Egyptian Informatics Journal 30 (2025) 100641

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

Clipper: An efficient cluster-based data pruning technique for biomedical data to increase the accuracy of machine learning model prediction

M.B. Karadeniz [a,*], Ebru Efeoğlu [b], Burak Çelik [a], Adem Kocyigit [a,c], Bahattin Türetken [a]

[a] Department of Electronics and Communication Engineering, Kocaeli University, Kocaeli, 58140, Turkiye
[b] Department of Software Engineering, Kütahya Dumlupınar University, Kütahya, 43020, Turkiye
[c] Department of Electronics and Automation, Bilecik Seyh Edebali University, Bilecik, 11000, Turkiye

ARTICLE INFO

Keywords:
Machine learning
Clustering
Biomedical data
Pruning

ABSTRACT

The exponential rise in clinical research costs can potentially be mitigated by half through the implementation of machine learning-driven efficient data processing techniques. Traditional methods like data preprocessing and hyperparameter tuning, which are effective for model optimization, often introduce complexities that can diminish the benefits of machine learning integration. To overcome this issue, we present Clipper: a novel, cluster-based data pruning approach designed specifically for biomedical data, aiming to enhance the predictive accuracy of machine learning models. Clipper's key advantage lies in its ability to automate the data pruning process, optimizing accuracy without the need for manual hyperparameter adjustments—a typically cumbersome aspect of machine learning tasks. Upon comprehensive comparative analysis, the proposed Clipper methodology demonstrates superior performance across various medical and biological datasets. Our experiments reveal Clipper's consistent superiority over baseline models, with significant accuracy improvements: 44% for Heart Disease, 7% for Breast Cancer, 40% for Parkinson's, and 20% for Raisin classification. Specifically, the model achieves remarkable predictive accuracy, with classification rates of 99.5% for Heart Disease, 99.64% for Breast Cancer, 99.47% for Parkinson's Disease, and 93% for Raisin Classification, thereby substantially outperforming contemporary state-of-the-art computational techniques. The empirical evidence suggests that Clipper serves as an effective accuracy enhancer for baseline models, eliminating the need for parameter tuning or complex preprocessing steps. Furthermore, Clipper produces robust outputs even at very low split rates, where baseline models typically perform poorly.

1. Introduction

Machine learning (ML) solutions for biomedical challenges are becoming increasingly effective [1]. As clinical research expenses rise exponentially due to laboratory time usage, efficient data processing techniques introduced by ML algorithms can potentially halve these costs [2,3]. Due to insufficient data, researchers are conducting new studies to increase the predictive accuracy of the ML-based model they are using. Data preprocessing, hyperparameter tuning, custom model or feature selection, and hybrid models are extensively studied to improve ML models for medical research [4,5]. While these methods are well-known for improving model fit, complex data structures and time-consuming implementation problems can offset the advantages of integrating ML [6].

Recent developments in the field of ML have increased the importance of more flexible and powerful methods for modeling imbalanced datasets. In this context, comprehensive analyses of multiple algorithms, including Decision Tree, AdaBoost, Bagging, Extra Random Trees, Gaussian Process Classifier, Ridge, Gaussian Naive Bayes, K-Nearest Neighbors, Multilayer Perceptron, and Support Vector Classifier, have been conducted [7–11]. Among them, nonlinear modeling approaches, such as Gaussian Process Regression and autoregressive neural networks, provide the potential for highly accurate predictions in complex dataset structures, such as agriculture and biomedical data [12,13].

Clustering, as an unsupervised classification method, is employed to categorize clinical data into smaller feature sets that exhibit similar properties. However, as dataset dimensions expand, conventional clustering frameworks often struggle to manage the requisite computational demands [14]. In response, novel clustering frameworks are being developed to harness parallel computing capabilities, thereby mitigating temporal costs [15,16]. A self-optimizing parameter setting approach has been developed to improve clustering performance in high-dimensional datasets [17]. Nevertheless, a significant limitation of

* Corresponding author.
E-mail address: karadeniz17@itu.edu.tr (M.B. Karadeniz).

https://doi.org/10.1016/j.eij.2025.100641
Received 12 October 2024; Received in revised form 27 December 2024; Accepted 5 March 2025
Available online 20 March 2025
1110-8665/© 2025 The Authors. Published by Elsevier B.V. on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

clustering persists: it does not consistently ensure optimal true positive classification accuracy [18].

Researchers have investigated various methods to improve classification performance, including hyperparameter tuning as a key approach to optimize model effectiveness. The effects of Quinlan's C4.5 algorithm and Classification and Regression Tree (CART) hyperparameter tuning on DT-based machine learning models have been investigated [19]. The study suggests that tuning a specific small subset of hyperparameters is a good alternative for achieving optimal predictive performance. In another study, grid search hyperparameter tuning combined with K-means clustering is proposed to enhance accuracy [20]. When applied to a customer churn dataset, this approach achieves significant accuracy improvement over benchmark models. However, hyperparameter tuning approaches are practical only under certain conditions, as a tuning strategy effective for one model may fail to work for another.

In order to alleviate the time-consuming hyperparameter tuning problem, effective data preprocessing techniques like data partitioning and data augmentation are studied [21]. Using data augmentation is expected to address the low precision problem in imbalanced datasets, such as those found in bank fraud cases and medical data [22]. However, generating synthetic data requires significant computational resources, and models can easily overfit when trained on augmented data. Data pruning is applied to mitigate overfitting issues [23]. By employing selective data pruning approaches, including pruning less informative or irrelevant data points, classification accuracy is improved along with reduced training time [24,25].

Hybrid pruning techniques that combine data pruning with custom feature selection are commonly applied to improve classification performance of complex multi-featured datasets. By selecting the most relevant features and data points, the prediction accuracy of models implementing ML algorithms is increased [26,27]. To address high-dimensional feature selection challenges, enhanced RIME (ERIME) algorithm, which integrates feature information entropy pruning and DBSCAN clustering, is proposed [28].

In contrast to the overfitting problem, the data pruning approach is vulnerable to worst-case scenarios, especially when dealing with sensitive data. Thus, several innovative approaches have been developed to improve model robustness. Selecting appropriate class pruning rates and applying random pruning within classes, called "fairness-aware" pruning approach, has been proposed for benchmark models, showing improvements in worst-class performance [29]. However, more intelligent pruning techniques are required that can analyze the interrelationships between data points and selectively remove redundant information while preserving sparse but valuable data.

The aforementioned techniques can reduce impurities and improve accuracy in some cases; however, they are often application- and algorithm-specific, limiting their global applicability. To address this challenge, we introduce Clipper, an efficient cluster-based data pruning technique designed to enhance the predictive accuracy of machine learning models in biomedical contexts. Clipper automates the data pruning process to achieve optimal accuracy, thereby alleviating the burden of manual hyperparameter adjustment typically associated with machine learning tasks. The underlying premise of Clipper is that unbalanced datasets, such as those frequently encountered in medical research, are susceptible to overfitting, which can lead to the inadvertent elimination of outlier groups. Clipper addresses this issue by judiciously pruning excessive data, thereby accentuating outlier groups and facilitating their accurate prediction. This enhanced ability to correctly identify outlier samples reduces false negatives, ultimately improving the overall model accuracy.

Clipper employs hierarchical clustering and automatically prunes predictable clusters to mitigate overfitting and enhance accuracy. The process begins with clustering the training features along with their annotated labels, facilitating the identification of outlier data structures within the training set. This approach diverges from conventional clustering models, which typically operate on unlabeled data [30]. Notably, Clipper repurposes clustering as a pruning tool rather than a classification method, treating labels as additional features in the training data for pruning purposes. During the hierarchical clustering process, the test data is segregated and preserved intact for subsequent evaluation.

Consider a dataset depicting customer characteristics, as illustrated in Fig. 1, which provides information about individuals' salaries and ages and their corresponding product purchase decisions [31]. This dataset comprises two features and 290 instances. To optimize the training data's efficacy, 25% of the data is reserved for testing purposes. Fig. 1(a) presents five hierarchically clustered customer groups. The inclined yellow line in Fig. 1(b) represents the trained classifier. It is hypothesized that customers whose features fall to the right of this line will purchase the product, while those to the left will not. The classifier's performance is evaluated using the reserved test data, as shown in Fig. 1(c). The model, without pruning, yielded 14 erroneous predictions out of 73 test instances, resulting in an accuracy of 80.8%. Notably, the model failed to accurately classify customers around 50 years of age in the lowest income bracket, who, contrary to expectations, did purchase the product.

