Bioengineering

https://www.aimspress.com/journal/Bioengineering

AIMS Bioengineering, 13(2): 209–238.
DOI: 10.3934/bioeng.2026009
Received: 26 February 2026
Revised: 12 April 2026
Accepted: 27 April 2026
Published: 30 April 2026

Research article

Graph-based feature extraction for drug classification using machine learning

Sava Mohammed Hamazyad[1], **Sadegh Sulaimany**[2] **and Sarbaz H.A.Khoshnaw**[1,*]

[1] Department of Mathematics, University of Raparin, Ranya, Sulaimani, Iraq
[2] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

* **Correspondence:** Email: sarbaz.hamza@uor.edu.krd; Tel: +9647503161780.

Abstract: Drug classification is a key task in medical decision-making, and this process can be supported through appropriate selection of drugs that fit patient attributes and medical history. Currently, state-of-the-art solutions for this problem tend to make use of attribute-based representation and may overlook the relational structure that could exist in the data points. This paper seeks to examine a machine learning algorithm for multiclass drug classification, where the target variable is the prescribed type of drug, using a graph-based feature extraction method. The proposed method can be used thorough data preprocessing strategies, including handling imbalance and removal of outliers. Evaluation of the method requires a 10-fold cross-validation strategy to achieve a genuine and impartial evaluation. In each training set, a similarity graph is created and graph-based attributes such as degree centrality measures and clustering coefficient are extracted by using three different distance measures to consider graph relationships between samples. In the case of the testing data, graph attributes are computed by supplanting a weighted k-nearest neighbor strategy, wherein there is absolute prevention of information leakage. The graph attributes obtained along with other attributes are used to train different machine learning classifiers. The results show that the performance across all distance measures plays a great role for such classifications.

Keywords: drug classification; machine learning; graph based feature; distance measures; multiclass classification

1. Introduction

Drug classification is an important area in the healthcare and drug development industry, as it assists in clinical decision-making, identification of pharmacological uses of a drug compound, and estimation mechanisms related to drug-drug interaction [1]. Accurate drug classification plays an important role in ensuring improved patient outcomes as well as speeding up the development of medication. By being able to accurately identify classes of pharmaceuticals and predict their interactions with respect to those classes of substances or drugs, clinicians enable them to move closer to identifying therapeutic uses that would be appropriate for individual molecular profiles that will improve treatment outcomes [2]. Through a more refined and systematic categorization of drugs based on their manner of action and therapeutic attributes, a more adept capacity to predict drug behavior has been enabled through machine learning models, thus ensuring that more targeted and effective medications are developed and created [3]. Furthermore, a more refined and maximized pace in terms of streamlining and drug development has been enabled through recent advances in deep learning models and its allied capabilities, thus reducing costs and time in terms of the development and creation of various aspects of pharmaceutical products and medications [4].

Since they can model complex, non-linear relationships inherent in biomedical data, machine learning approaches have become one of the most frequently used methods for medical classification [5]. Classical algorithms such as support vector machines, decision trees, and logistic regression methods, among others, have shown encouraging results in classifying drugs into therapeutic classes or prescription outcomes [6, 7], and further improvements in accuracy have been achieved by ensemble methods and deep models in light of richer feature representations [4, 8]. Nevertheless, more current models still largely based on attribute or property features, and treating each patient or drug instance as an independent observation, also do not fully capture the complex relationships inherent in biomedical data [9]. This might negatively affect model generalization capabilities, especially when working with a more heterogeneous dataset. Hence, these relations between samples should be incorporated to develop more informative models.

Recent advances in the field of computation have led to the introduction of graph-based feature extraction methods to address these limitations. Graph-based feature extraction has emerged as a powerful approach to learning from relational data by representing samples as nodes and their connection as edges [10]. Specifically in the biomedical literature, various graph models have been employed to describe graphical representations for molecule graphs, protein-protein interaction graphs, disease networks, and similarity graphs of patients [11, 12]. Through the use of graph topology, it is possible to incorporate both local neighborhood patterns and global topological properties into the graph that cannot be extracted through traditional feature models [13]. Recently, research works have clearly demonstrated that adding graph-based features into machine learning models can increase predictive accuracy in different biomedical and network domains [14–17]. Measures such as centrality analysis, clustering coefficient, and connectivity information are complementary in nature and therefore provide information on the structural characterization of instances in the dataset [18].

Although traditional machine learning methods for drug classification have achieved great success, the existing methods mainly focus on the samples independently and only use attribute-based information. However, biomedical samples have inherent relationship information between samples with similar clinical characteristics. This relationship information may affect the performance of the model if it is not taken into consideration. Thus, there is a need for approaches that can effectively incorporate the hidden relationship between the samples during the learning process.

To address this limitation, this paper presents a machine learning framework for multiclass drug classification, in which graph-based feature extraction takes place within a robust setting for evaluation. By capturing structural relationships through graph representations, the proposed method aims to improve the performance of the classification and provide a more informative representation. The method presented here characterizes the combination of data preprocessing with similarity-based graph construction and graph-theoretic feature extraction. We built a system integrated into the classification process to enable the model to learn latent dependencies among samples. Unbiased performance estimation is secured thanks to a cross-validation strategy, while relational features of unseen samples can be generated with a graph-based propagation mechanism, hence without data leakage. We demonstrate herein how the incorporation of relational modeling can lead to the reinforcement of drug classification tasks and provide interpretable insights into the interactions between patients and drugs in biomedical datasets.

In order to ensure a fair evaluation of the proposed graph learning framework, three popular distance metrics such as the Gower, Cosine, and Euclidean distances are used while creating the graph [19, 20]. These metrics are considered under identical preprocessing setups and experimental settings, enabling a consistent assessment of the framework across different similarity definitions. The results of classifications based on each of these distances are presented and discussed.

There are multiple main contributions of this study. First, we propose a novel graph-based feature extraction for multiclass drug classification tasks with hidden relationships between samples. It constructs similarity networks using multiple distance measures and presents how different similarity measures affect graph construction and classification performance. In addition, we extract graph-based centrality features to enhance the representation of tabular medical data. Then, we introduce a new graph-based feature propagation method to generate relational features for test sets without causing leakage. Furthermore, it shows the effectiveness of the proposed method by demonstrating the improvement in classification performance by integrating graph-based features with traditional machine learning approaches, and balancing techniques are also used. Finally, a comprehensive evaluation using multiple performance metrics and cross-validation is used.

The structure of this paper is as follows: Section 2 will discuss current literature on drug classification and learning with graphs in the biomedical field. Section 3 will describe the methodologies employed in this study, which include data preprocessing, graph building, feature extraction, and classification algorithms used in the study for findings and analysis. Section 4 will describe the findings and analysis, and Section 5 will conclude the study and address the limitations and potential avenues for further study.

2. Literature review

Machine learning models have been applied extensively in the field of biomedical and pharmaceutical research because the models are capable of modeling complex, high-dimensional problems and recognizing complex patterns from enormous biological and chemical datasets. The main applications in drug discovery and development are in target identification, compound screening, drug repurposing, toxicity prediction, and clinical trial optimization using machine learning [21]. In the field of drug classification and other medical decision-support systems, the initial work was dominantly carried out in pattern recognition or traditional machine learning, where handcrafted features were extracted from medical or molecular datasets [22]. These methods typically adopt molecular descriptors and structured clinical attributes as input features for model prediction. Previous studies showed that traditional machine learning algorithms enjoyed widespread applications in the prediction tasks related to drugs by showing promising results with feature-based representations only [23–25]. However, these works had limited success and made the assumption of independence between data samples and their inability to fully capture complex relationships within biomedical data.

Ensemble learning methods, particularly ensemble tree-based approaches such as random forests, gradient boosting, and various variants of boosted decision trees, are strong classification techniques that combine multiple learners to achieve improved accuracy in classification results [26]. These approaches were also successfully employed in subsequent studies in various biomedical and pharmaceutical predictive tasks due to their ability to handle high-dimensional and heterogeneous data, reduce model variance, and improve robustness. For instance, random forest models have been successfully applied in drug-target interaction predictions using molecular and similarity feature groups, achieving high prediction accuracy [27], while under-sampling methods focusing on ensemble methods, such as random forests, have emerged as effective methods for imbalanced datasets [28]. Furthermore, a gradient boosting decision tree ensemble-based method proved effective in identifying drug–target associations from network-derived features [29]. In addition, extreme gradient boosting (XGBoost) proved effective in predicting the chemical compound bioactivity compared to traditional ML classifiers [30]. These findings highlight the suitability of ensemble and tree-based techniques for intricate drug discovery and prediction problems.

Despite these encouraging results, the majority of existing approaches rely primarily on attribute-based representations of drugs or patients, without explicitly modeling the relational structure among samples. To address this limitation, graph-based learning approaches have gained increasing attention in biomedical data analysis. In graph representations, samples are modeled as nodes, and edges represent similarity or distance-based relationships between them [10]. This formulation enables the capture of both local and global structural information, which is often lost in traditional vector-based representations. Graph-based methods have demonstrated promising results in various applications, including patient similarity analysis, disease classification, and drug discovery [31].

Beyond drug informatics, there have been many applications of graph-based feature extraction in various real-world problems, such as biomedical diagnostics, industrial condition monitoring, cybersecurity, education, and so on. For instance, Wang et al. [32] proposed a graph-theory-based framework in which brain functional connectivity networks were constructed and node-level degree centrality features were extracted and combined with classical machine learning classifiers to predict response to treatment with antiepileptic drugs in children with epilepsy (achieved 84.22%). Additionally, Hosseini et al. [33] introduced a graph-based feature engineering approach for the diagnosis of rolling element bearing fault, where implicit relationships among vibration signal entities were modeled as a graph and structural graph features were used with traditional machine learning models (achieved 100% AUC). In cybersecurity, a malware detection approach was proposed by Mafakheri and Sulaimany in 2024 [17], using a combination of complex network analysis and traditional application attributes (achieved 99% and 98%). In the field of biomedical and healthcare signal processing, the method has been employed to classify cough signals [34] and improve the performance of heart disease classification [35]. In the educational problem, graph-based topological feature extraction with machine learning and graph convolutional networks has been employed to predict at-risk students [36].

Previous studies on the same drug classification dataset that were used in this study were mainly based on traditional machine learning pipelines integrated with preprocessing and interpretability techniques. Initially, in 2021, D. V. Gala [37] using decision tree, random forest, and logistic regression models, the interpretability of the models was done using the LIME and SHAP methods. Their results showed that random forest and decision tree performed better, achieving an accuracy of 97.5%, and the feature importance analysis showed that the Na-to-K ratio and blood pressure were the most influential features for drug classification. Another work T. A. Vu [38] in 2023 focused on improving model performance by integrating data binning with the SMOTE oversampling technique to handle class imbalance. Various classifiers, such as k-nearest neighbors, logistic regression, stochastic gradient descent, naïve Bayes and gradient boosting, were evaluated. Their results demonstrated high accuracy, with logistic regression having 97.2% accuracy and 96.6% F1-score, also gradient boosting achieved an accuracy of 96.9% and an Fl-score of 96.8%. P. Purwono et al. in 2021 [39] used ensemble learning techniques, such as extra trees, XGBoost, gradient boosting, LightGBM, and CatBoost, along with explainable AI approaches, such as SHAP and permutation feature importance. The results demonstrated a near-perfect and perfect classification accuracy, with the extra tree algorithm having 98% accuracy, and another model having 100% accuracy, also identifying age, sex, and BP as the most significant features for prediction.

