Improved classification for imbalanced data using ensemble clustering

Sharanjit Kaur[1], Manju Bhardwaj[2], Adi Maqsood[1], Aditya Maurya[1], Mayank Kumar[1], Nishant Pratap Singh[1]
[1]Department of Computer Science, Acharya Narendra Dev College, University of Delhi, Delhi, India
[2]Department of Computer Science, Maitreyi College, University of Delhi, Delhi, India

Article Info

Article history:

Received Jan 6, 2025
Revised May 29, 2025
Accepted Aug 1, 2025

Keywords:

Auxiliary features
Classification
Ensemble clustering
Imbalanced data
Minority class

ABSTRACT

Imbalanced datasets frequently occur in fields like fraud detection and medical diagnosis, where the number of instances in the majority class vastly exceeds those in the minority class. Traditional classification algorithms often become biased towards the majority class in these scenarios. To address this challenge, we introduce a novel method called improved classification using ensemble clustering (ICEC) for imbalanced datasets in this paper. ICEC merges classification with the strengths of consensus clustering to improve the classifier's generalization ability. This approach utilizes a cluster ensemble to capture the structural characteristics of both the majority and minority classes, and the stable clustering scheme thus delivered is used to generate new auxiliary features. These features enhance the existing feature set, helping classifiers develop a more robust predictive model. Extensive testing on fifteen imbalanced datasets from the knowledge extraction based on evolutionary learning (KEEL) repository demonstrates the effectiveness of our proposed method. The approach was evaluated for random forest (RF) and linear support vector machine (SVM) classifiers on these data sets. Results indicate that ICEC proved to be effective for both classifiers, with an observed F1-score improvement of more than 10% for SVM and 3% for RF.

Corresponding Author:

Manju Bhardwaj
Department of Computer Science, Maitreyi College, University of Delhi
Delhi, India
Email: mbhardwaj@maitreyi.du.ac.in

1. INTRODUCTION

Imbalanced datasets are commonly observed in applications like intrusion detection, e-commerce, stock prediction, spam identification, and medical diagnosis, where the identification of rare class is a crucial issue. In such imbalanced datasets, one class (majority class) significantly outnumbers the other (minority class) [1]. Traditional classification methods may struggle in this context, as they fail to effectively utilize the information contained in the minority class. This imbalance can lead to classifiers that are biased towards the majority class, resulting in poor predictive performance, especially for the minority class [2], [3].

Several techniques have been developed to tackle the class imbalance problem, including resampling methods such as under-sampling and oversampling, cost-sensitive learning [4], and ensemble approaches [5]. Among these, oversampling with synthetic minority oversampling technique (SMOTE) and its variants has been reported to be quite effective by researchers [6], [7]. But these techniques cannot effectively tackle data complexities like noise and class overlap, and may introduce outliers and bias in modeling [8].

To address these challenges, this research suggests employing clustering for classification of imbalanced datasets. As an unsupervised technique, clustering is capable of detecting patterns in unlabeled data. Several researchers have supported the application of clustering to enhance classifier performance on balanced datasets [9]-[11]. Similarly, clustering-based techniques have proven effective in addressing imbalanced data classification problems [1], [12]-[14]. These methods successfully mitigate issues such as overfitting and bias toward majority classes. Lin Sun *et al.* [12] proposed a feature reduction method for imbalanced datasets, which combined similarity-based clustering with adaptive weighted k-nearest neighbor algorithm. Khandokar *et al.* [13] suggested two clustering-based priority sampling techniques for the imbalanced datasets in Liu of random undersampling/oversampling methods. An adaptable framework proposed by Liu *et al.* [14] for incremental learning, employed clustering to group similar instances and selecting representative instances from each cluster, especially from the minority class to create a balanced set of representatives from each class.

In this paper, we propose an intuitive method improved classification for imbalanced datasets using ensemble clustering (ICEC) which leverages clustering to generate distribution-based auxiliary features to improve the performance of a classifier. This research aims to leverage the strengths of both approaches to address the challenge of class imbalance in data as:

- Clustering for better representation: Clustering techniques identify natural groupings within the data, which might not be apparent when simply classifying [15]. The essential idea for clustering imbalanced data is to capture the distribution of each class. By clustering the data, instances of each class (even the minority class) are included in the clustering scheme, although the density of clusters may vary for the minority class.
- Enhanced feature space: Clustering contributes to the creation of new features that capture the underlying structure of the data [16]. These features not only enhance the original feature set but also provide deeper insights into the inherent structure of the data. The enhanced feature set is quite useful for building a robust predictive model and thus, boosts the performance of the classifier. Statistics like minimum, maximum and average distance serve as additional inputs to the classifier, giving the model more context about the relationships of different samples in the same cluster.
- Focus on minority class: The clustering process, being unsupervised, gives equal weight to the minority class. This ensures that the classifier is trained with a balanced perspective without getting biased towards majority class.