Fig. 1(d) illustrates the pruning of Clusters 2 and 4 from the original dataset. This pruning addresses the potential bias introduced by the predominance of customers earning salaries in the 60-80k range. It is noteworthy that the test set, derived from the original dataset, remains unaltered throughout this process. Fig. 1(e) depicts the updated classifier boundary, fitted after training on the pruned customer data. As evidenced in Fig. 1(f), the pruned model yielded 9 incorrect predictions out of 73 test instances, achieving an accuracy of 87.7%. This represents a substantial 6.9% improvement in accuracy, primarily attributable to the enhanced capture of outlier groups. This concept expresses the cornerstone of the present study.

The primary contributions and novel aspects of this research are outlined as follows:

- We introduce Clipper, an innovative methodology designed to augment classification performance in machine learning models. Diverging from existing approaches, Clipper strategically repurposes clustering techniques as a pruning mechanism to mitigate overfitting and systematically enhance the predictive accuracy of machine learning algorithms.
- This research provides a comprehensive, heuristic-driven pruning approach for integrating Clipper into machine learning models with a standard baseline configuration, thereby facilitating a systematic approach to improving model prediction performance.
- Empirical evidence demonstrates that the Clipper automation acts as a accuracy booster across various model architectures. Notably, this approach potentially obviates the need for extensive and computationally intensive hyperparameter tuning, offering a significant methodological efficiency for machine learning developers.
- A distinctive contribution of Clipper is its capacity to rehabilitate baseline machine learning models that previously exhibited suboptimal predictive performance under constrained data partitioning scenarios, thereby expanding the resilience and adaptability of computational learning systems.

The rest of this paper is organized as follows: Section 2 elucidates the methodology and approach underlying Clipper. Section 3 details the experimental design and test environment. Section 4 presents a comparative evaluation of Clipper in relation to state-of-the-art architectures. Finally, Section 5 offers concluding remarks.

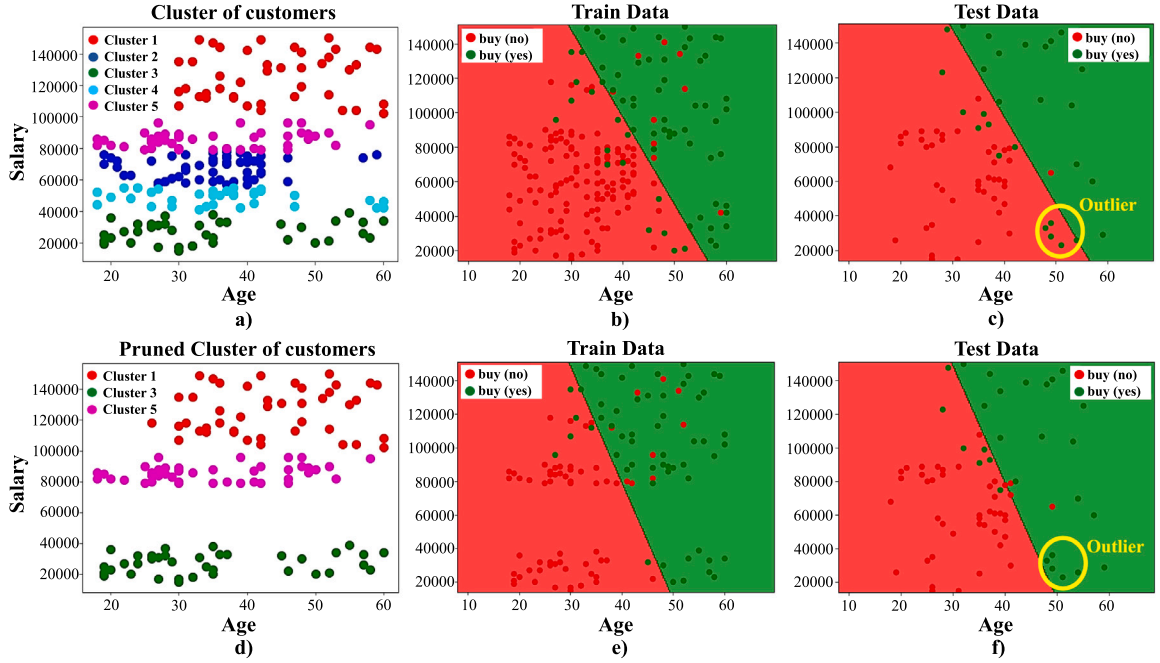


Fig. 1. Implementing customer characteristics dataset training with and without Clipper: (a) clustering of training data without Clipper, (b) train classifier without Clipper, (c) test data without Clipper, (d) implementing Clipper on training data, (e) train classifier with Clipper, (f) test data with Clipper.

2. Methodology and approach

Clipper utilizes a hierarchical clustering-based pruning approach to enhance the accuracy of machine learning models. Although the core principles of Clipper are broadly applicable to various classification algorithms, including Naive Bayes, Logistic Regression, and Decision Trees, this section specifically examines Logistic Regression to illustrate the methodology of the proposed model.

Logistic Regression is a probabilistic classification technique that estimates the likelihood of an instance belonging to a particular class by employing the logistic (sigmoid) function. Consider a categorical variable $y \in [0, 1]$ dependent on a set of independent variables $x \in \mathbb{R}$. The probability of $y = 1$ given x is denoted as $p(x)$. To constrain $p(x)$ within the range [0, 1], a sigmoid function is employed:

$$p(x) = \frac{1}{1 + e^{-\beta x}}, \tag{1}$$

where β represents the fitting parameter. The model creates a decision boundary by:

$$P(Y = 1|X) = p(\beta^T X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}, \tag{2}$$

where $P(Y = 1|X)$ is the probability of the positive class, β_0 is the intercept, $\beta_1, \ldots, \beta_n$ are model coefficients, and X is the input feature vector.

To determine the optimal fitting (β) for $y = f(x)$, the Maximum Likelihood Estimation is computed as:

$$\mathcal{L}(\beta) = \prod_{y_i=1} p(x_i) \times \prod_{y_i=0} 1 - p(x_i), \tag{3}$$

where x_i denotes the features to be classified as 1 or 0. Applying the natural logarithms to Eq. (3) yields the log-likelihood function as:

$$\ell(\beta) = \sum_{i=1}^{n} y_i ln(p(x_i)) + (1 - y_i) ln(1 - p(x_i)), \tag{4}$$

where n represents the number of samples in the dataset. Hierarchical clustering is then applied to n, using either single or complete linkage methods. The distance function D from point l to the cluster of points i, j is defined as:

$$D(C_l, (C_i, C_j)) = \overset{min}{max}(D(C_l, C_i), D(C_l, C_j)). \tag{5}$$

This approach partitions the training dataset into c clusters, where c can range up to n. Assume that within the clustered dataset, k represents the number of clusters with the least commonality, while $(n-k)$ denotes the number of clusters with similar D.

Considering the clustered x_i and revisiting Eq. (4), we obtain:

$$\begin{aligned}\ell(\beta) = &\sum_{i=1}^{k} Y_i ln(P(C_i)) + (1 - Y_i) ln(1 - P(C_i)) \\ &+ \sum_{i=k+1}^{n} Y_i ln(P(C_i)) + (1 - Y_i) ln(1 - P(C_i)),\end{aligned} \tag{6}$$

where C_i represents the i_{th} clustered features, and the capitalized Y and P denote the label and probability of the cluster, respectively. To maximize the logistic function, we solve:

$$\frac{\partial}{\partial \beta} = 0. \tag{7}$$

Applying partial derivative and chain rule, we obtain:

$$\begin{aligned}0 = &\sum_{i=1}^{k} \frac{Y_i}{P(C_i)} \frac{\partial P(C_i)}{\partial \beta} + \frac{(1 - Y_i)}{(1 - P(C_i))} \frac{\partial (1 - P(C_i))}{\partial \beta} \\ &+ \sum_{i=k+1}^{n} \frac{Y_i}{P(C_i)} \frac{\partial P(C_i)}{\partial \beta} \frac{(1 - Y_i)}{(1 - P(C_i))} \frac{\partial (1 - P(C_i))}{\partial \beta}.\end{aligned} \tag{8}$$