Although there is an increase in the use of machine learning methods in drug classification, most studies have used tabular features directly and treated samples independently. However, such classification models overlook the hidden relational structure that might exist between samples among tabular datasets. Consequently, a significant research gap exists for the improvement of multiclass drug classification through incorporating hidden features derived from graph-based representations of tabular data. This motivated us to propose a graph-based feature extraction method that captures the relationships between data samples and combines the relational features into the classification process, with the objective of improving the overall accuracy of drug classification models.

3. Data and methodology

Each sample in the biomedical data set contains a set of characteristics and descriptive attributes that collectively define a patient situation and environment. In traditional machine learning classifications, these attributes are used for classification purposes in models that assume independence in the samples, thus failing to observe any connections and similarities between patients. Such inter-sample relationships are particularly significant in biological data, wherein patients with similar attributes or responses tend to display correlated behavior that is not directly referenced in the original feature space. Ignoring these implicit relations may thus limit the expressiveness of feature representations and generalization in classification tasks. To overcome this, the proposed methodology introduces the relational element through the embedding of the patient in the form of a graph.

More specifically, pairwise correlations between samples are calculated in order to assess how strong the relationships are, and these correlations are employed to construct a similarity graph, with samples as graph nodes and their relationships encoded in edges, indicate their strength of the relationship. This relational representation has the ability to obtain graph-based local features that describe each sample not only by its own features, but also by its location, role, and importance within the graph. These features are then used subsequently in machine learning classifiers to enrich the learning process.

The overall framework is broken down into a series of stages.

- Preprocessing and Preparation of Data
- Building the Patient Similarity Network
- Graph-Based Feature Extraction Strategy
- Model Training and Evaluation

The complete framework of this study is presented in Figure 1, which describes the embedding of graph-based methods into the process of classification.

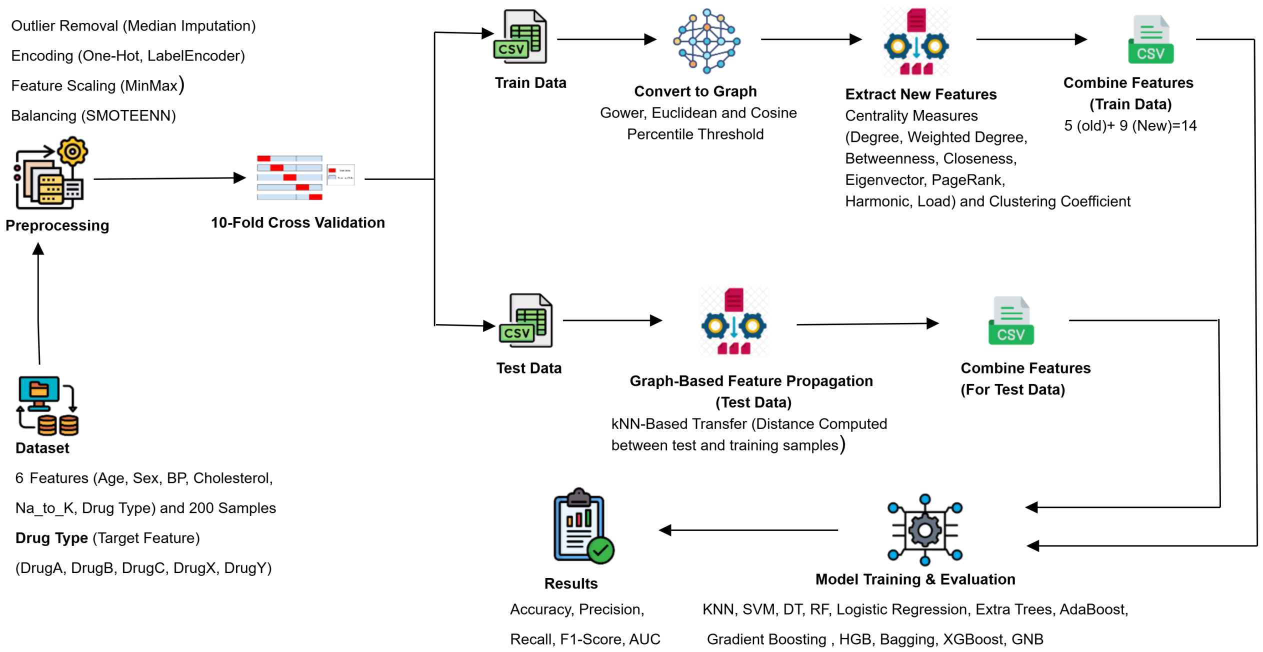


Figure 1. Overall workflow of the proposed research process.

3.1. Data and preprocessing

The dataset used in this work was obtained from the Kaggle repository [40]. In summary, the dataset consists of 200 samples and 6 features; these features include demographic information such as age and sex, clinical indicators blood pressure and cholesterol (denoted BP and Chol), a biochemical measurement sodium to potassium rate (Na_to_K), and the target feature drug representing five different types of drugs. The dataset is mixed; some features are numerical, while others are categorical.

Preprocessing is therefore necessary before training different models. A preliminary verification of the data ensured that no missing values were found in any of the features. Then, outlier detection was been carried out for numerical features by using the IQR approach [41]. This analysis identified 8 outliers in the Na_to_K feature, and in order to avoid the influence of the outliers and not lose any samples, a median-based imputation strategy was applied by substituting the outliers with the median value of Na_to_K (13.9365).

3.1.1. Encoding of categorical variables and feature scaling

The categorical features can now be represented numerically to help the different machine learning models that require numerical input. Therefore, one-hot encoding was applied to the categorical features (Sex, BP, Chol). For the target feature drug, the LabelEncoder is used, mapping each class to a unique integer label. This approach maintains the class distinction without increasing the dimension of the target variable. In addition, numerical features (Age, Na_to_K) were scaled using MinMax normalization that maps the values in the range between 0 and 1. This step ensures that all numerical features contribute proportionally during model training and the features that have larger numerical ranges do not dominate the learning process.

3.1.2. Handling class imbalance using SMOTEENN

Examination of the class distribution indicated that there was a problem of imbalance with regard to the classes of drugs, which can affect the performance of the classification task. To correct this problem, a method that integrates the synthetic minority over-sampling technique (SMOTE) and edited nearest neighbors (ENN) was used [42, 43].

The SMOTE technique creates additional instances for minority classes by forming connections with neighboring instances, hence augmenting the class. The ENN technique then removes some dubious instances, particularly noise, using nearest-neighbor analysis, hence refining the class boundaries [44, 45]. The mix of these two techniques makes the data set cleaner and more balanced, hence a smoother generalized classifier for all classes of drugs.

3.2. Cross-validation

With the intention of conducting a precise and unbiased analysis of the proposed model, the 10-fold CV technique was used in this paper. In this case, the dataset was divided into ten roughly equal subsets and each subset was used for training purposes. The steps allowed in each iteration are described below:

1. Data splitting for training and test set: In the collection, one subset is used as a test set, while combining its remaining nine subsets forms the training set.
2. Graph construction and feature extraction (for training data): A similarity graph is constructed over the training samples, and for each node, graph based features are extracted. Then, by combination of the original features and new extracted features, an enriched training dataset was generated.
3. Feature propagation (for test data): Features for every test samples are generated through a weighted interpolation method from the training set in such a way that no data leakage occurs.
4. Training and Evaluation of the models: Train the machine learning models on the enriched training dataset. Then, performance metrics such as accuracy, recall, precision, AUC, and F1-score are recorded for this fold.

Finally, the average of the metrics over all ten folds provides the final performance. In this way, this cross-validation setting ensures that the impact of graph-based features is assessed reliably and reported, with results reflecting the model's true generalization ability on unseen data.

3.3. Graph conversion on training data

In every iteration of the cross-validation process, the graph conversion is performed only on the training data. This helps to strictly divide the samples of data into training and testing samples, and this process converts the attribute representation of the data into a graph structure that shows the similarities between patients. The graph conversion process mainly has two steps, and they are as follows.

3.3.1. Similarity computation

In order to quantify the similarity among the training samples, the pairwise correlations of these samples are calculated using three distinct distance metrics: *Gower, Euclidean, and Cosine* measures. Initially, the main advantage of the Gower distance is its suitability in handling various types of variables, both numeric and categorical within a unified formulation. Suppose that y_i and y_j are two samples, each of which has p features. For each k^{th} feature, the per-feature similarity denoted by s_{ijk} is defined as follows:

- **Numerical features:**
$$s_{ijk} = 1 - \frac{|y_{ik} - y_{jk}|}{\text{range}_k}, \tag{3.1}$$
where range_k refers to the difference between maximum and minimum values of feature k.
- **Categorical features:**
$$s_{ijk} = \begin{cases} 1, & \text{if } y_{ik} = y_{jk}, \\ 0, & \text{if } y_{ik} \neq y_{jk}. \end{cases} \tag{3.2}$$
An exact match between categories contributes maximum similarity, while differing categories contribute none.

The overall similarity between samples i and j is calculated as the weighted average of per-feature similarities:

$$S_{ij} = \frac{\sum_{k=1}^{p} w_{ijk} s_{ijk}}{\sum_{k=1}^{p} w_{ijk}}, \tag{3.3}$$

where w_{ijk} is a weight indicating whether the comparison for feature k is valid (e.g., not missing). Finally, the Gower distance d_{ij} is obtained by converting similarity to a distance measure:

$$d_{ij} = 1 - S_{ij}. \tag{3.4}$$

The Gower distance ranges from 0, indicating identical samples, to 1, indicating maximally different samples.

Subsequently, Euclidean distance calculates the distance of a straight line connecting points in a feature space for samples and its commonly used with numerical features. For two samples $\mathbf{x}_i$ =

$(x_{i1}, x_{i2}, \ldots, x_{ip})$ and $\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jp})$, the Euclidean distance is defined as

$$d_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2}. \tag{3.5}$$

This captures absolute differences for all variables and treats each dimension equally.

Finally, the Cosine distance considers the difference between samples not in terms of the angle between the vectors, but in relation to their magnitudes. However, it is applicable when one considers the relative positioning of features instead of the absolute values. Among two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, the formula for cosine distance is written as

$$d_C(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}, \tag{3.6}$$

where $\mathbf{x}_i \cdot \mathbf{x}_j$ is the dot product of the two vectors, and $\|\mathbf{x}_i\|$ is the Euclidean norm.

3.3.2. Graph construction using percentile threshold

After computing the pairwise distances among all training samples, a similarity graph is constructed. In this graph, each node represents a training sample and the edges encode strong similarity relationships between samples.

