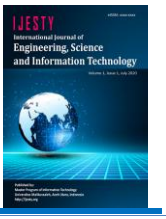
International Journal of Engineering, Science and Information Technology
Volume 5 No. 3 (2025) pp. 51-57
ISSN 2775-2674 (online)
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i3.894
Research Paper, Short Communication, Review, Technical Paper



A Hybrid GDHS and GBDT Approach for Handling Multiclass Imbalanced Data Classification

Hartono[1*], Muhammad Khahfi Zuhanda[1], Rahmad Syah[1], Sayuti Rahman[1], Erianto Ongko[2]

[1]Department of Informatics, Faculty of Engineering, Universitas Medan Area, Medan, Indonesia
[2]Department of Computer Science, Faculty of Engineering, Institut Modern Arsitektur dan Teknologi, Medan, Indonesia

**Corresponding author Email: hartono@staff.uma.ac.id*

The manuscript was received on 25 November 2024, revised on 2 January 2025, and accepted on 1 April 2025, date of publication 11 May 2025

Abstract

Multiclass imbalanced classification remains a significant challenge in machine learning, particularly when datasets exhibit high Imbalance Ratios (IR) and overlapping feature distributions. Traditional classifiers often fail to accurately represent minority classes, leading to biased models and suboptimal performance. This study proposes a hybrid approach combining Generalization potential and learning Difficulty-based Hybrid Sampling (GDHS) as a preprocessing technique with Gradient Boosting Decision Tree (GBDT) as the classifier. GDHS enhances minority class representation through intelligent oversampling while cleaning majority classes to reduce noise and class overlap. GBDT is then applied to the resampled dataset, leveraging its adaptive learning capabilities. The performance of the proposed GDHS+GBDT model was evaluated across six benchmark datasets with varying IR levels, using metrics such as Matthews Correlation Coefficient (MCC), Precision, Recall, and F-Value. Results show that GDHS+GBDT consistently outperforms other methods, including SMOTE+XGBoost, CatBoost, and Select-SMOTE+LightGBM, particularly on high-IR datasets like Red Wine Quality (IR = 68.10) and Page-Blocks (IR = 188.72). The method improves classification performance, especially in detecting minority classes, while maintaining high accuracy.

Keywords: *Multiclass Imbalanced Classification, Machine Learning, Hybrid Sampling, GDHS, GBDT.*

1. Introduction

Imbalanced data classification is an increasingly prevalent challenge across various application domains such as fraud detection [1], medical diagnosis [2], security systems [3], and pattern recognition [4]. In the context of multiclass data, the challenge becomes even more complex due to the imbalance between two classes [5] and among multiple overlapping majority and minority classes, each with varying degrees of sample representation [6]. In multiclass imbalanced classification, the difficulty lies in the fact that a model must learn to differentiate not just between a majority and a single minority class (as in binary imbalance) but among several classes [7], which may have drastically different numbers of instances [8]. This situation often leads to what is known as the "multiclass imbalance problem," where: One or more classes dominate the training process (majority classes)[9], Several other courses are severely underrepresented (minority classes), Some classes may be borderline[10] or overlapping [11], leading to ambiguity in decision boundaries [12].

Traditional classification algorithms perform poorly under these conditions because they optimize for overall accuracy [13]. This can be misleading: achieving high accuracy by correctly classifying majority class samples while consistently misclassifying minority ones [14]. Furthermore, the inter-class imbalance (i.e., the imbalance between any pair of classes) and intra-class variability (variance within the same class) introduce further difficulties, especially when classes have overlapping feature distributions [15].

One widely used approach to improve classification performance is the Gradient Boosting Decision Tree (GBDT). GBDT is known for its ability to build strong ensembles of decision trees gradually and adaptively. However, recent studies show that while GBDT performs well on balanced datasets, it remains vulnerable to distribution bias when applied to imbalanced data, especially in multiclass settings. Several studies have proposed modifications to the loss function, such as Focal Loss, Class Balanced Loss, and Asymmetric Loss, which aim to penalize misclassification of minority classes more heavily. While these methods improve sensitivity toward minority classes, they rely heavily on the original data distribution [16].

As an alternative algorithm-level approach, data-level resampling methods offer distinct advantages because they do not alter the model architecture and can be applied across various learning algorithms [17]. In this domain, Yan et al developed the GDHS (Generalization potential and learning Difficulty-based Hybrid Sampling) method. [18] provides a significant contribution. GDHS is a hybrid sampling technique designed explicitly for multiclass imbalanced data, combining two main strategies: intelligent oversampling based on learning difficulty and the generalization potential of minority samples, and majority class cleaning using three different techniques to handle class overlaps. The study demonstrated that GDHS[18] consistently outperforms 12 state-of-the-art methods, such as SMOTE, MDO, and SHSampler, in terms of mGM (mean geometric mean) and MAUC (multiclass area under the curve) metrics.