Since $P(C_i)$ is also fitted to the sigmoid curve with respect to β, the derivative of sigmoid function is defined as:

$$\frac{d}{d\beta} P(C_i) = P(C_i)(1 - P(C_i))C_i. \tag{9}$$

Combining Eqs. (9) and (8), we simplify to:

$$0 = \sum_{i=1}^{k} (Y_i - P(C_i))C_i + \sum_{i=k+1}^{n} (Y_i - P(C_i))C_i. \tag{10}$$

In unbalanced datasets, such as medical data, where $n \gg k$, the clustered regular data dominates the logistic curve, leading to:

$$0 = 1 - P(C_i). \tag{11}$$

The logistic curve becomes biased towards the regular clusters where $Y_i = 1$, shifting the curve rightward along the feature axis such that $P(C_i) \to 1$. As a result, the outlier groups are eradicated. To mitigate

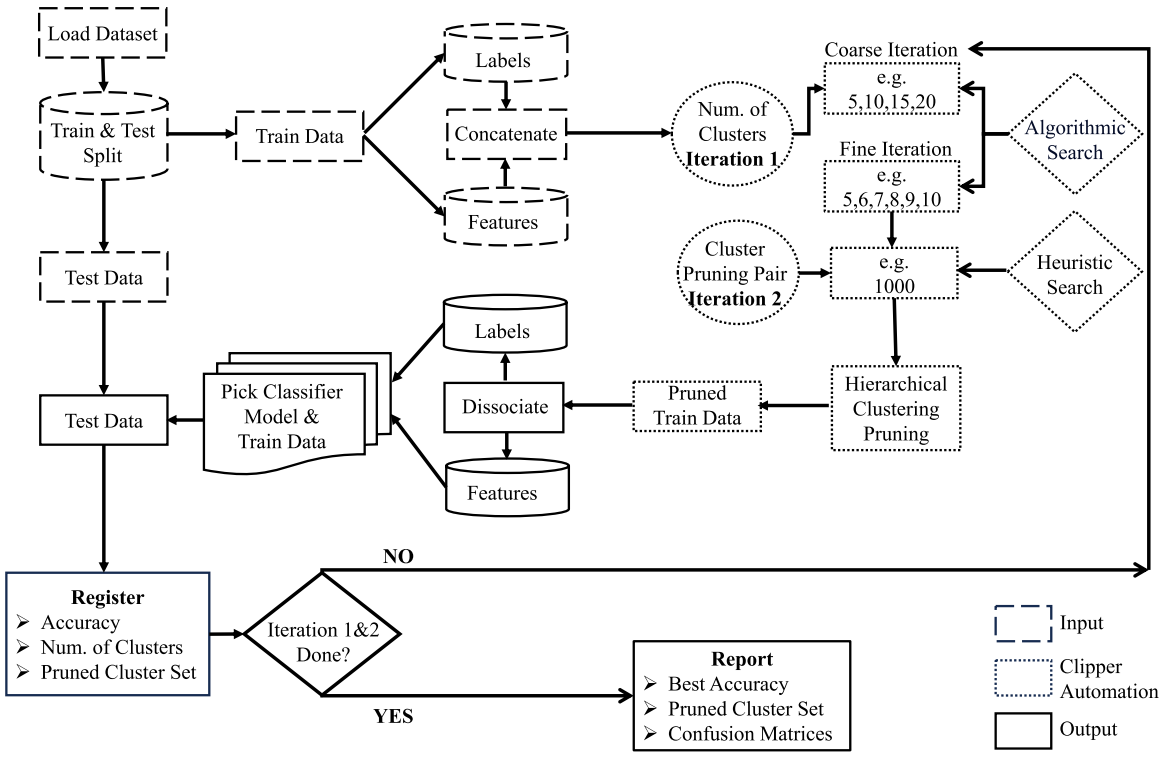


Fig. 2. Clipper Architecture: Input, Output, and Clipper Automation blocks are represented with distinct dashed styles.

this bias, pruning is applied to reduce $(n - k)$, thereby balancing the contributions of the clusters.

It is important to note that outlier groups with low probability values decrease the estimated $\ell(\beta)$ significantly, whereas regular clusters with high probability exert minimal influence. This can be illustrated by calculating the logistic curve: an outlier group with a probability of 37% has a 10× impact compared to a regular cluster with a probability of 90%, as $\ln(0.37) \approx 10\times\ln(0.9)$. This observation suggests that a modest reduction in the number of regular clusters is sufficient to achieve balanced contributions across all clusters. Under this pruning, Eq. (10) is resolved as follows:

$$0 = \frac{1}{2} - P(C_i). \tag{12}$$

Assuming a symmetrical distribution of outlier clusters, they are likely to fall into either the false negative or false positive groupings. Consequently, the logistic curve oscillates between $Y_i = 0$ and $Y_i = 1$, ultimately converging to the mean of the cluster labels as $P(C_i) \rightarrow \frac{1}{2}$. By improving the fitting of outlier groups, the accuracy enhancement achieved by Clipper can be quantified as:

$$Impv_{Clipper}(\%) = \frac{FN + FP}{2n} \tag{13}$$

where FN and FP represent the number of false negatives and false positives, respectively, before the application of Clipper to the baseline model.

3. Proposal and experimental design

In this section, Clipper architecture and design environment will be discussed.

3.1. Clipper architecture

The proposed Clipper architecture is illustrated in Fig. 2. Algorithm 1 provides a detailed, step-by-step breakdown of Clipper's implementation technique. Initially, the dataset is loaded and partitioned by the user (Algorithm 1, lines 1–3). The selection of the split rate is particularly crucial for datasets with fewer instances, where the amount of available data for training is already limited. The impact of the split rate will be elaborated upon in Section 4. The labels and features of the training data are concatenated to serve as input for Clipper Automation (Algorithm 1, line 5).

Clipper employs two search engines: the first engine identifies the optimal number of clusters (NOC), while the second engine captures the pruning pair to be extracted from the training data. The first engine iterates, $Iteration1$, both coarsely and finely to determine the optimal number of clusters (Algorithm 1, lines 9–37), while the second engine iterates, $Iteration2$, according to user-defined steps. In each iteration, the second engine employs a heuristic approach to determine the cluster pair for pruning, aiming to reduce processing time while maintaining Clipper's simplicity (Algorithm 1, lines 11–19).

Consider N clusters iterated by the first engine and cluster pairs iterated by the second engine. Using a conservative approach, the total number of iterations would be $\sum_{k=2}^{N} C_2^k$, where C represents the combination of all possible cluster pairs from the N clusters. It is noteworthy that clusters within a pair can be identical, resulting in the pruning of a single cluster. The complexity of this model is $O(N^3)$, which could lead to computational deadlock as N increases.

Clipper, however, employs logarithmic and heuristic approaches to mitigate this complexity. The first engine iterates over N using a coarse step of 5 (Algorithm 1, lines 21–27), while the second engine iterates a user-defined number of steps as $Iteration2$, e.g., 1000. The coarse step of 5 is intuitively chosen to account for the potential number of features a dataset might include.

At each coarse iteration step, Clipper randomly selects cluster pairs for pruning from the training data. Subsequently, labels and features are dissociated. The user selects a classifier model and trains it on the remaining training data. The untouched test data is then used to evaluate the fitted classifier. Test accuracy, the number of clusters, and pruned cluster pairs are recorded to identify the best accuracy across varying numbers of clusters (Algorithm 1, lines 15–18). Clipper then performs a fine iteration over the optimal number of clusters with 5

additional steps to search for further improvements (Algorithm 1, lines 28–32).

The implementation of a heuristic-logarithmic collaborative approach reduces the total number of iterations to $Iteration2\times((N/5)+5)$, effectively decreasing Clipper's computational complexity from $O(N^3)$ to $O(N)$. Consequently, Clipper's automated process achieves optimized cluster iterations while minimizing computational overhead. This optimization enables Clipper to achieve optimal results within seconds, demonstrating its agility and efficiency.