To determine which sample pairs should be connected, a percentile-based thresholding strategy is applied to the distance matrix. Specifically, an edge is generated between the nodes if the distance is less than the X^{th}*percentile* of the total distance distribution, this condition ensures that the only most similar samples are connected.

3.4. Graph-based feature extraction

The next step is extracting a set of graph-theoretic features after creating the similarity graph from the training data in the previous step, this technique is useful for characterizing the structural role of each node in the graph. Through this approach, the features are able to capture and provide both local and global properties of the graph, which give complementary information beyond the original attribute-based representation [46], which was the goal of our work. The following features are calculated for every node.

Let $G = (V, E)$ denote the undirected graph, where V is the set of nodes representing samples, and E is the set of edges representing similarity relationships. For a given node $v \in V$, $N(v)$ denotes the set of its neighboring nodes, w_{vu} is the weight of the edge between nodes v and u, and $d(v, u)$ represents the length of the shortest path between nodes v and u.

1. **Degree Centrality**: Degree centrality measures the immediate connectivity of a node. For a given node v, it is defined as the total number of edges connected to that node:

$$C_D(v) = \deg(v). \tag{3.7}$$

2. **Weighted Degree**: Weighted degree extends degree centrality by incorporating edge weights. For node v and w_{uv}, the weight of the edge connecting node u to node v, it is computed as the sum of the weights of all edges incident to the node:

$$C_{WD}(v) = \sum_{u \in N(v)} w_{vu}. \tag{3.8}$$

3. **Betweenness Centrality**: Betweenness centrality quantifies the importance of a node in facilitating communication between other nodes. Let σ_{st} denote the total number of shortest paths between nodes s and t, and $\sigma_{st}(v)$ denote the number of those paths passing through node v:
$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}. \tag{3.9}$$
4. **Closeness Centrality**: Closeness centrality reflects how close a node is to all other nodes in the graph. For node v, it is defined as the inverse of the total shortest path distance to all other nodes:
$$C_C(v) = \frac{1}{\sum_{u \neq v} d(v, u)}. \tag{3.10}$$
5. **Eigenvector Centrality**: Eigenvector centrality assigns higher importance to nodes connected to other influential nodes. Let A denote the adjacency matrix of the graph and λ its largest eigenvalue. For node v, eigenvector centrality is defined as
$$C_E(v) = \frac{1}{\lambda} \sum_{u \in N(v)} A_{vu} C_E(u). \tag{3.11}$$
6. **PageRank**: PageRank measures the importance of a node based on the idea that connections from important nodes contribute more. Originally developed for ranking web pages. Let $d \in (0, 1)$ denote the damping factor (usually 0.85), N the total number of nodes in the network, $M(v)$ the set of nodes that link to node v, and $L(u)$ the number outgoing links from node u. The PageRank score of node v is computed as
$$PR(v) = \frac{1 - d}{N} + d \sum_{u \in M(v)} \frac{RP(v)}{L(u)}. \tag{3.12}$$
7. **Clustering Coefficient**: The clustering coefficient measures the tendency of a node's neighbors to form connections among themselves. Let e_v denote the number of edges between neighbors of node v:
$$C_{Clust}(v) = \frac{2e_v}{\deg(v)(\deg(v) - 1)}. \tag{3.13}$$
8. **Harmonic Centrality**: Harmonic centrality evaluates node centrality based on reciprocal distances and remains well-defined for disconnected graphs. For node v, it is given by
$$C_H(v) = \sum_{u \neq v} \frac{1}{d(v, u)}. \tag{3.14}$$
9. **Load Centrality**: Measures the importance of a node based on the total weight of its connections to neighboring nodes. A higher value indicates a more influential or "central" node in the network. Here, $C_L(v)$ refers to the "load centrality" of a given node v, representing the total amount of the "load" that the given node has with regards to its connections.
$$C_L(v) = \sum_{u \in N(v)} w_{uv}. \tag{3.15}$$

For each training sample, these features are integrated with the existing feature set based on original attributes, thus forming an enriched representation of the training data.

3.5. Graph-based feature propagation for test data

After obtaining the features based on graph for the training set, a set of equivalent relational features must be obtained for the test instances in order to keep the same representation for the set of features across the entire dataset. Because the derivation of the similarity graph is founded in the observed relations in the training set, a graph-based feature propagation is developed to infer corresponding graph features for unseen test samples. In this way, the set of features for testing will be supplemented with information that is relational, based on their connection with the instances in the training set.

For any test case, similarity scores are calculated between the test case and all other training samples in order to ensure consistence with the graph construction performed on the training samples as well. Based on these scores, the nearest neighbors $k = 3$ are determined from among the training samples. We tested different values of k and average macro F1-score for each, its selection based on the highest observed F1-score.

The graph-based features of the selected neighbors are then propagated to the test instance using a weighted interpolation scheme. Specifically, each neighbor contributes proportionally to the inverse of its distance to the test sample, such that closer neighbors exert a stronger influence. The weights are normalized to ensure that their sum equals one, resulting in a convex combination of the neighboring graph features.

Let x_t denote a test sample and let $\{x_1, x_2, \ldots, x_k\}$ represent its k nearest neighbors in the training set. Each neighbor x_i is associated with a graph-based feature value f_i and a distance d_i from the test sample x_t. The graph-based feature value f_t for the test sample is estimated using inverse distance weighting as follows

$$f_t = \frac{\sum_{i=1}^{k} w_i f_i}{\sum_{i=1}^{k} w_i}, \tag{3.16}$$

where the weight w_i assigned to neighbor x_i is defined as

$$w_i = \frac{1}{d_i + \varepsilon}. \tag{3.17}$$

Here, d_i denotes the distance between the test sample x_t and the *ith* training neighbor, and ε is a small positive constant introduced to avoid division by zero.

After obtaining the estimates of features for all test samples, they are appended to the original feature set to obtain an enriched test data. In this way, structural information will be taken into account during the test process without building a graph on the test samples, thus avoiding the risk of information leakage and ensuring an honest assessment of all classification models to be compared later on. After that step, all that remains to be done is to present the resulting feature space to the learning algorithms for model comparison and evaluation.

3.6. Machine learning model selection and evaluation

Based on the effective representation of features achieved through the incorporation of the original attributes as well as graph features, it is proposed to make use of different existing machine learning classifiers in order to prove the credibility of the proposed set of features. For each of the models proposed in this study, training is performed on the augmented training set in each iteration in order to obtain testing results accordingly.

3.6.1. Machine learning algorithms

A wide range of classification algorithms are used to ensure a comprehensive evaluation. These models span across the paradigm of linearity, non-linearity, probability, tree-based, and ensemble learning [47–49].

- **K-Nearest Neighbors (KNN)** with $k = 5$ is employed as a distance-based classifier that directly benefits from the similarity structure introduced by the graph-based features.
- **Support Vector Machines (SVM)** with linear and radial basis function (RBF) kernels are considered to evaluate the effectiveness of the proposed features under both linear and nonlinear classification settings.
- **Logistic Regression,** a linear classifier that provides a lower bound and is interpretable, and secondly gives a reference point for the gain in performance due to the relational features.
- **Gaussian Naïve Bayes,** included as a probabilistic classifier to examine the behavior of the proposed features under strong independence assumption.
- **Decision Tree** classifiers make use of nonlinear feature interactions by means of hierarchical decision rules.
- **Extra Trees** and **Random Forest** classifiers are adopted as tree based methods to assess robustness and generalization performance in the enriched feature space.
- **AdaBoost**, **Gradient Boosting**, and **Histogram-Based Gradient Boosting** are introduced in order to explore how boosting strategies with iteration may affect classification performance.
- **Bagging** features to reduce the variance by aggregating many base learners trained on bootstrap samples.
- **Extreme Gradient Boosting (XGBoost)** gives an integration of a stat-of-the-art boosting algorithm that is appreciated because of its strong performance and efficient handling of feature interactions in complex scenarios.

3.6.2. Performance evaluation metrics

For an overall assessment of the prediction abilities of classification techniques in multiclass problems, various typical metrics are used [50]. For each of these criteria, results are calculated for every fold and then averaged to get robust estimates.

- **Accuracy:** Calculates the fraction of correctly classified observations out of all instances examined.
- **Precision:** The proportion of correctly predicted positive cases that are actually positive.
- **Recall:** Estimates how well instances from each class are identified as true by the model.
- **F1-Score:** The harmonic mean of precision and recall with respect to the macro-average metric, mainly in multiclass and imbalanced problem.
- **Area Under the ROC Curve (AUC):** This measure is calculated using a one-versus-all method and represents the model's overall discriminative performance across all classes.

4. Results and discussion

This section presents and discusses the experimental results that were achieved by employing the proposed graph based drug classification system. It begins with the exploratory analysis of the dataset, including descriptive statistics, data visualization, outlier analysis, and class distribution before and after balancing. Also, the procedure for choosing the best threshold in constructing the graph are then discussed. Subsequently, various learning models were evaluated in relation to drug classification using cross validation, and it explains the effect of leveraging graph-based features in the models.

4.1. Data analysis and visualization

This subsection shows a comprehensive analysis and visualization of the dataset. The dataset used in this study is the "Drug Classification" dataset, available on the Kaggle website. The dataset consists of 200 clinical data points related to patients, characterized by 6 features such as age, sex, blood pressure (BP), cholesterol, and the sodium-to-potassium ratio (*Na_to_K*), with the final classification into five different drug classes (Drug A, B, C, X, and Y). Table 1 summarizes the dataset's columns, their descriptions, and data types. Furthermore, the data is complete, with no missing values for either variable.

Table 1. Dataset column description and type.

Column	Data Type	Description
Age	Integer	The age of the patient in years
Sex	Object	The gender of the patient (e.g., Male, Female)
BP	Object	Blood pressure level (Low, Normal, High)
cholesterol	Object	Cholesterol level (Normal, High)
Na_to_k	Float	The sodium to potassium ratio in the blood
Drug	Object	The type of drug prescribed to the patient

In Figure 2, it is analyzed how the drugs are being distributed in different patient groups. In the case of the most frequently prescribed medicines, DrugY and DrugX seems relatively balanced between male and female populations. There may be minor variations with respect to the least frequently prescribed drugs. For example, it seems that DrugA seem to be prescribed slightly more often to male patients. However, no significant indication is found that dictates the association between gender and the drug type. A clearer relationship is seen between the blood pressure level and the prescription of drugs. High BP patients are mostly prescribed with DrugY. Also, DrugX is primarily administered to the patients with low or normal BP, and patients with normal BP are nearly treated exclusively with DrugX. DrugA and DrugB also seem to be prescribed largely to the patients with high BP, but less frequently compared to DrugY. Conversely, DrugC seems to be specifically prescribed to patients with low BP. Regarding cholesterol levels, DrugY is prescribed to both patients having high and normal cholesterol. The notable pattern for DrugX is that is highly prescribed to patients with normal level. On the other hand, DrugC seems to be specifically related to patients having high cholesterol.