Non-conclusive results on the usage of a particular clustering algorithm for generating features motivated us to utilize cluster ensemble to generate auxiliary features for distinguishing classes in imbalanced data. While the literature presents various clustering methods, each comes with its unique strengths and weaknesses [17]. Ensemble clustering, also known as consensus clustering, integrates the insights gained from multiple clustering techniques to better understand the inherent similarities among data points [17]-[19].

To the best of the authors' knowledge, no existing work has used cluster ensemble to generate additional distribution-based features to enrich the dataset. The major contributions of the proposed work include:

- A novel method for enriching the dataset with distribution based auxiliary features.
- Use of robust clustering scheme delivered by cluster ensemble to generate auxiliary features.
- Extensive experimentation with 15 imbalanced datasets to evaluate the efficacy of the proposed ICEC method.

Organization of the paper: The proposed method for generating auxiliary features to boost classifier performance on imbalanced datasets is outlined in section 2. The imbalanced datasets, accompanied by a statistical analysis of the results are briefed in section 3, followed by conclusion in section 4.

2. METHOD

In this section, we describe the ICEC approach adopted to enhance classification robustness by integrating clustering with supervised learning. As depicted in Figure 1, ICEC consists of two phases: i) auxiliary feature generation using cluster ensemble, and ii) model building and prediction, as described below. A stepwise delineation of both the phases is presented in Algorithm 1, and described in the subsections below.

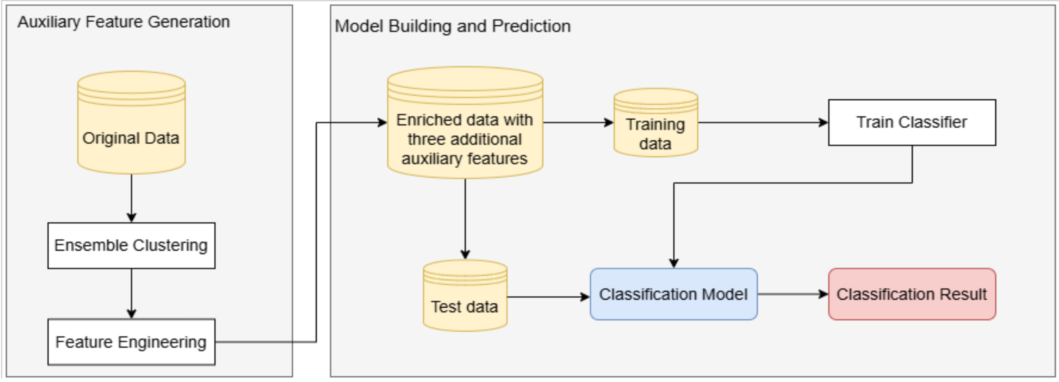


Figure 1. Workflow of ICEC method

Algorithm 1 ICEC method

```
Input: Data set $D$ with #Instances $N$, #Clustering schemes $B$, #Classes $M$, classifier $C$
Output: Enriched data set $\hat{D}$, Classifier model $L$
Phase 1: Generate auxiliary features using ensemble clustering to get $\hat{D}$
  Generate $B$ clustering schemes. Let $C_{ij}$ represent the $j^{\text{th}}$ cluster in $i^{\text{th}}$ clustering scheme (See section 2.1.1. for details)
  Compute co-association matrix $\mathcal{X}$ using (1)
  Use $\mathcal{X}$ to generate final ensemble clustering scheme $\mathcal{F} = \{\mathcal{C}_1 ... \mathcal{C}_K\}$ with $K = 2M$ using $K$-means algorithm.
  for each cluster $\mathcal{C}_j \in \mathcal{F}$ do
      for each point $p$ in $\mathcal{C}_j$ do
          Compute auxiliary features MIN$_p^j$, AVG$_p^j$and MAX$_p^j$ using data members of $\mathcal{C}_j$ (See section 2.1.2.)
          Concatenate features MIN$_p^j$, AVG$_p^j$and MAX$_p^j$ with the feature vector of $p$ to get augmented feature vector
      end for
  end for
  Enriched Data $\hat{D} \leftarrow D$ with auxillary features
Phase 2: Model building and prediction
  $L \leftarrow$ TrainClassifier($C$, $\hat{D}$)
  $T \leftarrow$ Unseen test instance
  Use nearest neighbour approach to identify the cluster $C_t$ in the clustering scheme $\mathcal{F} = \{\mathcal{C}_1 ... \mathcal{C}_K\}$ to which $T$ belongs
  Compute auxiliary features MIN$_T$, AVG$_T$ and MAX$_T$ using data members of $\mathcal{C}_t$
  Concatenate features MIN$_T$, AVG$_T$ and MAX$_T$ with the feature vector of $T$ to get augmented feature vector $T_{aug}$
  PredictedLabel $\leftarrow$ EvaluateClassifier($L$, $T_{aug}$)
```

2.1. Auxiliary feature generation

Following the recommendation of Piernik and Morzy [15], we generate distance-based clustering features, referred to as auxiliary features. Rather than focusing on the distance from the centroid - a method that poses challenges for various clustering algorithms - we leverage the distribution of points within each cluster to create additional features. These auxiliary features are then combined with the existing ones, as suggested in [15], to improve classification performance. The resulting dataset is referred to as the enriched dataset.