Algorithm 1 Clipper Technique

procedure CLIPPER(Dataset)
  Z(Data) ▷ Load Dataset
  X_train, X_test, y_train, y_test ← Z ▷ Split Dataset
  X_test, y_test ▷ Reserve Test Data
  Xy(X_train ⋃ y_train) ▷ Concatenate Labels and Features
  $NOC \leftarrow 5$ ▷ Number of Clusters
  $CIF \leftarrow 1$ ▷ Coarse Iteration Flag is set
  $FIF \leftarrow 0$ ▷ Fine Iteration Flag is reset
  while $Iteration1 > NOC$ **do**
    $HC(Xy, NOC)$ ▷ Hierarchical Clustering
    for $k \leftarrow 1$ to $Iteration2$ **do**
      $Cls1 \leftarrow c1$ ▷ $1 \leq c1 \leq NOC$
      $Cls2 \leftarrow c2$ ▷ $1 \leq c2 \leq NOC$
      $Xy \leftarrow (\bigcup Xy_i, \forall i \epsilon NOC)$ except $i\epsilon\{Cls1, Cls2\}$
      $X_train, y_train \leftarrow Xy$ ▷ Dissociate
      classifier_fit (X_train, y_train)
      classifier_predict (X_test, y_test)
      Register: ACC, NOC, Cls1, Cls2
    end for
    if CIF **then**
      $NOC \leftarrow NOC + 5$
      if $NOC \geq Iteration1$ **then**
        $NOC \leftarrow NOC(where(ACC == Best))$
        $Iteration1 \leftarrow NOC + 5$
        $CIF \leftarrow 0$ ▷ CIF is reset
        $FIF \leftarrow 1$ ▷ FIF raised
      end if
    else if FIF **then**
      $NOC \leftarrow NOC + 1$
      if $NOC \geq Iteration1$ **then**
        $FIF \leftarrow 0$ ▷ FIF is reset
      end if
    else
      Print('Best ACC, NOC, Cls1, Cls2')
      BREAK
    end if
  end while
end procedure

3.2. Databases

The datasets utilized in this study were obtained from the UC Irvine Machine Learning Repository, each identified by a unique ID. Four distinct datasets were evaluated: Heart Disease [32], Diagnostic Wisconsin Breast Cancer [33], Parkinson's [34], and Raisin [35]. These datasets exhibit varying class distributions: Heart Disease (55%-18%-12%-11%–4%, from absence to level 1–4 disease, respectively), Diagnostic Wisconsin Breast Cancer (benign:63% - malignant:37%), Parkinson's (healthy:25% - disease:75%), and Raisin ('Besni':50% - 'Kecimen':50%). This selection enables the assessment of Clipper's performance on both unbalanced datasets, such as those typically encountered in medical research, and highly balanced datasets, as exemplified by the Raisin dataset. Detailed information regarding the number of features and labels for each dataset is presented in Table 1.

Table 1
Characteristics of datasets from the UC Irvine Machine Learning Repository.

Dataset	ID	Num. of Features	Num. of Lables	Num. of Instances
Heart Disease	45	13	5	303
Wisconsin Breast Cancer	17	30	2	569
Parkinson's	174	22	2	195
Raisin	850	7	2	900

3.3. Classifier

To demonstrate Clipper's competitiveness, we evaluate a diverse array of classifiers from the machine learning library available in the scikit-learn library [31]. These include Naive Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), Kernel Support Vector Machine (Kernel SVM), Decision Tree (DT), K-Nearest Neighbors (KNN), and Random Forest (RF). To maintain an unbiased evaluation environment, we utilize the default configurations of these classifiers as provided by the library. This approach aims to isolate and reveal the effect of Clipper on ML model performance. Notably, Clipper can act as an accuracy booster for any model, regardless of whether it has undergone preprocessing or hyperparameter tuning.

3.4. Scaling and clustering

Before classifier fitting, StandardScaler is employed to normalize training data, and accordingly test data. For the clustering component, we focus on Agglomerative Clustering, a subset of Hierarchical Clustering known for its bottom-up approach. Each data point is initially treated as an individual cluster before it progressively merges to another data point to form clusters in this methodology. Such an approach renders outlier groups more distinguishable and facilitates their identification by Clipper. We utilize a default implementation of Agglomerative Clustering from [31], defined as *AgglomerativeClustering(n_clusters, metric, linkage)*, where *n_clusters* specifies the number of clusters to form. Clipper's first search engine iterates over this parameter in the algorithmic manner as described earlier.

4. Results and analysis

This section provides a comprehensive evaluation of the proposed model. The first part examines Clipper's performance compared to baseline, traditional ML models, highlighting the results with and without Clipper's application. The second part benchmarks Clipper versus state-of-the-art models from the literature.

4.1. Clipper vs. baseline models

Clipper is evaluated using four distinct datasets, i.e. Heart Disease, Breast Cancer, Parkinson's, Raisin, using seven different classifiers, i.e. NB, LR, SVM, Kernel SVM, DT, KNN, RF. To rigorously test the model's robustness, we employ varying split rates (test-to-train data sizes proportion) of 50%, 30%, 20%, 10%, and 5%. Performance metrics for Clipper and baseline models are presented in Table 2, with the best-reported metrics highlighted in red. Clipper's first engine iterates over a maximum of 50 clusters, while the second engine performs 1000 iterations.

The comparative analysis demonstrates that the Clipper method consistently achieves performance improvements over baseline models across various classifiers. When integrated with NB classifier, Clipper achieves consistent improvements ranging from 1% to 25%. A maximum accuracy improvement of 40% is observed at the 5% split rate. Moderate improvements are noted with LR classifier, particularly in the Parkinson's dataset, where accuracy increases by up to 25%. SVM and Kernel SVM classifiers, when used with Clipper, show accuracy

Table 2
Performance comparison between Clipper and baseline models evaluated on four datasets using seven classifiers under various test-train split ratios (5%–50%). Results for Clipper are shown with a maximum of 50 clusters and 1000 iterations per configuration.

Classifier	Split Rate	Heart Disease (ID=45)			Breast Cancer (ID= 17)			Parkinson's (ID=174)			Raisin (ID=850)		
	(Test/Train)	Baseline Accuracy %	Clipper Accuracy %	Accuracy Imprv. %	Baseline Accuracy %	Clipper Accuracy %	Accuracy Imprv. %	Baseline Accuracy %	Clipper Accuracy %	Accuracy Imprv. %	Baseline Accuracy %	Clipper Accuracy %	Accuracy Imprv. %
NB	0.5	48	60	12	93	95	2	65	73	8	84	86	2
	0.3	37	56	19	91	94	3	68	78	10	84	85	1
	0.2	39	57	18	90	94	4	69	79	10	86	88	2
	0.1	48	65	17	89	95	6	60	85	25	83	89	6
	0.05	50	56	6	86	93	7	40	80	40	76	80	4
LR	0.5	57	63	6	98	98	0	87	90	3	86	87	1
	0.3	58	65	7	98	99	1	88	93	5	86	87	1
	0.2	61	67	6	96	98	2	79	92	13	89	91	2
	0.1	55	61	6	100	100	0	70	95	25	90	91	1
	0.05	44	56	12	100	100	0	70	90	20	84	87	3
SVM	0.5	53	61	8	96	98	2	85	91	6	86	88	2
	0.3	56	65	9	96	98	2	83	93	10	86	87	1
	0.2	55	66	11	98	99	1	77	92	15	91	92	1
	0.1	52	65	13	100	100	0	65	95	30	90	91	1
	0.05	44	56	12	100	100	0	60	90	30	84	87	3
Kernel SVM	0.5	55	60	5	97	98	1	87	90	3	86	88	2
	0.3	55	63	8	98	98	0	93	93	0	86	88	2
	0.2	59	67	8	98	99	1	92	92	0	90	92	2
	0.1	58	68	10	100	100	0	95	95	0	88	90	2
	0.05	50	63	13	97	100	3	90	90	0	82	87	5
DT	0.5	47	56	9	94	95	1	81	95	14	81	85	4
	0.3	49	59	10	94	97	3	90	97	7	82	86	4
	0.2	54	69	15	93	99	6	85	95	10	81	90	9
	0.1	45	65	20	91	98	7	65	100	35	83	90	7
	0.05	25	69	44	97	100	3	80	100	20	76	91	15
KNN	0.5	57	61	4	95	97	2	93	94	1	84	87	3
	0.3	63	67	4	96	98	2	92	97	5	83	87	4
	0.2	67	72	5	96	97	1	92	97	5	88	89	1
	0.1	71	77	6	98	100	2	95	95	0	86	90	4
	0.05	50	75	25	100	100	0	90	100	10	82	89	7
RF	0.5	51	61	10	91	97	6	80	95	15	83	87	4
	0.3	54	64	10	95	98	3	85	98	13	84	88	4
	0.2	61	72	11	92	99	7	90	100	10	87	91	4
	0.1	45	74	29	95	100	5	90	100	10	86	93	7
	0.05	50	69	19	93	100	7	100	100	0	73	93	20

improvements of up to 30% and 13%, respectively, over baseline models. KNN classifier combined with Clipper also demonstrates the method's effectiveness, with an accuracy improvement of 25% in the Breast Cancer dataset. Consistent and substantial improvements are obtained across this dataset.