The boxplot (Figure 3) shows the distribution of the Na_to_K ratio across the different types of drug. The ratio of Na_to_K for DrugY patients is decidedly higher, the median for DrugY is around 20.5, and the interquartile range is approximately between 17.0 and 26.0. The patient group for other drugs are clustered together in a much lower range of Na_to_K, and these medians are close to each other. According to the distribution of age across the drug categories, the patients who used DrugB were relatively the oldest, and the median is represented by the range of late 50s and early 60s. Therefore, age is also one of the crucial attributes for differentiating the categories of the drugs.

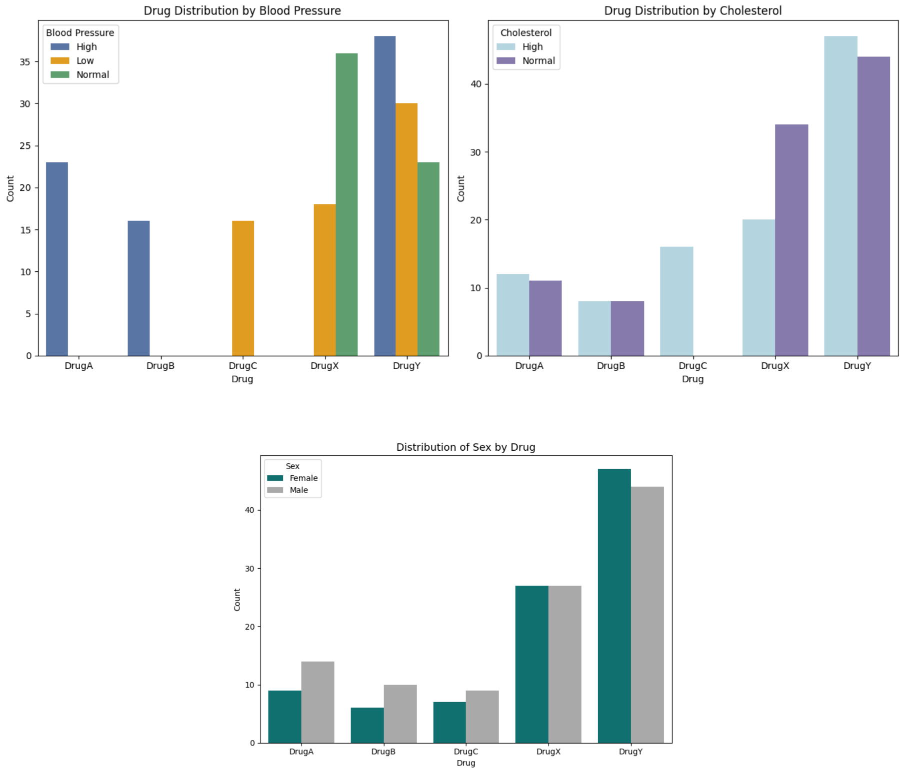


Figure 2. The distribution of gender, BP, and cholesterol by drug categories.

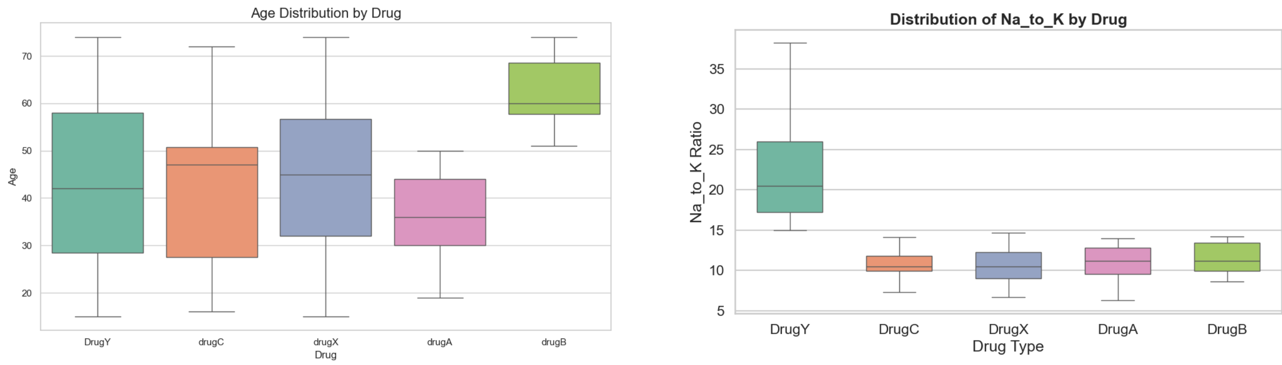


Figure 3. Boxplot visualization of age and Na_to_K ratio for each drug category.

4.1.1. Outlier detection and balancing

To detect and remove unusual data points which might affect the training of the models negatively, the Interquartile Range (IQR) technique was used to detect outliers in the numerical features. However, the outliers also appear in the boxplot of Na_to_K in Figure 4, the boxplot shows some extreme values above the whiskers for this particular feature, indicating the presence of high value outliers for this variable. After imputation, the identified outliers were replaced by the median value of the respective feature. This preprocessing step reduces any influences of outliers and results in a generally more balanced distribution of Na_to_K.

Table 2 presents the class distribution of the dataset before and after using the SMOTEENN method. Before resampling, the dataset exhibited a significant class imbalance. The majority class (DrugY) contained 91 instances, while other minority classes such as DrugC and DrugB had just 16 samples each. This imbalance could bias machine learning models toward the majority class, thereby influencing their performance. For this purpose, the SMOTEENN technique was used. SMOTE is the synthetic minority over-sampling technique, which generates synthetic samples for minority classes, while ENN is the edited nearest neighbors algorithm, which cleans the data set by removing noisy data points. The combination aims to both increase minority representation and improve class boundary quality.

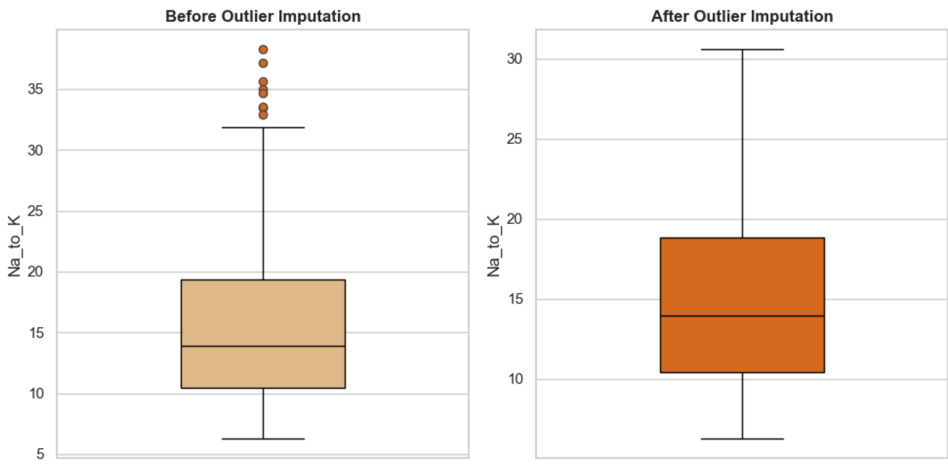


Figure 4. A boxplot comparison of the data before and after treatment of outliers using median imputation.

Table 2. Distribution of drug class before and after SMOTEENN technique.

Drug Class	Original Count	Resampled Count
DrugY	91	46
DrugX	54	81
DrugA	23	82
DrugC	16	90
DrugB	16	91

After resampling, the classes were much more uniformly distributed. Most classes contained between 81 and 91 samples, while DrugY was reduced to 46 instances due to ENN-based cleaning. A significant reduction was observed with respect to the original imbalance ratio. Although the classes are not perfectly equal, the new distribution reflects a considerably improved balance. This adjustment is expected to enhance model generalization and improve performance metrics such as macro-averaged F1-score and balanced accuracy, especially for the underrepresented classes

4.1.2. Sensitivity analysis for threshold selection

The elbow curve analysis was conducted for three distance measures: Gower, Euclidean, and Cosine to determine the optimal threshold that could maximize graph modularity while keeping the graph connected (see Figure 5). Considering the Gower distance, the connected graph with the highest modularity of (0.72) was obtained at the 12th percentile threshold, the graph resulted in a single connected component with 8931 edges and an average degree of 45.8. In the case of Euclidean distance, the optimal modularity (0.720) occurred at the 13th percentile threshold, leading to one connected component with 9692 edges and an average degree of 49.70. Similarly, for the Cosine distance, the highest modularity (0.678) was obtained at the 14th percentile threshold, which returned a fully connected graph with 10,452 edges and an average degree of 53.60. Therefore, while all three distances were able to return a fully connected graph at their respective optimal thresholds, Euclidean distance had the highest modularity value.