2.1.1. Cluster ensemble generation

The proposed method uses ensemble clustering to produce robust and consistent cluster labels to generate auxiliary features, thus improving class separability in the labeled dataset D. Each clustering algorithm has its own unique strengths: for instance, K-means and K-medoids are particularly good at detecting spherical clusters, while agglomerative and spectral clustering excel at capturing hierarchical relationships or graph-based structures. By combining these different clustering methods, the ensemble approach ensures a comprehensive and well-rounded representation of the entire data [20], leading to improved representational accuracy as compared to a single clustering algorithm.

Of all the clustering methods, we chose the methods that are suitable to generate a defined number of clusters. We selected three clustering approaches, viz. K-means, spectral clustering and agglomerative hierarchical clustering to generate three base clustering schemes respectively for observing clustering structures from different views. K-means clustering is a traditional and well-defined approach and is used for its simplicity and computational effectiveness [17], [21]. Spectral clustering uses graph-based structures and the graph cut method to deliver the desired number of connected components called clusters [22], [23]. It works well for arbitrary shape non-convex datasets and makes no assumptions for the global structure of the data. Agglomerative hierarchical clustering makes use of a greedy approach which starts with each point as a singleton cluster, merges a pair of clusters at a time as per selected linkage method till all points are part of a single cluster. The resultant output is in the form of a dendrogram that represents classificatory relationships in the data based on the proximity method used [24].

After the three base clustering schemes are generated, the evidence accumulation model is used for combining the information of multiple partitions in base clusterings to make cluster ensembles. We use a co-association matrix $\mathcal{X}$ to store the association of each pair of points (p, q) as gathered from B clustering schemes. Each entry $\mathcal{X}(p, q)$, denoting number of times two points p and q appear in the same cluster across all B clustering schemes is computed as:

$$\mathcal{X}(p,q) = \frac{1}{B}\sum_{i=1}^{B}\sum_{i=j}^{K}\mathcal{S}(p,q,C_{ij}) \tag{1}$$

Here K is number of clusters, and $\mathcal{S}(p, q, C_{ij})$ is an indicator function for the cluster membership C_{ij} in cluster j of base clustering scheme B_i for any two points p and q as defined:

$$S(p,q,C_{ij}) = \begin{cases} 1 & \text{if both points } p, q \ \in C_{ij} \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

The co-association matrix $\mathcal{X}$ serves as a data matrix for the K-means algorithm to produce the desired number of clusters. Each generated cluster consists of a subset of rows aka points of $\mathcal{X}$ that exhibit greater similarity to one another than to other points. Thus, the final clustering scheme consisting of K clusters is represented as $\mathcal{F} = \{\mathcal{C}_1 ... \mathcal{C}_K\}$. It is important to highlight that the number of clusters (K) is determined by the actual number of classes (M), with K set to $2M$ to avoid creating very small clusters. Since we are considering binary class imbalanced datasets in this study, each dataset results in the creation of four clusters.

2.1.2. Feature engineering from clustering scheme

Once the ensemble clustering process is complete, new features are generated to enrich the original dataset D so as to enhance class separability. It is worthwhile mentioning here that the number of class labels (M) provided with the labeled dataset D are not modified, only additional features are curated to assist classifier to build model with improved predictability. These features capture intra-cluster relationships, offering valuable insights into the internal structure and distribution of data within each cluster. Since the clusters do not overlap, each point is associated with only one cluster $\mathcal{C}_j$. For each data point, the following three new auxiliary features are calculated based on the distribution of the members of the cluster $\mathcal{C}_j$ to which it belongs to.

As clusters are non-overlapping, each point p belongs to one cluster $\mathcal{C}_j$ only and three new auxiliary features are computed for each data point using the distribution of members of $\mathcal{C}_j$ as given:

- Minimum distance (MIN_p^j): this measures the distance of the data point p ($p \in \mathcal{C}_j$) to the closest point x in the same cluster. This feature captures local density and compactness around a point in a cluster. Formally, it is computed as:

$$\text{MIN}_p^j = min(\mathcal{D}(p,x)) \wedge p \neq x \ \forall x \in \mathcal{C}_j \tag{3}$$

where $\mathcal{D}(p, x)$ denotes distance between two data points p and x.