The most dramatic improvements are observed with DT and RF classifiers integrated with Clipper. A maximum accuracy improvement of 44% at the 5% split rate is achieved using the DT classifier in the Breast Cancer dataset, where the baseline model collapses. RF classifier with Clipper shows significant accuracy improvements of 29% and 20% in the Breast Cancer and Raisin datasets, respectively, particularly at split rates where baseline models exhibit poor predictive performance.

These findings reveal that the proposed method shows consistent and significant improvements across various ML classifiers and diverse datasets. The biggest improvements occur at lower split rates (5%–10%), suggesting Clipper's capability in handling limited training data. This analysis highlights Clipper's potential as a generalized enhancement technique with broad applicability in ML.

4.2. Clipper vs. state-of-the-art models

In this section, we present a comparative analysis of the proposed model, Clipper, versus state-of-the-art models. A detailed analysis is provided of sensitivity, specificity, F-scores, and precision metrics. The performance metrics for Clipper are derived without any parameter tuning to ensure a fair comparison.

Considering the constraints of limited medical data and the potential for significant class imbalance, we employed Stratified K-Fold Cross-Validation (*StratifiedKFold(n_splits =10, shuffle=True, random_state=1*), for performance assessment. This methodological approach ensures an adequate representation of each class in the training data, thereby enhancing the robustness and reliability of the evaluation.

4.2.1. Heart disease dataset

Heart Disease datasets from two sources are utilized: [32] alone and a combination of [32,36]. The dataset from [32] comprises 303 instances, while the combined dataset from both sources contains 573 instances. Both datasets share 13 common features representing various symptoms.

It is important to note the difference in classification schemes between the two sources. In [32], patients are categorized into five levels: no heart disease (level 0) or varying degrees of heart disease (levels 1–4). Conversely, [36] employs a binary classification system, indicating either the presence or absence of heart disease. To harmonize these datasets for combined analysis, the data from [32] are recoded, assigning 0 to patients having no disease and 1 to patients having disease levels ranging from 1 to 4.

Table 3 demonstrates the superior performance of our proposed method, Clipper, over the existing state-of-the-art models. When comparing Clipper and [37] in the binary classification scenario with 573 instances, both NB and RF classifiers integrated with Clipper exhibit a 3.8% increase in accuracy. Notably, DT classifier with the proposed method achieves near-perfect accuracy of 99.1%, with 100% precision and specificity, outperforming [37] in every metric.

Table 3
Performance metrics comparison between Clipper and state-of-the-art models on Heart Disease dataset.

Evaluation Criteria	NB [37]	SVM [37]	RF [37]	NB (Clipper)	DT (Clipper)	RF (Clipper)	SVM [38]	SVM [38]	RF (Clipper)	SVM (Clipper)
Num. of classes[a]	2	2	2	2	2	2	2	5	2	5
Num. of objects[b]	573	573	573	573	573	573	303	303	303	303
Accuracy (%)	86.4	97.5	95.7	90.2	99.1	99.5	84.8	60.4	96.8	67.7
Sensitivity (%)	86.4	97.5	95.8	87.1	98	99.6	N/A	N/A	91.7	N/A
Specificity (%)	82.8	94.9	92.6	92.7	100	99.4	N/A	N/A	100	N/A
Precision (%)	86.4	95.9	95.9	90.7	100	99.3	N/A	N/A	100	N/A
F-score	86.4	96.7	95.7	88.8	99	99.6	N/A	N/A	95.7	N/A

[a] The number of classes to be predicted by the model.

[b] The number of instances used by the model.

Training with a smaller dataset of 303 instances, Clipper achieves 96.8% accuracy, along with 100% specificity and precision, which represents an improvement of at least 12% higher than the performance of [38]. Particularly compelling is Clipper's performance in multi-class classification challenges. In the 5-class SVM scenario, accuracy improved from 60.4% to 67.7%, highlighting Clipper's potential to enhance classification performance in complex predictive tasks.

4.2.2. Breast cancer wisconsin (diagnostic) dataset

The Breast Cancer Wisconsin (Diagnostic) dataset [33] has been widely used for breast cancer detection. Numerous studies have explored various machine learning algorithms to enhance classification performance. For instance, a maximum accuracy of 99.12% is reported using LR among eight evaluated algorithms [7]. In another study, [39] achieves a peak accuracy of 96.46% using a Neural Network algorithm while assessing 11 different machine learning approaches to classify breast cancer as benign or malignant.

A shallow Artificial Neural Network (ANN) model with a single hidden layer, devoid of feature optimization or selection algorithms, is proposed in [40]. This model delivers remarkable results, achieving 99.47% average accuracy, 99.53% specificity, 99.59% sensitivity, 98.71% precision, and a 99.13% F-score. To further enhance accuracy, [41] utilizes ensemble classifiers and the Recurrent Feature Elimination (RFE) technique, achieving an accuracy rate of 99.02%.

Other notable approaches include the optimization of KNN algorithm with effective k-nearest values and distance functions, as well as applying soft computing algorithms, such as Emperor Penguin Optimization (EPO), Gravitational Search Optimization Algorithm (GSA), and a proposed hybrid approach of GSA and EPO (hGSAEPO) algorithm as presented in [42]. This method reports 98.31% accuracy, 98% precision, 97% sensitivity, 98.87% specificity, and a 95.39% F1 score. Additionally, an innovative approach introduced in [43] combines kernel-based Principal Component Analysis (K-PCA) with NB algorithm and Chi-square-based feature selection, achieving an impressive accuracy of 99.28%.

The reference metrics and Clipper's performance on the Breast Cancer Wisconsin (Diagnostic) dataset are summarized in Table 4. Our proposed method achieves prediction accuracy rates ranging from 98.04% to 99.64%, precision values from 98.61% to 100%, sensitivity values from 96.19% to 99.05%, and specificity values from 99.14% to 100%. Compared to reported reference metrics, such as LR [7] with 99.12% accuracy, ANN model [40] with 99.47%, the hybrid approach [41] with 99.02%, hGSAEPO algorithm [42], and Kernel-PCA method [43] with 99.28%, Clipper demonstrates competitive and, in some cases, slightly superior prediction accuracy. Notably, employing RF and LR classifiers, Clipper achieves accuracy rates of 99.64% and 99.10%, respectively.

Clipper demonstrates superior accuracy compared to state-of-the-art models without requiring hyperparameter tuning. While the referenced studies introduces innovative approaches, such as ensemble methods, feature selection techniques, and advanced neural network architectures, Clipper reinforces the potential of both traditional and advanced machine learning algorithms in medical diagnostics.

Table 4
Performance metrics comparison between Clipper and state-of-the-art models on Breast Cancer Wisconsin (Diagnostic) Dataset.

Classifier	Accuracy (%)	Precision (%)	Sensitivity (%)	F-Score (%)	Specificity (%)
LR [7]	99.12	–	97.73	–	100.00
ANN [40]	99.47	98.71	99.59	99.13	99.53
XGB+RFE [41]	99.02	99.00	99.00	99.00	–
hGSAEPO [42]	98.31	98.00	97.00	95.39	98.87
K-PCA [43]	99.28	–	100.00	99.46	97.87
LR (Clipper)	99.10	100.00	97.62	98.77	100.00
SVM (Clipper)	98.93	98.64	98.57	98.57	99.14
DT (Clipper)	98.04	98.61	96.19	97.34	99.14
KNN (Clipper)	98.93	99.07	98.10	98.56	99.43
RF (Clipper)	99.64	100.00	99.05	99.51	100.00

4.2.3. Parkinson's disease dataset

Various machine learning and deep learning models, including SVM, RF, DT, KNN, and Multilayer Perceptron (MLP), have been employed for Parkinson's Disease detection. To enhance model performance, techniques such as the Synthetic Minority Oversampling Technique (SMOTE), custom feature selection, and hyperparameter tuning (GridSearchCV) have been utilized in the literature. Notable results are achieved using MLP and SVM with a 70:30 training/testing split, incorporating SMOTE and GridSearchCV [9]. Their findings indicate that MLP achieves 98.31% accuracy, 98% recall, 100 sensitivity, and 99% F-score, while SVM attains 95% accuracy, 96% recall, 98% sensitivity, and 97% F-score.