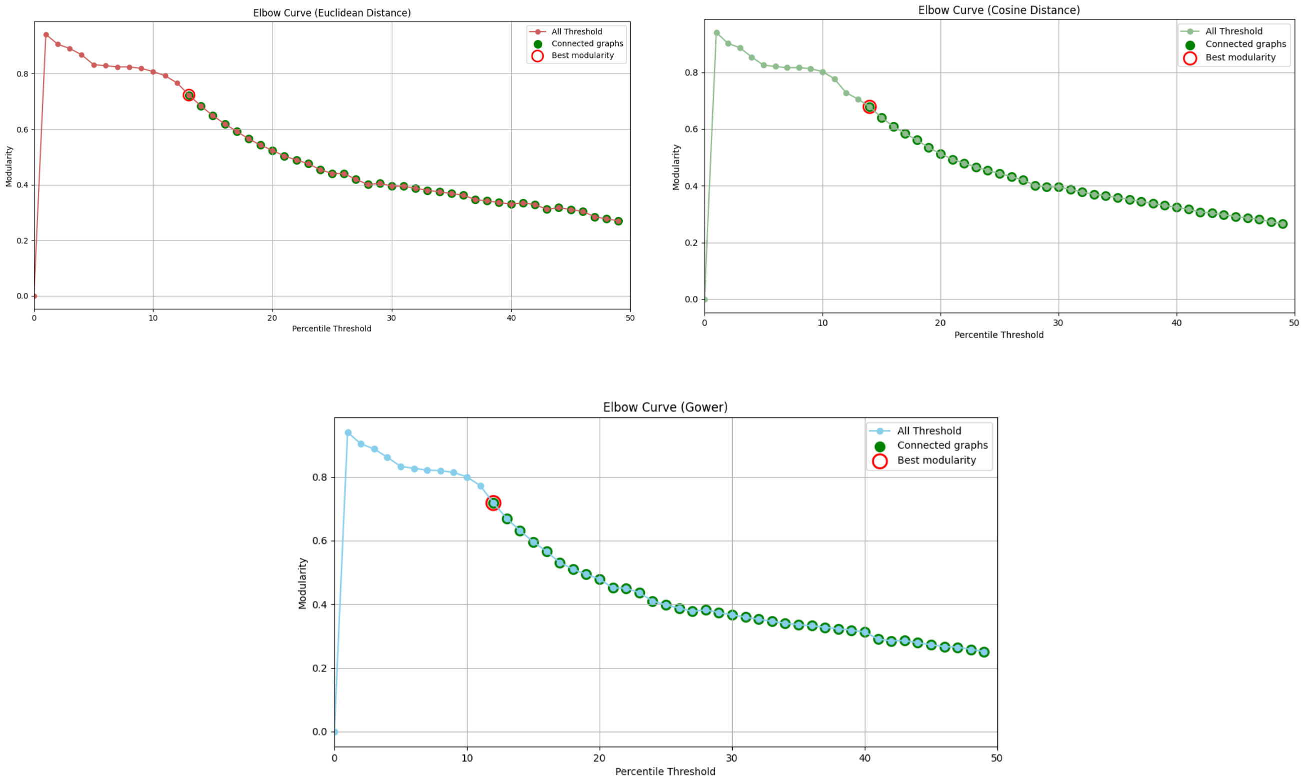
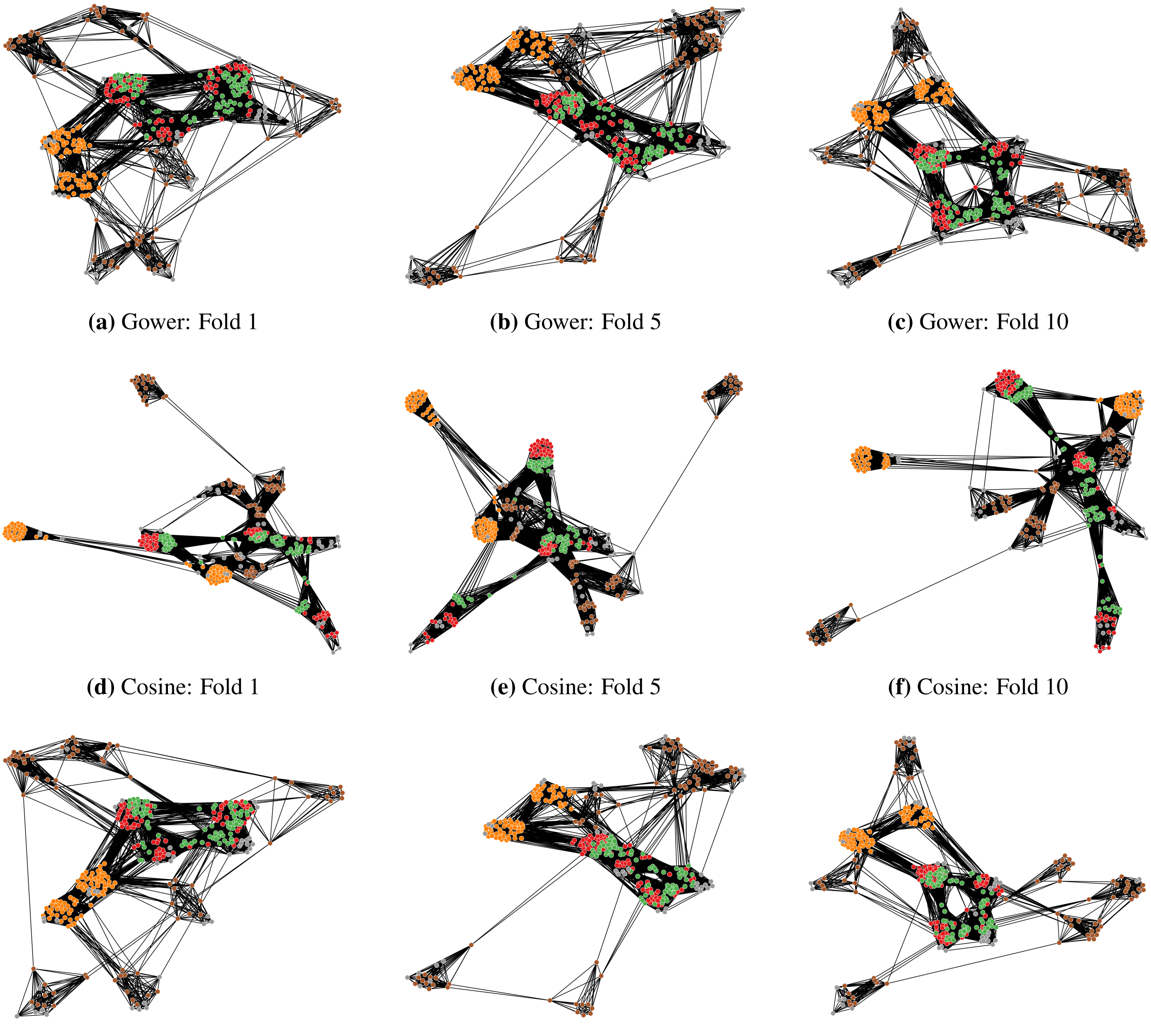


Figure 5. Elbow curves of Gower, Cosine, and Euclidean distance used for computing the optimal threshold of similarity measure.

4.1.3. Graph construction on training data

To further reinforce the validity of the feature extraction process based on the proposed graph model, the research data was converted into a topological structure during each iteration of the training process. Figure 6 shows the resulting topology for the research data in folds 1, 5, and 10, using all the similarity metrics. From the visualizations, it is evident that the data is transformed from a table format into a more connected structure, where each data point is represented by a node that is connected to other nodes if the similarity is above a certain percentile threshold. From the visualizations in Figure 6, it is evident that a distinct community is formed in all the data sets, where the nodes of the same class are highly connected. Moreover, the visual similarity of the data in Folds 1, 5, and 10 further reinforces the stability of the process of building the graphs during the cross-validation process, ensuring that the resulting extracted centrality features are derived from reproducible topological patterns.



(a) Gower: Fold 1 **(b)** Gower: Fold 5 **(c)** Gower: Fold 10

(d) Cosine: Fold 1 **(e)** Cosine: Fold 5 **(f)** Cosine: Fold 10

(g) Euclidean: Fold 1 **(h)** Euclidean: Fold 5 **(i)** Euclidean: Fold 10

Figure 6. Topological graph networks of the training datasets across folds 1, 5, and 10 for three similarity measures.

4.2. The performance using balancing before cross validation

Overall, these results indicate that the proposed methodology performs strong and consistent performance across multiple classifiers in all three distance measures. Numerical performance of each model by applying balancing before cross-validation using Gower distance is summarized in Table 3, using Euclidean distance in Table 4, and Cosine distance in Table 5. In addition, for better comparison summaries among the above measures, a corresponding bar chart visualization is given (Figures 7, 8, 9).

In general, the outcomes suggest that the proposed framework achieve strong, consistent performance among multiple classifiers, especially the gradient-boosting–based models. Most models received a high score in macro-average, thus predicting well for most classes despite class imbalance.

Table 3. Performance of models using Gower including accuracy, macro-averaged precision, recall, F1-score, and AUC for each model.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
HistGradientBoosting	0.9974	0.9978	0.9978	0.9976	1.0000
GradientBoosting	0.9974	0.9978	0.9978	0.9976	1.0000
DecisionTree	0.9974	0.9980	0.9950	0.9961	0.9972
Bagging	0.9923	0.9940	0.9870	0.9889	1.0000
XGBoost	0.9923	0.9936	0.9888	0.9903	0.9999
RandomForest	0.9795	0.9839	0.9660	0.9698	0.9999
ExtraTrees	0.9821	0.9857	0.9700	0.9750	1.0000
LogisticRegression	0.9692	0.9750	0.9548	0.9598	0.9977
SVM_linear	0.9692	0.9750	0.9533	0.9584	0.9996
GaussianNB	0.9692	0.9765	0.9590	0.9641	0.9979
KNN	0.7718	0.7719	0.7306	0.7291	0.9389
AdaBoost	0.6179	0.6514	0.6572	0.5869	0.8719
SVM_rbf	0.5590	0.3473	0.5024	0.4077	0.8538

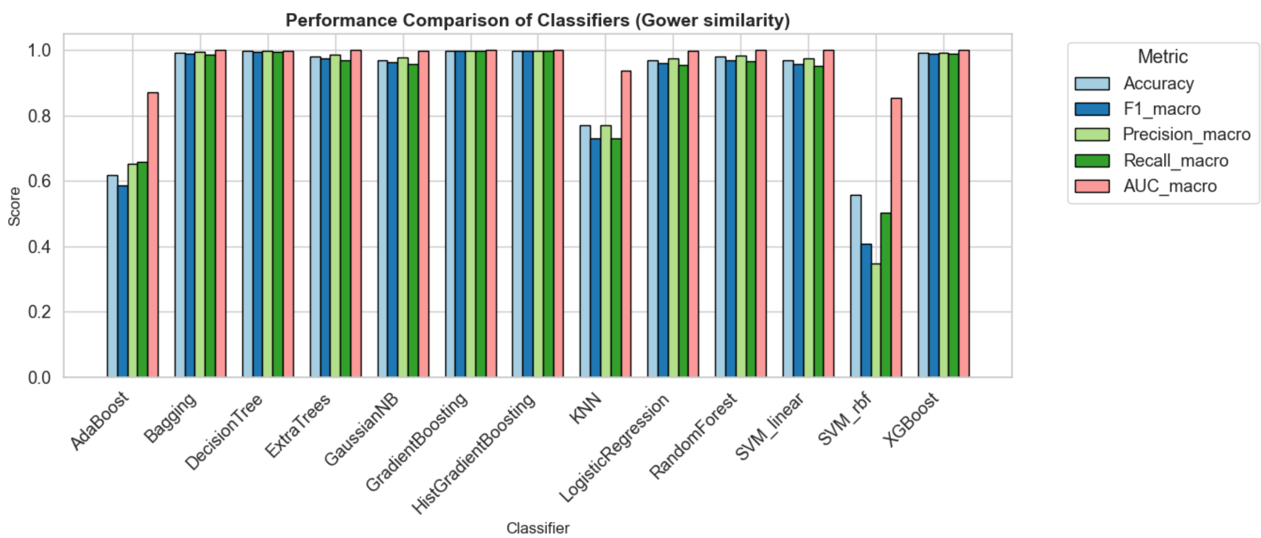


Figure 7. Bar chart demonstrating the performance of the best models in classifying using the Gower distance metric.

Table 4. Classification results of each of these models computed using the Euclidean distance.