- Average distance (AVG_p^j): this metric captures the overall cohesion of the cluster by calculating the mean distance of a sample p from all other members of its cluster ($\mathcal{C}_j$).

$$\text{AVG}_p^j = avg(\mathcal{D}(p, x)) \wedge p \neq x \tag{4}$$

- Maximum distance (MAX_p^j): this represents the farthest distance of a sample p from other points in the same cluster $\mathcal{C}_j$, which reflects its spread or boundary, thus capturing the wideness of the cluster.

$$\text{MAX}_p^j = max(\mathcal{D}(p, x)) \wedge p \neq x \tag{5}$$

Cluster-derived features significantly enhance the original dataset D with n features by embedding structural information for each point, resulting in an enriched dataset $\hat{D}$ with $n + 3$ features. The new features reflect relationships rooted in the data distribution within each cluster, offering insights not provided by the original n features. The time complexity for generating these auxiliary features is O($BKN + N^2$), where B is the number of base clustering schemes, K is the number of clusters in each scheme, and N is the total number of points in the dataset.

2.2. Model building and prediction

Once the dataset is enriched with auxiliary features ($\hat{D}$), the classification algorithm is used to build a model L which is used to predict class labels of unseen instances. Given a test instance T, the cluster label is computed employing the nearest neighbor approach. The centroids of the generated clusters in the clustering scheme $\mathcal{F} = \{\mathcal{C}_1 ... \mathcal{C}_K\}$ are used to identify the cluster label of T. Subsequently, three auxiliary features are computed for T and augmented with the original feature vector of size n. Once the updated feature vector T_{aug} of size $n + 3$ is obtained, it is fed to the trained classifier L to predict the class label.

3. RESULTS AND DISCUSSION

In this section, we analyze how enriching a dataset with clustering-based auxiliary features affects classifier performance. For the sake of simplicity, we have opted to analyse the performance of two simple and widely recognized classifiers: random forest (RF) and linear support vector machine (SVM) in this study.

3.1. Datasets used

Table 1 lists the fifteen imbalanced datasets downloaded from knowledge extraction based on evolutionary learning (KEEL) repository [25], used in this study. Each dataset is a binary class dataset and is characterized by a skewed class distribution, meaning that the number of instances in one (majority) class substantially exceeds that of the other (minority) class. The column imbalance ratio (IR) in the table shows the ratio of the number of instances in the majority class to those in the minority class as mentioned for each dataset.

Table 1. Imbalanced datasets used in the study; IR-imbalance ratio

S. No	Name	IR	#Attributes	#Instances
1	Ecoli1	3.36	7	336
2	Glass0	2.06	9	214
3	Glass5	22.78	9	214
4	Glass6	6.38	9	214
5	Haberman	2.78	3	306
6	New-thyroid1	5.14	5	215
7	New-thyroid2	5.14	5	215
8	Vehicle0	3.25	18	846
9	Vehicle1	2.9	18	846
10	Vehicle2	2.88	18	846
11	Vehicle3	2.99	18	846
12	Vowel0	9.98	13	988
13	Wisconsin	1.86	9	683
14	Yeast1	2.46	8	1484
15	Yeast6	41.4	8	1484

3.2. Evaluation metrics

It is a well established fact that the F1-score is an effective metric for assessing the performance of any classifier on an imbalanced dataset compared to the accuracy metric [26]. F1-score is defined as harmonic mean of $Precision$ and $Recall$ (See (6)). We have used the extension of F1-score, macro F1-score, to assess the classifier performance across both classes taken together.

$$Macro\,F1\text{-}score = \frac{\sum(F1\text{-}score)}{No.\ of\ Classes} \text{ where } F1\text{-}score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

3.3. Performance Analysis

In this subsection, we assess the effectiveness of the enhanced feature sets developed by the proposed method by performing a comparative analysis of classifier performance on select imbalanced datasets.

3.3.1. Impact of auxiliary features

Our first goal is to analyze the impact of additional clustering-based features on the performance of RF and linear SVM classifiers on 15 imbalanced datasets (Table 1). Experiments were performed on the original feature sets (ORG) and the enhanced feature sets curated by extending the original feature set by generating all possible seven combinations of the three clustering-based auxiliary features (MAX, MIN and AVG) outlined in section 2.1.2. Ten-fold cross-validation was carried out for each data set, and average macro F1-score was computed. Table 2 presents the average macro F1-scores obtained for the two classifiers using original (column 4) and enhanced feature sets (column 5-11) respectively.