Compared to other methods like Local density-based adaptive sampling[19], which also adopt a hybrid approach, GDHS offers improvements by integrating safe sampling and generalization-aware weighting, considering sample density and heterogeneous distribution in the synthetic space. Meanwhile, methods like MC-RBO[20] and MC-CCR[21] focus on potential and energy-based oversampling functions but tend to neglect majority class cleaning, which often plays a critical role in classification inaccuracy.

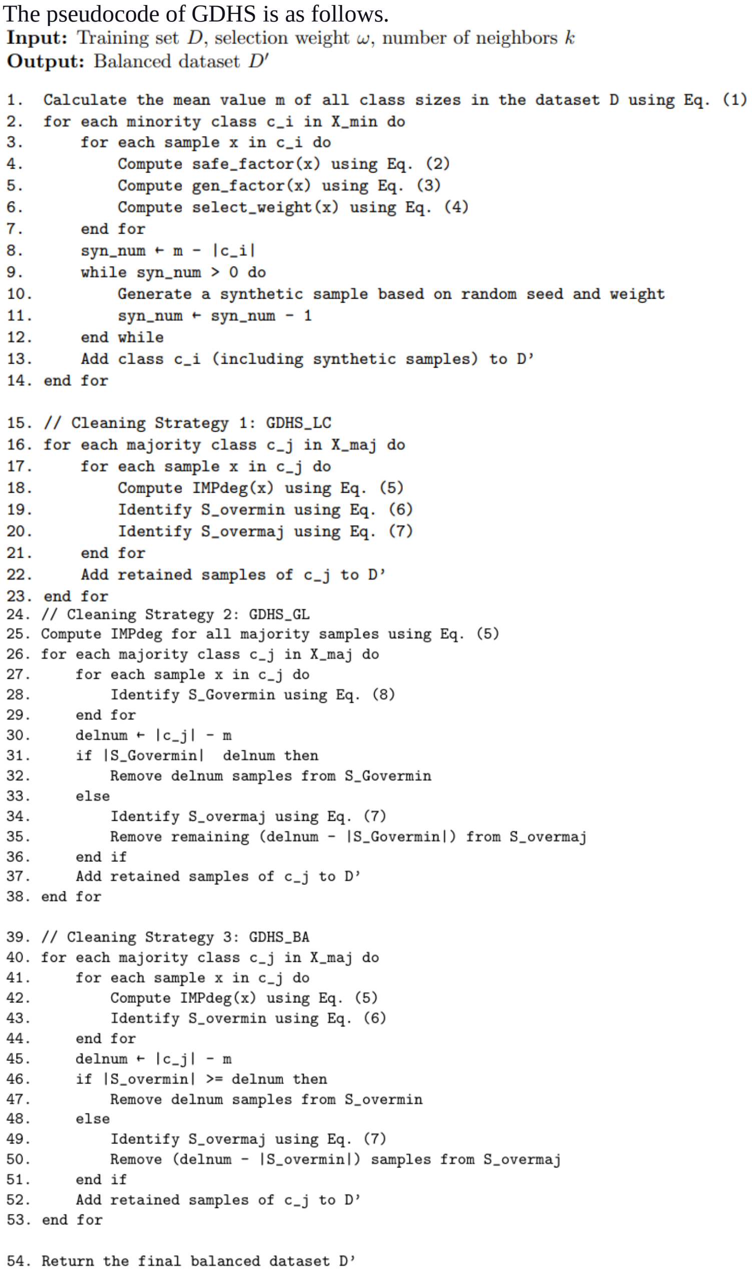
In this context, integrating GDHS and GBDT[22] represents a promising synergistic strategy. GDHS prepares a more representative and clean dataset through intelligent sampling-based preprocessing. At the same time, GBD, modified with a loss function adaptive to minority classes, can optimally leverage this data during training. Combining data-level preprocessing and algorithm-level learning strengths, this approach is expected to enhance accuracy, Recall, and AUC for minority classes without sacrificing the model's overall performance.

The urgency of this integration becomes even more critical in real-world applications that require high sensitivity to minority classes, such as rare disease detection or financial fraud analysis. Therefore, this study proposes the integration of GDHS as a preprocessing stage and GBDT as the primary classifier, and compares it against other baselines such as SMOTE+XGBoost[23], CatBoost[24], and Select-SMOTE+LightGBM[25] to evaluate the effectiveness of the proposed hybrid model.

2. Method

2.1. Generalization Potential and Learning Difficulty-Based Hybrid Sampling

The pseudocode of GDHS is as follows.