Model	Accuracy	Precision	Recall	F1-Score	AUC
DecisionTree	1.0000	1.0000	1.0000	1.0000	1.0000
GradientBoosting	0.9974	0.9978	0.9978	0.9976	1.0000
HistGradientBoosting	0.9949	0.9958	0.9953	0.9953	1.0000
XGBoost	0.9949	0.9958	0.9928	0.9937	1.0000
Bagging	0.9949	0.9960	0.9910	0.9928	1.0000
RandomForest	0.9897	0.9913	0.9830	0.9859	1.0000
ExtraTrees	0.9821	0.9851	0.9728	0.9764	1.0000
GaussianNB	0.9692	0.9761	0.9571	0.9628	0.9972
LogisticRegression	0.9667	0.9728	0.9503	0.9560	0.9946
SVM_linear	0.9641	0.9702	0.9483	0.9529	0.9960
KNN	0.8000	0.8051	0.7754	0.7759	0.9460
AdaBoost	0.6128	0.6233	0.6547	0.5779	0.8798
SVM_rbf	0.5769	0.4307	0.5194	0.4557	0.8582

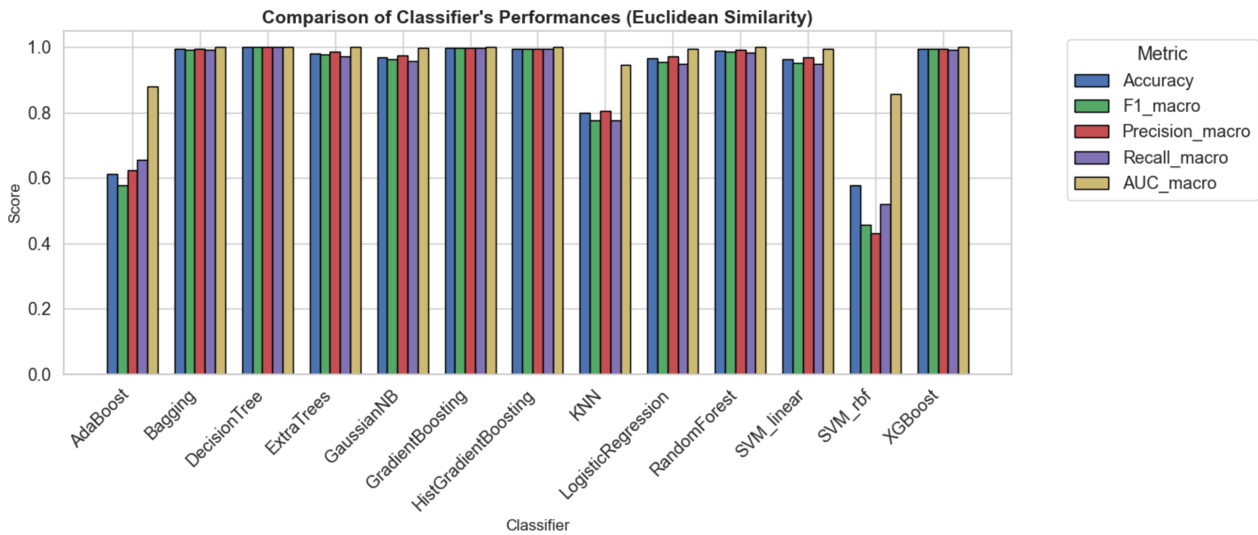


Figure 8. Bar chart showing performance comparison of top classifiers based on Euclidean distance metric.

Table 5. Results of evaluation of classifiers based on Cosine distance.

Model	Accuracy	Precision	Recall	F1-Score	AUC
GradientBoosting	1.0000	1.0000	1.0000	1.0000	1.0000
HistGradientBoosting	0.9974	0.9980	0.9975	0.9976	1.0000
Bagging	0.9949	0.9958	0.9938	0.9944	1.0000
DecisionTree	0.9949	0.9958	0.9938	0.9944	0.9962
XGBoost	0.9949	0.9960	0.9925	0.9937	1.0000
RandomForest	0.9897	0.9913	0.9840	0.9865	1.0000
ExtraTrees	0.9821	0.9851	0.9725	0.9763	0.9998
LogisticRegression	0.9538	0.9651	0.9355	0.9419	0.9960
SVM_linear	0.9538	0.9641	0.9405	0.9452	0.9947
GaussianNB	0.9385	0.9446	0.9259	0.9277	0.9896
KNN	0.8179	0.8004	0.7740	0.7626	0.9195
AdaBoost	0.7641	0.7534	0.7838	0.7465	0.9710
SVM_rbf	0.4795	0.3051	0.4282	0.3420	0.7855

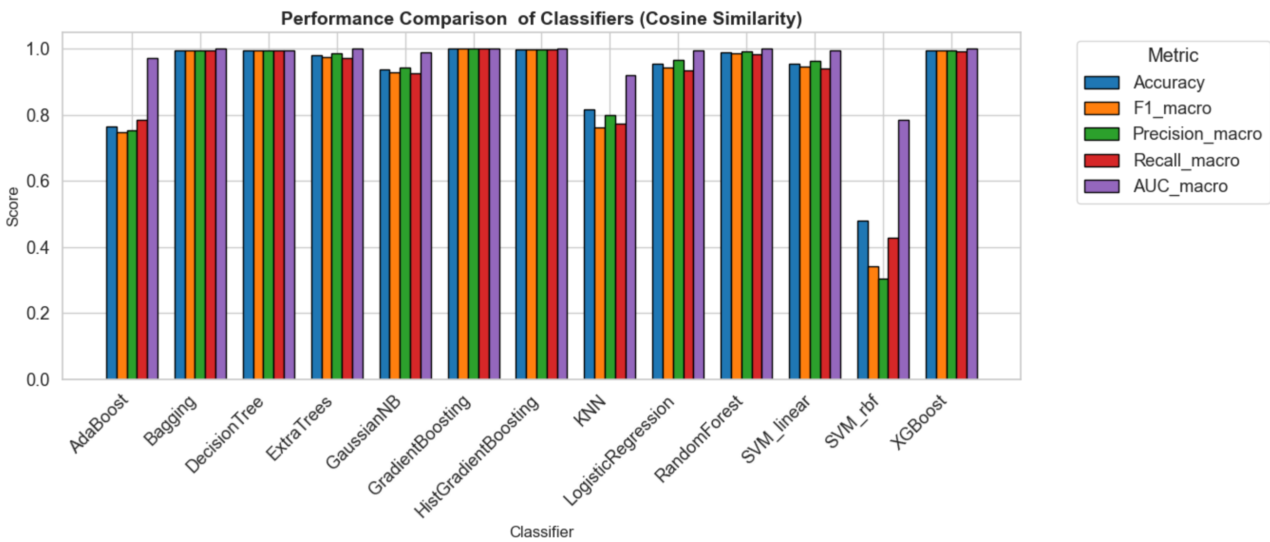


Figure 9. Comparison of evaluation metrics (using Cosine distance).

4.3. *Balancing within cross-validation*

In a strict evaluation framework aimed at minimizing bias when assessing model performance, resampling methods were applied exclusively to the training folds within the cross-validation procedure. This approach prevents any potential information leakage and ensures a highly realistic assessment of the model's generalization ability. Applying the SMOTEENN technique to the training folds resulted in notable changes in the class balance. Even though the technique managed to balance the minority classes, it also reduced the majority class samples. Specifically, in several folds, the majority class had around 80 samples before resampling, but only about 35 samples after applying the technique. The ENN part of the algorithm was responsible for removing noisy samples that may have led to misclassification. However, this can be beneficial in large datasets, but in this case study, since the sample size was only 200 samples, this aggressive approach resulted in the deletion of many real and useful samples.

In this case, SMOTE showed a more balanced and controlled approach. The use of synthetic examples that were generated by the model itself but did not replace real examples allowed SMOTE to keep the natural distribution of data, reducing at the same time its class imbalance. As a result, the performance gained with the help of SMOTE became more consistent.

The results for the classifiers tested with balancing within the cross-validation are shown in Tables 6,7, and 8.

Table 6. Model performance using Gower distance with balancing within cross-validation folds.

Model	Accuracy	Precision	Recall	F1-Score	AUC
GradientBoosting	0.9650	0.9597	0.9433	0.9448	0.9978
Bagging	0.9500	0.9407	0.9367	0.9284	0.9993
HistGradientBoosting	0.9400	0.9382	0.9204	0.9179	0.9931
XGBoost	0.9300	0.9068	0.8978	0.8890	0.9981
DecisionTree	0.9300	0.9204	0.9111	0.9012	0.9454
RandomForest	0.9250	0.8955	0.9313	0.8988	0.9917
ExtraTrees	0.8600	0.8252	0.9104	0.8428	0.9833
AdaBoost	0.8550	0.7882	0.8211	0.7863	0.9649
SVM_linear	0.8200	0.7926	0.8898	0.8081	0.9874
LogisticRegression	0.8150	0.7869	0.8987	0.8080	0.9731
GaussianNB	0.7550	0.7081	0.8449	0.7317	0.9377
KNN	0.5750	0.5739	0.7253	0.5736	0.8791
SVM_rbf	0.3500	0.3518	0.4160	0.3163	0.6732

Table 7. Classification results based on Euclidean distance under fold-wise resampling.

Model	Accuracy	Precision	Recall	F1-Score	AUC
GradientBoosting	0.9550	0.9285	0.9167	0.9122	0.9961
Bagging	0.9450	0.9268	0.9258	0.9132	0.9934
DecisionTree	0.9400	0.9334	0.9138	0.9102	0.9478
HistGradientBoosting	0.9350	0.9265	0.9189	0.9098	0.9945
XGBoost	0.9300	0.9251	0.9011	0.8953	0.9986
AdaBoost	0.9250	0.9099	0.9011	0.8920	0.9885
RandomForest	0.9050	0.8710	0.9180	0.8781	0.9825
ExtraTrees	0.8250	0.8026	0.8816	0.8196	0.9707
LogisticRegression	0.8100	0.8015	0.8787	0.8171	0.9573
SVM_linear	0.7150	0.7145	0.8236	0.7162	0.9613
GaussianNB	0.7000	0.6832	0.8069	0.6794	0.9398
KNN	0.5850	0.5924	0.7029	0.5783	0.8653
SVM_rbf	0.4600	0.3964	0.5798	0.4183	0.8350

Table 8. Performance evaluation of classifiers using Cosine similarity with resampling applied only on training data.

Model	Accuracy	Precision	Recall	F1-Score	AUC
GradientBoosting	0.9600	0.9721	0.9471	0.9497	0.9992
XGBoost	0.9450	0.9621	0.9404	0.9394	0.9988
Bagging	0.9400	0.9405	0.9480	0.9318	0.9981
HistGradientBoosting	0.9400	0.9402	0.9558	0.9372	0.9964
DecisionTree	0.9300	0.9421	0.9249	0.9195	0.9519
RandomForest	0.9250	0.9128	0.9580	0.9222	0.9964
LogisticRegression	0.9200	0.9078	0.9558	0.9180	0.9835
ExtraTrees	0.8800	0.8644	0.9320	0.8769	0.9916
AdaBoost	0.8700	0.7648	0.7960	0.7673	0.9714
SVM_linear	0.8700	0.8643	0.9169	0.8678	0.9896
GaussianNB	0.7900	0.7989	0.8580	0.7983	0.9620
KNN_5	0.5650	0.5371	0.6176	0.5130	0.8547
SVM_rbf	0.5050	0.4054	0.4093	0.3541	0.7714

4.4. Comparison of machine learning models

Starting with first case, which is using resampling technique before splitting the dataset, Table 3 shows the performance of models developed based on Gower distance. The best predictive performance obtained was for boosting methods, especially HistGradientBoosting and GradientBoosting with an accuracy of 99.74%, a macro F1-score of 0.9976, and AUC of 1.0000. GradientBoosting and extra trees follow it, with achieved accuracies of 99.23% and 98.21%, respectively, with perfect 1.0000 AUC. The ensemble methods random forest, extra trees, and

XGBoost performed very well with F1-scores greater than 0.96 and near perfect AUC values. This benefited from an enriched feature space combining original attributes with graph-based features, enabling them to model complex relationships easily. Linear and probabilistic models also show competitive performance, such as logistic regression and Gaussian naïve Bayes, with accuracy of about 0.9692.