In [44], the impact of kernel function (KF) selection in SVM on Parkinson's Disease detection is investigated. Using a fast correlation-based filter (FCBF), SVM achieves 86.15% accuracy, 93.87% sensitivity, and 62.5% specificity. Further research employs strategies such as custom feature selection and hyperparameter tuning via RandomizedSearchCV to identify salient features. Feed-forward Neural Network and Kernel SVM models, utilizing an 80:20 dataset split, achieve remarkable results, including 99.11% overall accuracy, 98.78% recall, 99.96 precision, and 99.23% F-score [45].

An innovative approach proposed in [46] involves a neural network model for generating speech signals, incorporating a pre-loss reduction module with pre-control for data preparation. This study introduces a novel multi-agent salp swarm (MASS) cleaner for feature derivation and a Parkinson illumination neural network (PCNN). The MASS-PCNN

Table 5
Performance metrics comparison between Clipper and state-of-the-art models on Parkinson's Disease dataset.

Classifier	Accuracy (%)	Precision (%)	Sensitivity (%)	F-Score (%)	Specificity (%)
MLP [9]	98.31	–	100.00	99.00	–
SVM [9]	95.00	–	98.00	97.00	–
FCBF [44]	86.15	–	93.87	–	62.50
Kernel SVM[45]	99.11	99.96	98.78	99.23	–
MASS PCNN[46]	99.10	97.80	94.70	99.50	–
LR (Clipper)	91.58	90.71	99.29	94.70	69.00
SVM (Clipper)	94.74	93.58	100.00	96.62	80.00
DT (Clipper)	99.47	99.33	100.00	99.66	98.00
KNN (Clipper)	98.42	98.71	99.29	98.96	95.50
RF (Clipper)	99.47	99.33	100.00	99.66	98.00

model demonstrates 99.1% accuracy, 97.8% precision, 94.7% recall, and a 99.5% F-score, outperforming existing models.

The reference metrics, along with Clipper's performance on the Parkinson's Disease dataset, are summarized in Table 5. Clipper demonstrates significant strengths, particularly with the DT and RF classifiers. These two methods by integrating Clipper achieve 99.47% accuracy, 99.33% precision, 100% sensitivity, 99.66% F-score, and 98% specificity, surpassing nearly all referenced studies in every metric. The Kernel SVM [45] with a precision of 99.96%, slightly outperforms Clipper's methods in that regard.

However, the analysis also reveals variability across different classifiers within the Clipper approach. While DT and RF classifiers excel, the LR classifier shows comparatively lower performance, with 91.58% accuracy and 69% specificity. SVM and KNN classifiers of Clipper yield more moderate results, with accuracies of 94.74% and 98.42%, respectively. This variability highlights the importance of classifier selection, suggesting that no single classifier is universally optimal for Parkinson's Disease detection. Another key strength of Clipper is its consistently high sensitivity across classifiers, with most methods achieving 99%–100% sensitivity. This is particularly essential in medical diagnostics, where identifying potential disease cases is essential.

4.2.4. Raisin dataset

The Raisin dataset, comprising 900 instances, is utilized for binary classification between two raisin varieties: 'Kecimen' and 'Besni'. Numerous studies have sought to enhance classification accuracy on this dataset. In [47], an accuracy of 87.67% is reported using feature selection with the Genetic Algorithm (GA) method to optimize the Support Vector Classifier (SVC) algorithm. In [48], a hybrid approach is explored, combining five machine learning methods (KNN, Ridge Classifier, XGBoost, SVC, and LDA) with Convolutional Neural Networks (CNN) to improve accuracy.

LR, DT, and ANN methods, employing PCA for feature reduction, are investigated in [10]. Among these, the ANN method achieves the highest accuracy of 88.9%. A comprehensive evaluation of multiple classifiers is conducted in [11]. Among the evaluated classifiers, LightGBM exhibits 98.4% accuracy, followed by the other classifiers as AdaBoost (90.3%), SVM (87.78%), KNN (87.04%), DT and RF (both 86.3%), GNB (84.81%), and XGB (83.7%). AdaBoost and LightGBM demonstrates superior performance with the adaptation of caret, H2O, neuralnet, and keras packages.

Backward Elimination algorithm for feature selection, followed by classification using NB, KNN, and SVM, is employed in [49]. The SVM algorithm achieves the highest accuracy of 87.22%. In [50], XGB, SVM, MLP, and LR are examined, yielding accuracies of 85.9%, 91%, 87.3%, and 86.7%, respectively. A specialized feature selection with RF algorithm is proposed in [51], achieving an accuracy of 94.07% with the MLP model.

The comparative evaluation of Clipper's performance against existing methodologies is demonstrated in Table 6. Most notably, Clipper achieves leading accuracy in five out of seven classifiers, with particularly impressive results in RF (93.0%), XGB (91.3%), and DT (90.0%). These results consistently surpassed previous implementations by significant margins, with improvements ranging from 3.7 to 10.1 percentage points over existing methods.

A distinguishing characteristic of Clipper is its remarkable consistency across different classification paradigms, maintaining an accuracy above 89% for all implemented algorithms. This consistency spans from simpler models like LR to more complex ensemble methods, suggesting robust feature engineering and preprocessing capabilities. Even in cases where Clipper does not achieve the highest accuracy, it still demonstrates competitive performance while outperforming all other referenced works.

It is noteworthy that, while MLP with optimized feature selection and LightGBM demonstrate marginally better classification accuracy compared to the standard, unoptimized Clipper-based machine learning models, these specialized techniques fall outside the scope of the current study. The results underscore Clipper's efficacy in raisin classification tasks, even without extensive feature engineering or model optimization.

5. Conclusions and future work

The principal advantage of Clipper lies in its automated approach to data pruning, which significantly reduces the necessity for manual hyperparameter tuning and offers a streamlined solution for managing imbalanced datasets. In contrast to conventional methodologies that necessitate extensive computational resources for preprocessing or synthetic data generation, Clipper's clustering-based approach enhances the model robustness while concurrently optimizing the search space complexity. The robustness of the proposed model is particularly notable under conditions of low data split rates, where baseline models often exhibit suboptimal performance. Empirical evaluations of Clipper demonstrates substantial improvements in predictive accuracy across a diverse range of biomedical datasets, including those related to Heart Disease, Breast Cancer, Parkinson's Disease, and Raisin classification. The observed maximum accuracy increases of 44%, 7%, 40%, and 20% on these respective datasets underscore Clipper's efficacy as a universal accuracy boosting tool.

While Clipper demonstrates remarkable effectiveness, opportunities for further enhancement merit consideration. The current implementation has primarily validated its performance on medium-scale datasets, particularly excelling in domains with constrained data availability such as biomedical and agricultural applications. To establish Clipper's broader applicability across diverse domains, exploring its performance with larger-scale datasets would be valuable. Although Clipper's heuristic approaches effectively manage computational demands, optimization opportunities exist for scenarios involving substantially larger datasets, particularly in resource-constrained environments.

Clipper's distinctive strength in preserving and accurately fitting outlier data points has proven particularly advantageous for addressing class imbalances. This characteristic enhances model performance specifically in scenarios where traditional approaches struggle with

Table 6
Comparative analysis of classification accuracy between Clipper and state-of-the-art models evaluated on the Raisin dataset.

Classifier	Clipper (%)	[47] (%)	[48] (%)	[10] (%)	[11] (%)	[50] (%)	[51] (%)
LR	89.1	–	–	88.3	–	86.7	–
SVM	90.3	87.7	87.5	–	87.8	91.0	93.7
DT	90.0	85.1	–	80.4	86.3	–	82.9
KNN	90.3	85.1	85.9	–	87.0	–	88.5
RF	93.0	85.6	–	–	86.3	–	82.9
MLP	92.8	86.7	–	88.9	–	87.3	94.1
XGB	91.3	–	85.8	–	83.7	85.9	–

boundary definition around outlier groups. The magnitude of improvement exhibits a positive correlation with dataset heterogeneity—more diverse data structures tend to yield more substantial accuracy gains. For highly uniform datasets, however, careful consideration must be given to the trade-off between marginal accuracy improvements and computational costs.

