forest is a solid general-purpose choice, while Decision Trees offer a balance between simplicity and accuracy when properly pruned.

Recent literature suggests the adoption of hybrid or ensemble models to mitigate individual weaknesses [12], [15]. Combining DBSCAN with K-Means or integrating Naïve Bayes with deep learning components are emerging strategies that have shown improvement in both academic and real-world applications.

5. Conclusion and Suggestions

Conclution

This paper conducted a comparative analysis of clustering and classification algorithms widely used in data mining and Big Data analytics. Through a review of the literature and synthesis of theoretical and empirical insights, seven algorithms were evaluated K-Means, Hierarchical Clustering, DBSCAN (clustering); Decision Tree, Naïve Bayes, SVM, and Random Forest (classification) based on five core performance metrics: computational efficiency, scalability, interpretability, robustness to noise, and ability to handle high-dimensional data.

The results highlight that:

- ✓ K-Means is computationally efficient but performs poorly with noisy data and irregular clusters.
- ✓ Hierarchical Clustering offers high interpretability but lacks scalability.
- ✓ DBSCAN is robust to noise and adaptable to arbitrary shapes but underperforms in high-dimensional spaces.
- ✓ Decision Trees are easy to interpret but vulnerable to overfitting.
- ✓ Naïve Bayes is ideal for high-dimensional data and fast execution but assumes feature independence.
- ✓ SVM excels in handling complex data structures but is computationally intensive for large-scale problems.
- ✓ Random Forest delivers strong generalization and noise resistance but at the cost of reduced transparency.

Based on these insights, no single algorithm can be universally recommended for all scenarios. The optimal algorithm must be selected based on dataset characteristics, performance priorities, and available computational resources. This aligns with recent trends favoring hybrid and ensemble approaches, which combine the strengths of multiple models for improved accuracy and adaptability [12], [15].

Suggestions

Based on the conclusions and the literature reviewed, the following suggestions are offered:

- ✓ Algorithm Selection Should Be Data-Driven.
  Practitioners should carefully examine the structure, size, and nature of their datasets before choosing a specific algorithm. For example, DBSCAN is more suitable for spatial or noisy data, while Naïve Bayes performs well on text or document classification tasks.
- ✓ Adopt Hybrid and Ensemble Techniques
  Future implementations should consider integrating traditional algorithms with modern methods such as deep learning or ensemble learning (e.g., boosting, bagging). These techniques help overcome the limitations of single-algorithm approaches and enhance performance in complex tasks.
- ✓ Utilize Big Data Frameworks
  For large-scale data processing, algorithms should be implemented using distributed computing platforms such as Apache Spark or Hadoop to improve execution time and scalability [13].
- ✓ Emphasize Interpretability in Critical Domains
  In domains like healthcare, finance, or law, where decisions must be explainable, interpretable models such as Decision Trees or interpretable variants of ensemble methods (e.g., Explainable Boosting Machines) are preferable.
- ✓ Encourage Empirical Testing
  Future research should perform experimental evaluations on real-world datasets across different industries to validate theoretical claims and measure actual performance trade-offs under practical constraints.

References

[1] T. Alfina and B. Santosa, "Analisa Perbandingan Metode Hierarchical Clustering, K-Means dan Gabungan Keduanya dalam Membentuk Cluster Data," J. Teknik Industri ITS, vol. 1, no. 1, pp. 1–5, 2012.

[2] C. L. Koo et al., "A review for detecting gene-gene interactions using machine learning methods in genetic epidemiology," BioMed Research International, vol. 2013, Art. no. 432375, 2013.

[3] A. Nordman, "Data Mining Classification Rules - Rule-based Classifier," [Online]. Available: http://staffwww.itn.liu.se/~aidvi/courses/06/dm/lectures/lec4.pdf

[4] "Case-Based Reasoning," [Online]. Available: https://students.warsidi.com/2017/06/pengertian-dan-cara-kerja-case-based-reasoning.html

[5] "Density-Based Clustering Algorithm," [Online]. Available: https://sites.google.com/site/dataclusteringalgorithms/density-based-clustering-algorithm

[6] Google Developers, "Clustering Algorithms: Advantages and Disadvantages," [Online]. Available: https://developers.google.com/machine

-learning/clustering/algorithm/advantages-disadvantages

[7] Sigma Magic, "Hierarchical Clustering," [Online]. Available: https://www.sigmamagic.com/blogs/hierarchical-clustering/

[8] Wikipedia, "DBSCAN," [Online]. Available: https://en.wikipedia.org/wiki/DBSCAN

[9] GeeksForGeeks, "Decision Tree," [Online]. Available: https://www.geeksforgeeks.org/decision-tree/

[10] HolyPython, "Random Forest: Pros and Cons," [Online]. Available: https://holypython.com/rf/random-forest-pros-cons/

[11] X. Ren, Y. Wu, and J. Zhang, "Deep clustering: A comprehensive review," arXiv preprint arXiv:2210.04142, 2022.

[12] J. Zhou, Y. Xu, and X. Liu, "A survey on deep clustering: Methods and challenges," arXiv preprint arXiv:2206.07579, 2022.

[13] F. Saeed, K. Benkhelifa, and A. Hossain, "Clustering techniques in Apache Spark: A survey," Applied Sciences, vol. 10, no. 7, pp. 1-25, 2020.

[14] S. Minaee et al., "Deep learning based text classification: A comprehensive review," arXiv preprint arXiv:2004.03705, 2020.

[15] M. Banait, S. Mehta, and R. Jain, "A review on efficient clustering methods for big data," International Journal of Computer Trends and Technology (IJCTT), vol. 71, no. 2, pp. 45-50, 2023.