Table 2. Macro F1-scores obtained using original and seven curated feature sets for the selected datasets

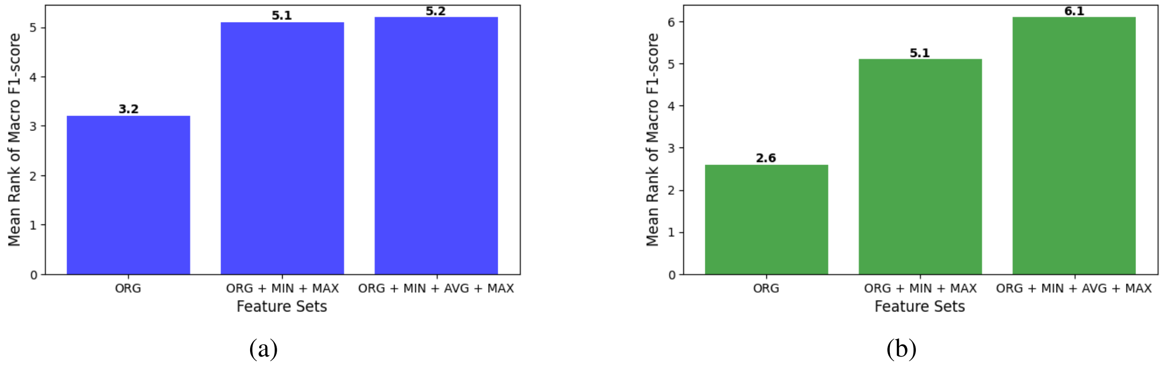
S. No	Dataset	CFR	ORG	ORG +MIN	ORG +AVG	ORG +MAX	ORG+MIN +AVG	ORG+MIN +MAX	ORG+AVG +MAX	ORG+MIN +AVG+MAX
1	Ecoli1	RF	85.77	86.67	84.17	84.35	85.61	85.14	86.07	86.44
		SVM	84.98	84.86	84.49	85.74	84.12	85.31	86.20	86.33
2	Glass0	RF	78.00	79.37	81.03	78.64	81.11	81.18	81.61	81.41
		SVM	41.06	42.05	42.14	41.65	43.15	42.74	42.49	42.74
3	Glass5	RF	82.60	82.60	79.15	84.40	79.15	82.60	82.60	82.60
		SVM	77.47	74.02	77.47	83.51	74.02	83.51	83.51	83.51
4	Glass6	RF	85.77	86.67	84.17	84.35	85.61	85.14	86.07	86.44
		SVM	84.98	84.86	84.49	85.74	84.12	85.31	86.20	86.33
5	Haberman	RF	54.50	56.11	54.03	51.06	57.06	54.42	56.07	54.23
		SVM	42.87	45.07	42.87	42.87	45.07	47.05	42.87	45.07
6	New-thyroid1	RF	97.22	98.07	98.07	96.28	98.07	98.07	96.52	97.37
		SVM	96.08	96.08	96.08	96.08	97.16	96.08	96.08	97.16
7	New-thyroid2	RF	95.76	96.37	96.89	96.37	94.94	97.08	93.04	96.46
		SVM	93.70	93.70	96.06	93.70	96.62	93.70	96.62	96.62
8	Vehicle0	RF	95.64	95.58	96.00	96.47	95.63	95.43	95.69	96.30
		SVM	95.00	95.11	96.15	95.28	95.61	95.25	96.70	96.73
9	Vehicle1	RF	66.82	68.23	68.27	67.96	65.15	67.20	66.20	67.59
		SVM	71.84	71.59	72.48	71.75	72.69	71.81	72.10	72.41
10	Vehicle2	RF	95.92	96.02	95.74	96.08	96.26	96.11	96.09	96.16
		SVM	95.15	95.01	94.86	95.10	95.63	95.07	95.10	95.30
11	Vehicle3	RF	66.54	67.05	67.20	66.93	68.63	68.15	66.99	67.28
		SVM	72.24	71.44	72.94	72.34	72.73	72.49	73.32	72.21
12	Vowel0	RF	98.77	98.77	99.28	98.77	98.62	98.52	99.54	99.03
		SVM	90.99	90.58	94.45	97.54	94.66	95.78	97.54	96.41
13	Wisconsin	RF	96.42	96.13	96.78	96.95	96.81	96.78	96.79	96.63
		SVM	95.89	96.42	96.78	96.78	96.26	96.43	96.78	96.26
14	Yeast1	RF	69.13	70.66	71.34	70.46	68.77	71.37	69.93	70.88
		SVM	59.42	60.81	59.62	59.31	62.36	60.22	59.25	62.74
15	Yeast6	RF	69.97	70.34	66.28	73.08	68.71	71.28	69.01	69.81
		SVM	49.40	51.90	49.40	49.40	56.45	51.90	51.40	60.47