In contrast, distance-based methods are more variable: KNN with k = 5 yields a moderate performance, accuracy = 0.7718, while for increasing k-value the performance decreases, indicative of large neighborhoods that dilute meaningful similarity. For the case of SVM, the classifier is sensitive to kernel choice, using the linear kernel results in reasonable performance, accuracy = 0.9692, while the performance of the RBF kernel significantly lags behind at accuracy = 0.559, which suggests sensitivity to hyperparameters and feature scaling.

In addition, Table 4 presents the performance of the classifiers using Euclidean distance. Noticeable results were obtained, including decision tree, with an accuracy measure of 100%, making it the best performing algorithm, followed by boosting methods and other classifiers. Interestingly, Euclidean distance improved some model performances slightly compared to Gower distance, including KNN (accuracy 77% vs. 80%), random forest (accuracy 97.94% vs. 98.97%), and SVM_rbf (accuracy 55.89% vs. 57.69%). These results also indicated that using Euclidean distance captured the related similarity among samples well, especially for tree-based models, and achieved very stable performance for other classic models.

Furthermore, the performance of the classifiers using Cosine distance is depicted in Table 5, the performance of gradient boosting is perfect with accuracy 100% and AUC equals 1.000. Decision tree also works very well, though a bit worse than Euclidean: accuracy = 0.9949. In summary, the Cosine distance has slightly worse performance for a distance and linear model, compared to Euclidean and Gower, but performs quite well on tree-based and ensemble classifiers.

In order to avoid any biases, class balancing was performed only within the training folds during cross-validation ensuring a completely leakage-free evaluation procedure of the proposed methodology. Though the data size is quite small, such an approach successfully ensured balancing of classes, without any elimination of original data instances for model training. As shown in Tables 6, 7, and 8, most of the classifiers were able to achieve high metrics consistently, with regards to accuracy, F1-score, and AUC. The results show that the majority of the models have performed well across multiple evaluation metrics. From the evaluated models, gradient boosting achieved the best performance, obtaining accuracy of around 97% with near-perfect AUC values, proving to be strong at discriminating the data. Also, the majority of the other models had an accuracy value above 90% and AUC scores close to 1.0.

Thus, in-fold resampling was seen to be an effective way to ensure a true representation of the data distribution, resulting in a realistic evaluation of generalization ability of the proposed framework. On the contrary, it allows successful training while avoiding any biases. Also, ensemble-based approaches like gradient boosting, XGBoost, and bagging were found to perform best using graph features in conjunction with balanced training data. This indicates that even when resampling is applied correctly, the proposed graph-based framework remains highly effective for multiclass drug classification, highlighting its potential for application in small biomedical datasets.

4.5. Discussion

The graph-based approach to drug classification was evaluated using different types of distances (Gower, Euclidean, and Cosine) and different classifiers. Initially, resampling was performed before cross-validation, and such an experimental arrangement has been frequently used in several of previous works. The obtained performance characteristics are exceptionally high, as almost all of the models demonstrate almost perfect results in terms of accuracy, F1-score, and AUC.

This observation is further supported by the bar chart graphs Figures 7, 8, and 9, in which the gains in performance are shown to favor models that leverage the relational information. Importantly, in the graph-based feature propagation technique, test samples can take advantage of relational information without building a graph over test data itself, and this approach maintains a strict division between the training and test sets, thereby eliminating leakage of knowledge from the training data.

However, the use of various similarity measures in this study is aimed at assessing the impact of various definitions of similarity on the structure of the graph and the performance of the classification. Each similarity measure is independently applied to generate a similarity graph, and the entire procedure is repeated for each one. Euclidean distance is a measure of the proximity between points in the feature space. This measure is usually applicable for continuous numerical features. Cosine similarity is a measure of the similarity between the patterns of the samples. This measure is usually applicable for cases where the magnitude of the features does not matter, but the proportion does. Gower distance is a measure that can handle both mixed and homogeneous data types. This means a unified similarity can be computed for cases where the data is both categorical and numerical. By evaluating these various similarity measures, the study aimed at assessing the impact and the role of multiple structural representations of the data on graph topology, feature extraction, and performance of the classification. Also, it is critically important to use cross validation to provide a realistic approximation of model performance in order to ensure the models are robust to unseen data, as all of the available data is used both for the training and testing across multiple splits.

In [37], the accuracy of 0.975 was achieved by the random forest and decision tree algorithms, but our results in the first case demonstrate even better performance, with the decision tree having a perfect accuracy of 1.0 by Euclidean distance and up to 0.97 in random forest. Although the second study [38] demonstrated the gradient boosting method with 96.9% accuracy and 96.8% F1-score, our models of gradient boosting performed better, achieving near-perfect accuracy of 0.997 to 1.0 across all distance measures. Moreover, our accuracy of about 0.982 in extra trees is a little more than the 98% accuracy reported in [39], which also stated that some ensemble methods were capable of achieving 100% accuracy, which was mirrored by our result of bagging, gradient boosting, and XGBoost. In conclusion, this comparison clearly shows that our approach yields improvement in accuracy over the methods described in the previous papers.

Although the outcomes prove the discrimination ability of the proposed graph features, using resampling techniques prior to cross-validation causes optimism bias because of possible information leakage in the process. In order to address this problem, balancing also was applied on a per-fold basis in cross-validation as well. This way, synthetic instances were created based only on the training set of each fold, thus avoiding any possible leakage, which is a more realistic assessment of generalization ability. Using fold-wise for balancing, the achieved results show that despite a more rigorous approach, the performance remains high, as visually summarized in the radar chart in Figure 10. For example, ensembles of algorithms like gradient boosting, XGBoost, and bagging retained an accuracy over 0.95 and AUC 0.999. When comparing the results of pre-CV and fold-wise CV, one can see that proper resampling is essential. Pre-CV provides almost perfect accuracy rates, but at the same time may result in leakage. The fold-wise CV results are somewhat smaller, yet they give reliable estimates without any bias, which shows that the method works well in spite of all the challenges faced. This means that the graph-based features play an important role in classification tasks.

Nevertheless, it must be noted that despite such successful results, the number of samples in the current dataset is still relatively small and could cause overly optimistic estimates of performance for some models. Future work will be aimed at validating the proposed approach on larger and independent datasets, as well as finding adaptive strategies for graph representation and model parameter optimization to ensure generalizability and improve robustness across different data distributions.

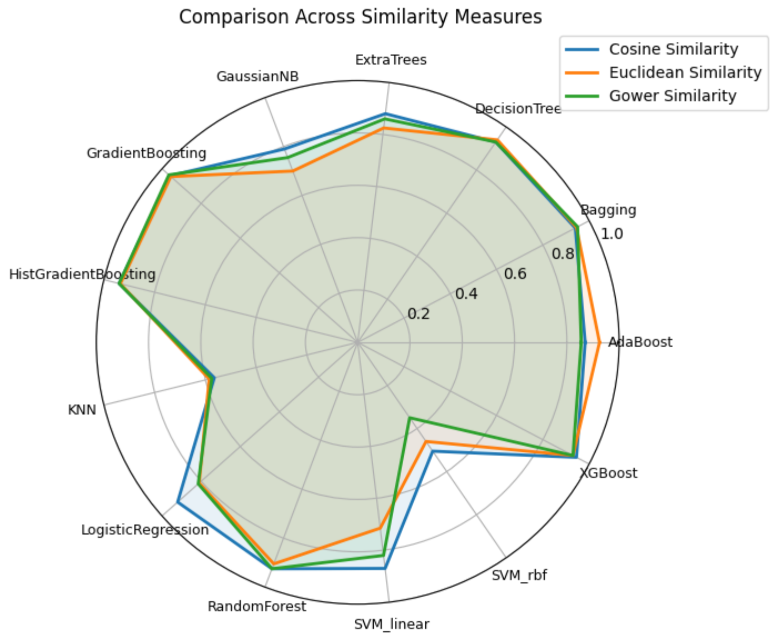


Figure 10. Radar chart summarizing performance across distinct similarity measures for balancing inside cross validation scenario.

5. Conclusion

This work shows the effectiveness of graph-based feature engineering in machine learning for multiclass drug classification tasks. The proposed framework integrates traditional patient-level attributes with the graph-based features to compute latent relationships among patients not explicitly captured by the original feature space. This enriched representation will allow classification models to better exploit individual characteristics and relational patterns present in the data.

The proposed approach will then convert patient samples into nodes in a similarity graph that is built using the Gower, Cosine, and Euclidean distance metrics. Graph theoretic features extracted from the training graph include degree-based measures, centrality indices, and clustering coefficients, which are combined with the original attributes. In order not to violate the principles of data separation for extending relational information onto unseen samples, a weighted feature propagation strategy based on nearest training neighbors is adapted for test data. This assures fairness in the evaluation process with leakage.

For evaluating our experiments, we used a 10-fold cross validation technique. The outcome of our experiments proves that a significant improvement is achieved by considering graph-based attributes with a variety of machine learning algorithms. It observed that the outcome reveals the performance of each distance metric had perfect or near-perfect classification performance, specifically perfect performance for decision tree and gradient boosting by using the Euclidean and Cosine similarity distances, respectively. These findings therefore have important practical implications, suggesting that relational modeling can play a valuable role within medical decision-support systems. Graph-based feature extraction that uncovers hidden similarities between patients can enhance the robustness and interpretability of drug classification models, with the possibility of more informed and accurate treatment decisions. Despite the positive outcomes, there are some limitations that need to be noted. First, it should be mentioned that, even though the dataset used in the proposed study is small, it may create an optimistic bias with certain algorithms. Moreover, the process of feature extraction with the knowledge from the test data using a fixed range of neighborhood parameters (k) may not be optimal for all datasets or data distributions. In addition, the proposed methodology has been evaluated on a single drug classification dataset, which may limit the generalizability of the findings. Future work may include extending the use of the proposed framework to larger drug classification datasets with varied dimensions, which can further test the robustness of the proposed framework. Moreover, more complex parameter optimization methods may be used. Alternative methods for constructing graphs may also be investigated to improve the quality of constructed graphs. Advanced approaches using graphs, such as graph neural networks (GNNs), may also allow us to learn node representations directly from raw data, thus avoiding the need for manually designed features. Hybrid approaches, using combinations of conventional machine learning methods and deep graph representations, may also represent a promising direction for future research.