Future work could explore several enhancements to the Clipper algorithm:

5.1. Scalability and computational efficiency enhancement

To enhance Clipper's capabilities, the implementation of parallel computing methodologies is proposed, particularly for analyzing large-scale datasets. Research indicates that parallel clustering algorithms significantly improve efficiency and reduce computational time when applied to big data [52,53]. Additionally, integrating both homogeneous and heterogeneous parallel clustering techniques could further bolster Clipper's performance in terms of processing speed and scalability. Such advancements have consistently demonstrated their ability to markedly enhance efficiency and minimize processing times in big data applications.

5.2. Dynamic pruning methodologies

Expanding Clipper's adaptive pruning to dynamically adjust cluster counts based on dataset characteristics could further refine model accuracy. Dynamic pruning approach would enhance Clipper's versatility, making it effective across more diverse datasets.

5.3. Application to diverse biomedical and clinical datasets

Clipper's effectiveness in biomedical applications highlights its potential for clinical decision support systems, where accurate predictions are critical. Future studies could focus on testing Clipper across different medical conditions, patient demographics, and data sources, which would broaden its applicability and enhance its performance in clinical settings.

5.4. Integration with ensemble learning techniques

The integration of Clipper with ensemble methodologies, such as boosting or bagging techniques, presents an opportunity for further enhancement of model performance. This synergistic approach would enable Clipper to leverage the collective strengths of multiple classifiers, potentially yielding superior accuracy and robustness, particularly when applied to highly complex datasets.

In conclusion, Clipper represents a significant advancement in machine learning-driven data processing within the biomedical domain. By optimizing the data pruning process, it provides a practical and effective tool for improving the accuracy of diverse machine learning models, thereby facilitating more efficient and precise healthcare solutions. Continued research and iterative enhancements could further expand Clipper's capabilities, solidifying its position as an adaptive tool for machine learning applications.

CRediT authorship contribution statement

M.B. Karadeniz: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ebru Efeoğlu:** Writing – original draft, Validation, Methodology, Investigation. **Burak Çelik:** Writing – review & editing, Investigation, Formal analysis, Data curation, Conceptualization. **Adem Kocyigit:** Writing – review & editing, Formal analysis, Conceptualization. **Bahattin Türetken:** Writing – review & editing, Visualization, Supervision, Project administration, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Jovel Juan, Greiner Russ. An introduction to machine learning approaches for biomedical research. Front Med 2021;8. http://dx.doi.org/10.3389/fmed.2021.771607.

[2] de Vos Juliette, Visser Laurenske A, de Beer Aletta A, Fornasa Mattia, Thoral Patrick J, Elbers Paul WG, et al. The potential cost-effectiveness of a machine learning tool that can prevent untimely intensive care unit discharge. Value Heal 2022;25(3):359–67. http://dx.doi.org/10.1016/j.jval.2021.06.018, URL https://www.sciencedirect.com/science/article/pii/S1098301521017423.

[3] Miao Rujia, Dong Qian, Liu Xuelian, Chen Yingying, Wang Jiangang, Chen Jianwen. A cost-effective, machine learning-driven approach for screening arterial functional aging in a large-scale Chinese population. Front Public Heal 2024;12. http://dx.doi.org/10.3389/fpubh.2024.1365479, URL https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2024.1365479.

[4] Sidey-Gibbons Jenni AM, Sidey-Gibbons Chris J. Machine learning in medicine: a practical introduction. BMC Med Res Methodol 2019;19. URL https://api.semanticscholar.org/CorpusID:84183143.

[5] Ayatollahi Haleh, Gholamhosseini Leila, Salehi Masoud. Predicting coronary artery disease: A comparison between two data mining algorithms. BMC Public Health 2019;19. http://dx.doi.org/10.1186/s12889-019-6721-5.

[6] V a Binson, Thomas Sania, Monikavasagom Subramoniam, Jayaseelan Arun, Subbaiyan Naveen, Madhu S. A review of machine learning algorithms for biomedical applications. Ann Biomed Eng 2024;52. http://dx.doi.org/10.1007/s10439-024-03459-3.

[7] Hossin Md, Shamrat F, Bhuiyan Md Rifat, Hira Rabea, Khan Tamim, Molla Shourav. Breast cancer detection: an effective comparison of different machine learning algorithms on the wisconsin dataset. Bull Electr Eng Informatics 2023;12:2446–56. http://dx.doi.org/10.11591/beei.v12i4.4448.

[8] Kadhim Rania, Kamil Mohammed. Comparison of breast cancer classification models on wisconsin dataset. Int J Reconfigurable Embed Syst (IJRES) 2022;11:166–74. http://dx.doi.org/10.11591/ijres.v11.i2.pp166-174.

[9] Alshammri Raya, Alharbi Ghaida, Alharbi Ebtisam, Almubark Ibrahim. Machine learning approaches to identify parkinson's disease using voice signal features. Front Artif Intell 2023;6. http://dx.doi.org/10.3389/frai.2023.1084001, URL https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2023.1084001.

[10] Tigga Onima, Pal Jaya, Mustafi Debjani. Performance analysis of machine learning algorithms for data classification. AIP Conf Proc 2024;3164(1):030001. http://dx.doi.org/10.1063/5.0214183, arXiv:https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/5.0214183/19968223/030001_1_5.0214183.pdf.

[11] Raihen Md Nurul, Akter Sultana. Prediction modeling using deep learning for the classification of grape-type dried fruits. Int J Math Comput Eng 2024;2(1):1–12. http://dx.doi.org/10.2478/ijmce-2024-0001.

[12] Jin Bingzi, Xu Xiaojie. Price forecasting through neural networks for crude oil, heating oil, and natural gas. Measurement: Energy 2024;1:100001. http://dx.doi.org/10.1016/j.meaene.2024.100001, URL https://www.sciencedirect.com/science/article/pii/S2950345024000010.

[13] Jin Bingzi, Xu Xiaojie. Forecasting wholesale prices of yellow corn through the Gaussian process regression. Neural Comput Appl 2024;36. http://dx.doi.org/10.1007/s00521-024-09531-2.

[14] Ros Frédéric, Guillaume Serge. A progressive sampling framework for clustering. Neurocomputing 2021;450:48–60. http://dx.doi.org/10.1016/j.neucom.2021.04.029, URL https://www.sciencedirect.com/science/article/pii/S0925231221005567.

[15] Bikku Thulasi, Nandam Sambasiva Rao, Akepogu Ananda Rao. A contemporary feature selection and classification framework for imbalanced biomedical datasets. Egypt Informat J 2018;19(3):191–8. http://dx.doi.org/10.1016/j.eij.2018.03.003, URL https://www.sciencedirect.com/science/article/pii/S1110866517301081.

[16] Jayapradha J, Muttashar Abdulsahib Ghaida, Ibrahim Khalaf Osamah, Prakash M, Uddin Mueen, Abdelhaq Maha, et al. Cluster-based anonymity model and algorithm for 1:1 dataset with a single sensitive attribute using machine learning technique. Egypt Informat J 2024;27:100485. http://dx.doi.org/10.1016/j.eij.2024.100485, URL https://www.sciencedirect.com/science/article/pii/S1110866524000483.

[17] Wang Zhengru, Wang Xin, Zhang Shuhao. Mostream: A modular and self-optimizing data stream clustering algorithm. 2024, URL https://arxiv.org/abs/2309.04799.

[18] Rojček Michal. System for fuzzy document clustering and fast fuzzy classification. 2015, p. 39–42. http://dx.doi.org/10.1109/CINTI.2014.7028711.

[19] Mantovani Rafael, Horvath Tomas, Rossi André, Cerri Ricardo, Barbon Junior Sylvio, Vanschoren Joaquin, et al. Better trees: an empirical study on hyperparameter tuning of classification decision tree induction algorithms. Data Min Knowl Discov 2024. http://dx.doi.org/10.1007/s10618-024-01002-5.

[20] Kumar Shivam, Singh Tushar, Singh Smita, Singh Shivam. Grid search hyperparameter tuning and K-means clustering toimprove the decision tree accuracy. Int J Innov Sci Res Technol 2022;7(9). http://dx.doi.org/10.5281/zenodo.7121626.

[21] Korjus Kristjan, Hebart Martin, Vicente Raul. An efficient data partitioning to improve classification performance while keeping parameters interpretable. PLOS ONE 2016;11:e0161788. http://dx.doi.org/10.1371/journal.pone.0161788.