Comparison of F1-scores across each row reveals improved performance over original feature set on all data sets because of enhanced feature set. Sometimes, adding just one auxiliary feature to the original feature set can yield the highest F1-score for a given dataset. For instance, the *Ecoli1* (S.No 1) dataset reports a maximum F1-score of 86.67 for the feature set (ORG+MIN) for RF classifier, as compared to a score of 85.77 on the ORG feature set and 86.44 for (ORG+MIN+AVG+MAX) feature set. On the other hand, consider the *Yeast6* (S.No 15) data set, where an increase of more than 10% for SVM classifier is observed for (ORG+MIN+AVG+MAX) feature set as compared to original feature set. In a limited number of instances, either no improvement or a decline in performance is noted; however, these cases are quite rare. Thus, in general, a positive effect of the clustering-based features on classifier performance cannot be ruled out, as addition of auxiliary features to the original feature set tends to improve classifier performance significantly in most of the datasets.

In order to identify the best performer among all feature sets, we compute the average rank score for each feature set. For each data set and classifier, the scores on the eight feature sets are ranked from 1 to 8, with 1 indicating the lowest score and 8 the highest. Average rank is assigned whenever there is a tie in ranks of two or more feature sets. Mean ranks for each feature set are computed by averaging the ranks across the feature set column. Table 3 shows the mean ranks of classifiers for the eight feature sets. It can be seen from the Table 3 that the mean rank for the ORG feature set is the lowest in the three rows, while the enhanced feature set that includes all three clustering-based auxiliary features MIN, AVG and MAX (last column of Table 2) has the highest mean rank. Mean ranks for the original feature set are plotted along with that of the two best performer feature sets - (ORG+MIN+MAX) (Figure 2(a)) and (ORG+MIN+AVG+MAX) (Figure 2(b)). Visual comparison further supports the superiority of the enriched feature set with all three auxiliary features.

Table 3. Mean ranks of macro F1-scores over all datasets for *ORG* and curated feature sets

S.No	Classifier	ORG	ORG +MIN	ORG +AVG	ORG +MAX	ORG+MIN +AVG	ORG+MIN +MAX	ORG+AVG +MAX	ORG+MIN +AVG+MAX
1	RF+SVM	2.9	3.8	4.4	4.2	5.1	5.1	4.9	5.6
2	RF	3.2	4.3	4.8	4.5	4.4	5.1	4.4	5.2
3	SVM	2.6	3.3	3.9	3.8	5.8	5.1	5.4	6.1



(a) (b)

Figure 2. Mean ranks of macro F1-scores for three feature sets of (a) RF and (b) linear SVM

3.3.2. Statistical analysis

In this subsection, we validate the superior performance of classifiers on enhanced feature sets by applying the Friedman rank sum test [27] to the multicolumn data in Table 2. Friedman test is a non-parametric statistical method used to assess multiple related groups for significant statistical differences in data distributions. According to the test, the null hypothesis states that there is no significant statistical difference among the F1-scores obtained for eight feature sets. Given $N(= 30)$ sets of scores for $f(= 8)$ feature sets, the F1-scores are ranked as described in the subsection above. The rank sum (R_j) is computed for each feature set $(j = 1 \ldots 8)$ to calculate the Q test statistic as:

$$Q = \frac{12}{N.f.(f+1)} \sum_{j=1}^{f} R_j^2 - 3N(f+1) \tag{7}$$

The Q test statistic follows a Chi-square distribution with $8 - 1 = 7$ degrees of freedom. The critical value of test statistic at 95% significance level is 14.067, which is less than the calculated statistical value Q of 26.67. Hence, the null hypothesis is rejected, indicating significant differences in the classifier performance on different feature sets. Higher rankings of enhanced features sets (as seen in Table 3) thus statistically confirm the positive effect of clustering-based auxiliary features on the performance of classifiers on imbalanced datasets.

4. CONCLUSION

The proposed ICEC method uses ensemble clustering to create auxiliary features that improve classifier performance on imbalanced datasets, as demonstrated by experiments on fifteen imbalanced data sets using RF and linear SVM classifiers. The study vindicates that the auxiliary features assist the classifier by providing comprehensive understanding of data patterns to generate a robust classification model. Hence, this approach proves useful for critical applications such as cyber attack monitoring, fraud detection and disease diagnosis, where effective identification of rare case is crucial. However, the results of the proposed method are highly dependent on the number of clusters (K) generated. Indeed the idea of utilizing various ensemble strategies to create an effective clustering scheme may be explored in near future, so that a prior specification of K is not required. Additionally, optimization techniques can be utilized for identification of best auxiliary features for a given dataset.

ACKNOWLEDGMENTS

We acknowledge the research grant received under Anusandhan Kosh (2024-2025) of Acharya Narendra Dev College, University of Delhi for carrying out research work of this paper.