Acknowledgments

We would like to express our gratitude to our respective universities for the outstanding facilities and resources they provide us with. These supports are greatly appreciated.

Conflict of interest

The authors declare no conflict of interest.

References

1. Safdari R, Esmaeili M, Marashi Shooshtari SS, et al. (2021) Drug classification systems: applications and characteristics. *Health Manage Inf Sci* 8: 149–158. https://doi.org/10.30476/jhmi.2022.91329.1083
2. Ashley EA (2016) Towards precision medicine. *Nat Rev Genet* 17: 507–522. https://doi.org/10.1038/nrg.2016.101

3. Gururaj HL, Flammini F, Kumari HC, et al. (2021) Classification of drugs based on mechanism of action using machine learning techniques. *Discover Artif Intell* 1: 13. https://doi.org/10.1007/s44163-021-00012-2
4. Askr H, Elgeldawi E, Aboul Ella H, et al. (2023) Deep learning in drug discovery: an integrative review and future challenges. *Artif Intell Rev* 56: 5975–6037. https://doi.org/10.1007/s10462-022-10306-1
5. Esteva A, Robicquet A, Ramsundar B, et al. (2019) A guide to deep learning in healthcare. *Nat Med* 25: 24–29. https://doi.org/10.1038/s41591-018-0316-z
6. Obaido G, Mienye ID, Egbelowo OF, et al. (2024) Supervised machine learning in drug discovery and development: algorithms, applications, challenges, and prospects. *Machine Learn Appl* 17: 100576. https://doi.org/10.1016/j.mlwa.2024.100576
7. Zhao H, Zhong J, Liang X, et al. (2025) Application of machine learning in drug side effect prediction: databases, methods, and challenges. *Front Comput Sci* 19: 195902. https://doi.org/10.1007/s11704-024-31063-0
8. Chen C (2024) Research on drug classification using machine learning model. *Highlights Sci Eng Technol* 81: 350–355. https://doi.org/10.54097/nfpj0845
9. Gallo K, Goede A, Preissner R, et al. (2022) SuperPred 3.0: drug classification and target prediction—a machine learning approach. *Nucleic Acids Res* 50: W726–W731. https://doi.org/10.1093/nar/gkac297
10. Newman M (2018) Networks. Oxford: Oxford University Press.
11. Barabási AL, Gulbahce N, Loscalzo J (2011) Network medicine: a network-based approach to human disease. *Nat Rev Genet* 12: 56–68. https://doi.org/10.1038/nrg2918
12. Zitnik M, Agrawal M, Leskovec J (2018) Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34: i457–i466. https://doi.org/10.1093/bioinformatics/bty294
13. Costa LdF, Rodrigues FA, Travieso G, et al. (2007) Characterization of complex networks: a survey of measurements. *Adv Phys* 56: 167–242. https://doi.org/10.1080/00018730601170527
14. Li MM, Huang K, Zitnik M (2022) Graph representation learning in biomedicine and healthcare. *Nat Biomed Eng* 6: 1353–1369. https://doi.org/10.1038/s41551-022-00930-6
15. Gaudelet T, Day B, Jamasb AR, et al. (2021) Utilizing graph machine learning within drug discovery and development. *Brief Bioinform* 22: bbab159. https://doi.org/10.1093/bib/bbab159
16. Dibaji A and Sulaimany S (2023) Improving machine learning classification of heart disease using graph-based techniques, *Proceedings of the 2023 13th International Conference on Computer and Knowledge Engineering (ICCKE),* IEEE, 474–479. https://doi.org/10.1109/ICCKE58845.2023.10326444
17. Mafakheri A and Sulaimany S (2024) Android malware detection through centrality analysis of applications network. *Appl Soft Comput* 165: 112058. https://doi.org/10.1016/j.asoc.2023.112058
18. Opsahl T, Agneessens F, Skvoretz J (2010) Node centrality in weighted networks: generalizing degree and shortest paths. *Soc Networks* 32: 245–251. https://doi.org/10.1016/j.socnet.2010.03.006

19.Levy A, Shalom BR, Chalamish M (2025) A guide to similarity measures and their data science applications. *J Big Data* 12: 188. https://doi.org/10.1186/s40537-025-01227-1

20.Gower JC (1971) A general coefficient of similarity and some of its properties. *Biometrics* 27: 857–871. https://doi.org/10.2307/2528823

21.Vamathevan J, Clark D, Czodrowski P, et al. (2019) Applications of machine learning in drug discovery and development. *Nat Rev Drug Discov* 18: 463–477. https://doi.org/10.1038/s41573-019-0024-5

22.Hodos RA, Kidd BA, Shameer K, et al. (2016) In silico methods for drug repurposing and pharmacology. *Wiley Interdiscip Rev Syst Biol Med* 8: 186–210. https://doi.org/10.1002/wsbm.1337

23.Lo YC, Rensi SE, Torng W, et al. (2018) Machine learning in chemoinformatics and drug discovery. *Drug Discov Today* 23: 1538–1546. https://doi.org/10.1016/j.drudis.2018.05.010

24.Lavecchia A (2015) Machine-learning approaches in drug discovery: methods and applications. *Drug Discov Today* 20: 318–331. https://doi.org/10.1016/j.drudis.2014.10.012

25.Liu Y, Tang H, Niu T, et al. (2026) A comparative study of deep learning and classical modeling approaches for protein-ligand binding pose and affinity prediction in coronavirus main proteases. *J Chem Inf Model* 66: 731–743. https://doi.org/10.1021/acs.jcim.5c02481

26.Zhang Y, Liu J, Shen W (2022) A review of ensemble learning algorithms used in remote sensing applications. *Appl Sci* 12: 8654. https://doi.org/10.3390/app12178654

27.Shi H, Liu S, Chen J, et al. (2019) Predicting drug-target interactions using lasso with random forest based on evolutionary information and chemical structure. *Genomics* 111: 1839–1852. https://doi.org/10.1016/j.ygeno.2018.12.007

28.Chen F, Zhao Z, Ren Z, et al. (2025) Prediction of drug target interaction based on under sampling strategy and random forest algorithm. *PLoS One* 20: e0318420. https://doi.org/10.1371/journal.pone.0318420

29.Thafar MA, Olayan RS, Albaradei S, et al. (2021) DTi2Vec: drug–target interaction prediction using network embedding and ensemble learning. *J Cheminform* 13: 71. https://doi.org/10.1186/s13321-021-00552-w

30.Mustapha IB and Saeed F (2016) Bioactive molecule prediction using extreme gradient boosting. *Molecules* 21: 983. https://doi.org/10.3390/molecules21080983

31.Zhou J, Cui G, Hu S, et al. (2020) Graph neural networks: a review of methods and applications. *AI Open* 1: 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

32.Wang X, Hu T, Yang Q, et al. (2021) Graph-theory based degree centrality combined with machine learning algorithms can predict response to treatment with antiepileptic medications in children with epilepsy. *J Clin Neurosci* 91: 276–282. https://doi.org/10.1016/j.jocn.2021.07.016

33.Hosseini M, Dibaji A, Sulaimany S (2024) Graph-based feature engineering for rolling element bearing fault diagnosis using vibration signals. *Eng Res Express* 6: 045234 https://doi.org/10.1088/2631-8695/ad8ff0

34.Renjini A, Swapna MS, Raj V, et al. (2021) Graph-based feature extraction and classification of wet and dry cough signals: a machine learning approach. *J Complex Netw* 9: cnab039. https://doi.org/10.1093/comnet/cnab039

35.Dibaji A and Sulaimany S (2023) Improving machine learning classification of heart disease using graph-based techniques, *Proceedings of the 2023 13th International Conference on Computer and Knowledge Engineering (ICCKE),* IEEE, 474–479. https://doi.org/10.1109/ICCKE58845.2023.10326444

36.Albreiki B, Habuza T, Zaki N (2023) Extracting topological features to identify at-risk students using machine learning and graph convolutional network models. *Int J Educ Technol High Educ* 20: 23. https://doi.org/10.1186/s41239-023-00389-3

37.Gala DV, Gandhi VB, Gandhi VA, et al. (2021) Drug classification using machine learning and interpretability, *2021 Smart Technologies, Communication and Robotics (STCR)*, Sathyamangalam, India, 1–8. https://doi.org/10.1109/STCR51658.2021.9588972

38.Vu TA, Hieu TM, Linh HTM, et al. (2023) Drug classification based on machine learning models with a combination of data binning and SMOTE technique, *2023 1st International Conference on Health Science and Technology (ICHST)*, Hanoi, Vietnam, 1–5. https://doi.org/10.1109/ICHST59286.2023.10565309

39.Purwono P, Wirasto A, Nisa K (2021) Comparison of machine learning algorithms for classification of drug groups. *Sisfotenika* 11: 196–207. https://doi.org/10.30700/jst.v11i2.1134

40.Drug200 dataset (2020) Kaggle. Available from: `https://www.kaggle.com/datasets/hunzaikashif49/drug200.`

41.Bruce P, Bruce A, Gedeck P (2020) *Practical statistics for data scientists: 50+ essential concepts using R and Python*, USA: O'Reilly Media.

42.Houssein EH, Ibrahim IA, Mostafa A, et al. (2025) SMENN-hybrid: an efficient technique combining the synthetic minority oversampling technique with ensemble learning for diabetes prediction. *Sci Rep* 15: 43104. https://doi.org/10.1038/s41598-025-26583-z

43.Hairani H and Priyanto D (2023) A new approach of hybrid sampling SMOTE and ENN to the accuracy of machine learning methods on unbalanced diabetes disease data. *Int J Adv Comput Sci Appl* 14: 585–590. https://doi.org/10.14569/IJACSA.2023.0140864

44.Vairetti C, Assadi JL, Maldonado S (2024) Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification. *Expert Syst Appl* 246: 123149. https://doi.org/10.1016/j.eswa.2024.123149

45.Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor Newsl* 6: 20–29. https://doi.org/10.1145/1007730.1007735

46.Landherr A, Friedl B, Heidemann J (2010) A critical review of centrality measures in social networks. *Bus Inf Syst Eng* 2: 371–385. https://doi.org/10.1007/s12599-010-0127-3

47.Tymoshchuk D, Didych I, Maruschak P, et al. (2025) Machine learning approaches for classification of composite materials. *Modelling* 6: 118. https://doi.org/10.3390/modelling6040118

48.Li G and Sheng H (2025) A hybrid machine learning framework for predicting aircraft scaled sound pressure levels: a comparative study. *J Eng Appl Sci* 72: 163. https://doi.org/10.1186/s44147-025-00714-9

49. Mienye ID and Jere N (2024) A survey of decision trees: concepts, algorithms, and applications. *IEEE Access* 12: 86716–86727. https://doi.org/10.1109/ACCESS.2024.3416838

50. Sujon KM, Hassan R, Choi K, et al. (2025) Accuracy, precision, recall, F1-score, or MCC? Empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models. *J Big Data* 12: 268. https://doi.org/10.1186/s40537-025-01313-4

AIMS Press