[22] dos Santos Tanaka Fabio Henrique Kiyoiti, Aranha Claus. Data augmentation using GANs. 2019, URL https://arxiv.org/abs/1904.09135.

[23] Stamate Daniel, Alghamdi Wajdi, Ståhl Daniel, Logofătu Doina, Zamyatin Alexander V. PIDT: A novel decision tree algorithm based on parameterised impurities and statistical pruning approaches. In: Artificial intelligence applications and innovations. 2018, URL https://api.semanticscholar.org/CorpusID:46896638.

[24] Saadatfar Hamid, Khosravi Samiyeh, Joloudari Javad Hassannataj, Mosavi Amir, Shamshirband Shahaboddin. A new K-nearest neighbors classifier for big data based on efficient data pruning. Mathematics 2020;8(2). http://dx.doi.org/10.3390/math8020286, URL https://www.mdpi.com/2227-7390/8/2/286.

[25] Yang Shuo, Xie Zeke, Peng Hanyu, Xu Min, Sun Mingming, Li Ping. Dataset pruning: Reducing training data by examining generalization influence. 2023, URL https://arxiv.org/abs/2205.09329.

[26] Chen Weihong, Xu Yuhong, Yu Zhiwen, Cao Wenming, Chen C, Han Guoqiang. Hybrid dimensionality reduction forest with pruning for high-dimensional data classification. IEEE Access 2020;PP:1. http://dx.doi.org/10.1109/ACCESS.2020.2975905.

[27] yuan Chen Yuan, bin Wang Zhi. Feature selection based convolutional neural network pruning and its application in calibration modeling for NIR spectroscopy. Chemometr Intell Lab Syst 2019;191:103–8. http://dx.doi.org/10.1016/j.chemolab.2019.06.004, URL https://www.sciencedirect.com/science/article/pii/S0169743919302126.

[28] Wu Huangying, Chen Yi, Zhu Wei, Cai Zhen-Nao, Heidari Ali Asghar, Chen Huiling. Feature selection in high-dimensional data: an enhanced rime optimization with information entropy pruning and DBSCAN clustering. Int J Mach Learn Cybern 2024;15:1–44. http://dx.doi.org/10.1007/s13042-024-02143-1.

[29] Vysogorets Artem, Ahuja Kartik, Kempe Julia. DRoP: Distributionally robust pruning. 2024, URL https://arxiv.org/abs/2404.05579.

[30] Xu Rui, Wunsch D. Survey of clustering algorithms. IEEE Trans Neural Netw 2005;16(3):645–78. http://dx.doi.org/10.1109/TNN.2005.845141.

[31] Akram. URL https://www.kaggle.com/datasets/akram24/social-network-ads.

[32] Janosi Andras, Steinbrunn William, Pfisterer Matthias, Detrano Robert. Heart Disease. 1989, http://dx.doi.org/10.24432/C52P4X, UCI Machine Learning Repository.

[33] Wolberg William, Mangasarian Olvi, Street Nick, Street W. Breast cancer wisconsin (diagnostic). 1993, http://dx.doi.org/10.24432/C5DW2B, UCI Machine Learning Repository.

[34] Little Max. Parkinsons. 2007, http://dx.doi.org/10.24432/C59C74, UCI Machine Learning Repository.

[35] Cinar Ilkay, Koklu Murat, Tasdemir Sakir. Raisin. 2020, http://dx.doi.org/10.24432/C5660T, UCI Machine Learning Repository.

[36] Statlog (Heart), UCI Machine Learning Repository, http://dx.doi.org/10.24432/C57303.

[37] Nashif S, Raihan MdRakib Hossain, Islam Md Rasedul, Imam Mohammad Hasan. Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system. World J Eng Technol 2018;06:854–73, URL https://api.semanticscholar.org/CorpusID:69438529.

[38] Huang Yin-Fu. Mobile health-monitoring system with inference, fall detection, and cardiovascular prediction. 2018, p. 7284–96. http://dx.doi.org/10.24297/ijct.v17i2.7621.

[39] Benbrahim Houssam, Hachimi Hanaa, Amine Aouatif. Comparative study of machine learning algorithms using the breast cancer dataset. 2020, p. 83–91. http://dx.doi.org/10.1007/978-3-030-36664-3_10.

[40] Alshayeji Mohammad H, Ellethy Hanem, Abed Sa'ed, Gupta Renu. Computer-aided detection of breast cancer on the wisconsin dataset: An artificial neural networks approach. Biomed Signal Process Control 2022;71:103141. http://dx.doi.org/10.1016/j.bspc.2021.103141, URL https://www.sciencedirect.com/science/article/pii/S1746809421007382.

[41] Abdulkareem Sulyman, Abdulkareem Zainab. An evaluation of the wisconsin breast cancer dataset using ensemble classifiers and RFE feature selection technique. Int J Sci Basic Appl Research (IJSBAR) 2021;55:67–80, URL https://www.gssrr.org/index.php/JournalOfBasicAndApplied/article/view/12300.

[42] Singh Law, Khanna Munish, Singh Dr. An enhanced soft-computing based strategy for efficient feature selection for timely breast cancer prediction: Wisconsin diagnostic breast cancer dataset case. Multimedia Tools Appl 2024;83:1–66. http://dx.doi.org/10.1007/s11042-024-18473-9.

[43] Mushtaq Zohaib, Qureshi Muhammad Farrukh, Abbass Muhammad Jamshed, Al-Fakih Sadeq Mohammed Qaid. Effective kernel-principal component analysis based approach for wisconsin breast cancer diagnosis. Electron Lett 2023;59(2):e212706. http://dx.doi.org/10.1049/ell2.12706, URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ell2.12706.

[44] Farida Yuniar, Ulinnuha Nurissaidah, Sari Silvia, Desinaini Latifatun. Comparing support vector machine and naïve Bayes methods with a selection of fast correlation based filter features in detecting parkinson's disease. Lontar Komp'ut J Ilm Teknol Inf 2023;14:80. http://dx.doi.org/10.24843/LKJITI.2023.v14.i02.p02.

[45] Srinivasan Saravanan, Ramadass Parthasarathy, Kumar M Sandeep, Selvam Karthikeyan, Shivahare Basu, Shah Mohd. Detection of parkinson disease using multiclass machine learning approach. Sci Rep 2024;14:13813. http://dx.doi.org/10.1038/s41598-024-64004-9.

[46] Akila B, Nayahi Jesu. Parkinson classification neural network with mass algorithm for processing speech signals. Neural Comput Appl 2024;36:1–17. http://dx.doi.org/10.1007/s00521-024-09596-z.

[47] Ramdhani Yudi, Apra Dhia, Alamsyah Doni. Feature selection optimization based on genetic algorithm for support vector classification varieties of raisin. Indones J Electr Eng Comput Sci 2023;30:192. http://dx.doi.org/10.11591/ijeecs.v30.i1.pp192-199.

[48] Kavalcı Yılmaz Esra, Oğuz Taha, Adem Kemal. A CNN-based hybrid approach to classification of raisin grains. 2023, URL https://as-proceeding.com/index.php/icfar/article/view/147.

[49] Dliyauddin Muhammad, Shidik Guruh, Affandy, Soeleman M. Enhancing machine learning accuracy in detecting preventable diseases using backward elimination method. J Media Informat Budidarma 2024;8:115. http://dx.doi.org/10.30865/mib.v8i1.7073.

[50] Zhou Xinyi. Raisin classification based on xgboost, SVM, MLP and logistic regression. In: Dai Wanyang, Jin Shi, editors. Second international conference on statistics, applied mathematics, and computing science, vol. 12597. International Society for Optics and Photonics, SPIE; 2023, p. 125970M. http://dx.doi.org/10.1117/12.2672686.

[51] Sahin Onur. Raisin grain classification using machine learning models. In: 2023 IEEE international students' conference on electrical, electronics and computer science. 2023, p. 1–6. http://dx.doi.org/10.1109/SCEECS57921.2023.10063039.

[52] Dafir Zineb, Lamari Yasmine, Slaoui Said Chah. A survey on parallel clustering algorithms for big data. Artif Intell Rev 2021;54(4):2411–43. http://dx.doi.org/10.1007/s10462-020-09918-2.

[53] Ibrahim Ahmed, Hassanien Rokaya. Homogenous and heterogenous parallel clustering: An overview. 2022, URL https://arxiv.org/abs/2202.06478.