FUNDING INFORMATION

The research was supported by the research grant received under Anusandhan Kosh (2024-2025) of Acharya Narendra Dev College, University of Delhi.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sharanjit Kaur	✓	✓		✓	✓	✓			✓		✓	✓		✓
Manju Bhardwaj	✓	✓		✓	✓	✓				✓			✓	
Adi Maqsood	✓		✓			✓			✓	✓				
Aditya Maurya		✓	✓			✓			✓		✓			
Mayank Kumar		✓	✓		✓			✓		✓				
Nishant Pratap Singh	✓		✓		✓					✓	✓			

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject Administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding Acquisition
Fo	: **Fo**rmal Analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The supporting data of this study are openly available in KEEL repository at `http://www.keel.es/` [25]. The data that support the findings of this study are available from the corresponding author, [initials: MB], upon reasonable request.

REFERENCES

[1] W. C. Lin, C. F. Tsai, Y. H. Hu, and J. S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409–410, pp. 17–26, Oct. 2017, doi: 10.1016/j.ins.2017.05.008.

[2] X. Yuan, C. Sun, and S. Chen, "A clustering-based adaptive undersampling ensemble method for highly unbalanced data classification," *Applied Soft Computing*, vol. 159, p. 111659, Jul. 2024, doi: 10.1016/j.asoc.2024.111659.

[3] V. Bhatnagar, M. Bhardwaj, and A. Mahabal, "Comparing SVM ensembles for imbalanced datasets," in *Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications*, *ISDA'10, IEEE*, Nov. 2010, pp. 651–657, doi: 10.1109/ISDA.2010.5687191.

[4] L. Zhou and H. Wang, "Loan Default Prediction on Large Imbalanced Data Using Random Forests," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 10, no. 6, pp. 1519–1525, Sep. 2012, doi: 10.11591/telkomnika.v10i6.1323.

[5] S. Abokadr, A. Azman, H. Hamdan, and N. Amelina, "Handling Imbalanced Data for Improved Classification Performance: Methods and Challenges," in *2023 3rd International Conference on Emerging Smart Technologies and Applications*, eSmarTA 2023, IEEE, Oct. 2023, pp. 1–8, doi: 10.1109/eSmarTA59349.2023.10293442.

[6] N. Alamsyah, Budiman, T. P. Yoga, and R. Y. R. Alamsyah, "A stacking ensemble model with SMOTE for improved imbalanced classification on credit data," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 22, no. 3, pp. 657–664, Feb. 2024, doi: 10.12928/TELKOMNIKA.v22i3.25921.

[7] Q. Chen, Z. L. Zhang, W. P. Huang, J. Wu, and X. G. Luo, "PF-SMOTE: A novel parameter-free SMOTE for imbalanced datasets," *Neurocomputing*, vol. 498, pp. 75–88, Aug. 2022, doi: 10.1016/j.neucom.2022.05.017.

[8] M. Alimoradi, R. Sadeghi, A. Daliri, and M. Zabihimayvan, "Statistic deviation mode balancer (SDMB): A novel sampling algorithm for imbalanced data," *Neurocomputing*, vol. 624, p. 129484, Apr. 2025, doi: 10.1016/j.neucom.2025.129484.

[9] J. Xiao, Y. Tian, L. Xie, X. Jiang, and J. Huang, "A Hybrid Classification Framework Based on Clustering," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2177–2188, Apr. 2020, doi: 10.1109/TII.2019.2933675.

[10] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, "K-Means+ID3: A novel method for supervised anomaly detection by cascading k-Means clustering and ID3 decision tree learning methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 345–354, Mar. 2007, doi: 10.1109/TKDE.2007.44.

[11] C. Kaewchinpom, N. Vongsuchoto, and A. Srisawat, "A combination of decision tree learning and clustering for data classification," in *Proceedings of the 2011 8th International Joint Conference on Computer Science and Software Engineering, JCSSE 2011*, *IEEE*, May 2011, pp. 363–367, doi: 10.1109/JCSSE.2011.5930148.

[12] L. Sun, J. Zhang, W. Ding, and J. Xu, "Feature reduction for imbalanced data classification using similarity-based feature clustering with adaptive weighted K-nearest neighbors," *Information Sciences*, vol. 593, pp. 591–613, May 2022, doi: 10.1016/j.ins.2022.02.004.

[13] I. A. Khandokar, Abdullah-All-Tanvir, T. Khondokar, N. T. Jhilik, and S. Shatabda, "A Clustering Based Priority Driven Sampling Technique for Imbalance Data Classification," in *International Conference on Software, Knowledge Information, Industrial Management and Applications, SKIMA, IEEE*, Dec. 2022, pp. 176–180, doi: 10.1109/SKIMA57145.2022.10029565.

[14] Y. Liu, G. Du, C. Yin, H. Zhang, and J. Wang, "Clustering-based incremental learning for imbalanced data classification," *Knowledge-Based Systems*, vol. 292, p. 111612, May 2024, doi: 10.1016/j.knosys.2024.111612.

[15] M. Piernik and T. Morzy, "A study on using data clustering for feature extraction to improve the quality of classification," *Knowledge and Information Systems*, vol. 63, no. 7, pp. 1771–1805, 2021, doi: 10.1007/s10115-021-01572-6.

[16] E. Hancer, B. Xue, and M. Zhang, "A survey on feature selection approaches for clustering," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4519–4545, Aug. 2020, doi: 10.1007/s10462-019-09800-w.

[17] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, Jun. 2010, doi: 10.1016/j.patrec.2009.09.011.

[18] A. Strehl and J. Ghosh, "Cluster ensembles - A knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, no. 3, pp. 583–617, 2003, doi: 10.1162/153244303321897735.

[19] V. Bhatnagar, S. Ahuja, and S. Kaur, "Discriminant analysis-based cluster ensemble," *International Journal of Data Mining, Modelling and Management*, vol. 7, no. 2, pp. 83–107, 2015, doi: 10.1504/IJDMMM.2015.069248.

[20] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee, and R. Enayatifar, "From clustering to clustering ensemble selection: A review," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104388, Sep. 2021, doi: 10.1016/j.engappai.2021.104388.

[21] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," *Electronics*, vol. 9, no. 8, p. 1295, Aug. 2020, doi: 10.3390/electronics9081295.

[22] M. C. V. Nascimento and A. C. P. L. F. de Carvalho, "Spectral methods for graph clustering – A survey," *European Journal of Operational Research*, vol. 211, no. 2, pp. 221–231, Jun. 2011, doi: 10.1016/j.ejor.2010.08.012.

[23] H. Jia, S. Ding, X. Xu, and R. Nie, "The latest research progress on spectral clustering," *Neural Computing and Applications*, vol. 24, no. 7–8, pp. 1477–1486, Jun. 2014, doi: 10.1007/s00521-013-1439-2.

[24] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *WIREs Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, Jan. 2012, doi: 10.1002/widm.53.

[25] J. Alcalá-Fdez et al., "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2–3, pp. 255–287, 2011.

[26] T.-T. Wong, "Linear Approximation of F-Measure for the Performance Evaluation of Classification Algorithms on Imbalanced Data Sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 753–763, Feb. 2022, doi: 10.1109/TKDE.2020.2986749.

[27] M. Friedman, "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance," *Journal of the American Statistical Association*, vol. 32, no. 200, p. 675, Dec. 1937, doi: 10.2307/2279372.

BIOGRAPHIES OF AUTHORS

Sharanjit Kaur with Doctoral Studies in the area of Stream Clustering is currently working as Professor in Department of Computer Science, Acharya Narendra Dev College, University of Delhi. Her research interest spans the area of databases, stream clustering, classification, graph mining, social network analysis, text mining, recommender system and epidemiology. She can be contacted at email: sharanjitkaur@andc.du.ac.in.

Manju Bhardwaj is currently an Associate Professor in Department of Computer Science at Maitreyi College, University of Delhi. Her doctoral research work is centered on machine learning with a focus on classification ensembles. Currently, her research interests encompass sentiment analysis, natural language processing and large language models. She can be contacted at email: mbhardwaj@maitreyi.du.ac.in.

Adi Maqsood is currently pursuing a B.Sc. (Hons) in Computer Science at Acharya Narendra Dev College, University of Delhi. His research interests include machine learning, data clustering, imbalanced data classification, and handling of unstructured data. He can be contacted at email: adimaqsood1@gmail.com.

Aditya Maurya is currently pursuing a B.Sc. (Hons.) in Computer Science from Acharya Narendra Dev College, University of Delhi. His interests include data mining, NLP, and machine learning. He has worked on projects in emotion visualization through NLP, and maritime situation awareness through text mining. He can be contacted at email: adim7305@gmail.com.

Mayank Kumar is currently pursuing a B.Sc. (Hons) in Computer Science at Acharya Narendra Dev College, University of Delhi. His academic interests include data mining, clustering techniques, natural language processing, and advanced machine learning methodologies. He is passionate about leveraging these technologies to solve real-world problems. He can be contacted at email: mayank.4126@gmail.com.

Nishant Pratap Singh is pursuing a B.Sc. (Hons.) in Computer Science at Acharya Narendra Dev College, University of Delhi. His research interests include data mining, data analysis, machine learning, data science, and database management systems (DBMS). He can be contacted at email: npratapsingh084@gmail.com.