Input: Training set D, selection weight ω, number of neighbors k

Output: Balanced dataset D'

```
Calculate the mean value m of all class sizes in the dataset D using Eq. (1)
for each minority class c_i in X_min do
    for each sample x in c_i do
        Compute safe_factor(x) using Eq. (2)
        Compute gen_factor(x) using Eq. (3)
        Compute select_weight(x) using Eq. (4)
    end for
    syn_num ← m - |c_i|
    while syn_num > 0 do
        Generate a synthetic sample based on random seed and weight
        syn_num ← syn_num - 1
    end while
    Add class c_i (including synthetic samples) to D’
end for

// Cleaning Strategy 1: GDHS_LC
for each majority class c_j in X_maj do
    for each sample x in c_j do
        Compute IMPdeg(x) using Eq. (5)
        Identify S_overmin using Eq. (6)
        Identify S_overmaj using Eq. (7)
    end for
    Add retained samples of c_j to D’
end for
// Cleaning Strategy 2: GDHS_GL
Compute IMPdeg for all majority samples using Eq. (5)
for each majority class c_j in X_maj do
    for each sample x in c_j do
        Identify S_Govermin using Eq. (8)
    end for
    delnum ← |c_j| - m
    if |S_Govermin|  delnum then
        Remove delnum samples from S_Govermin
    else
        Identify S_overmaj using Eq. (7)
        Remove remaining (delnum - |S_Govermin|) from S_overmaj
    end if
    Add retained samples of c_j to D’
end for

// Cleaning Strategy 3: GDHS_BA
for each majority class c_j in X_maj do
    for each sample x in c_j do
        Compute IMPdeg(x) using Eq. (5)
        Identify S_overmin using Eq. (6)
    end for
    delnum ← |c_j| - m
    if |S_overmin| >= delnum then
        Remove delnum samples from S_overmin
    else
        Identify S_overmaj using Eq. (7)
        Remove (delnum - |S_overmin|) samples from S_overmaj
    end if
    Add retained samples of c_j to D’
end for

Return the final balanced dataset D’
```

The provided pseudocode outlines the GDHS, designed to address class imbalance in supervised learning tasks. The algorithm performs oversampling for minority classes and various cleaning (undersampling) strategies for majority classes to yield a balanced dataset *D′*.

The process begins by calculating the mean value *m* of all class sizes in the original dataset *D* using Equation 1.

$$m = \frac{\sum_{i=1}^{n}|c_i|}{n} \quad (1)$$

Where c_i denotes the set of samples of the *i th* class of the imbalanced dataset *D*

For each minority class c_i, the algorithm iterates over its samples and computes three key metrics: safe_factor(x) using Equation (2), gen_factor(x) using Equation (3), and select_weight(x) using Equation (4).

$$Safe_{factor} = k_{hom}/k \quad (2)$$

Where k_{hom} denotes the number of samples belonging to the same class in the *k*-nearest neighbor

$$gen_{factor} = \frac{1}{(1+Dens_{rea}+Het_{rat})} \quad (3)$$

Where Het_{rat} denotes heterogeneous samples and $Dens_{rea}$ denotes density

$$Select_{weight} = w * safe_{factor} + (1 - w) * gen_{factor} \quad (4)$$

Where *w* denotes weight

These metrics guide the generation of synthetic samples. The number of synthetic samples required for class c_i is determined by the difference m−|c_i|. New samples are generated using a weighted random strategy until the desired number is reached. Once completed, the enhanced minority class (original plus synthetic data) is added to the balanced dataset D′.

Next, the algorithm applies one of three possible cleaning strategies to handle the majority classes. The first strategy, GDHS_LC (Local Cleaning), involves computing the importance degree IMPdeg(x) for each sample using Equation (5), and identifying two sets: S_overmin (majority samples similar to minority ones) and S_overmaj (samples that are outliers within the majority class), using Equations (6) and (7), respectively. The retained samples from the majority class, c_j, are then added to *D′*.

$$IM_{Peg} = \sum_{x \in S_{ast}(x_q)} Select_{weight}(x) \quad (5)$$

Where $S_{ast}(X_q)$ Denotes the set of all minority samples.

$$S_{overmin} = \left\{x_j | x_j \in x_{maj}^{c(x_q)} \wedge IM_{Peg}(X_j) > avg(IM_{Peg}\left(x_{maj}^{c(x_q)}\right)))\right\} \quad (6)$$

$S_{overmin}$ Denotes the set of majority samples that need to be cleaned.

$$S_{overmaj} = \left\{x_j | x_j \in x_{maj}^{c(x_q)} \wedge R_M(X_j) > \theta_{rem}\right\} \quad (7)$$

$S_{overmaj}$ Denotes the set of majority samples that overlapped with other majority class samples.

The second strategy, GDHS_GL (Global Cleaning), starts by computing IMPdeg for most samples. For each sample in a majority class, it identifies the set S_Govermin using Equation (8). The algorithm calculates how many samples must be removed (delnum |c_j-*m*). If S_Govermin has enough samples, the algorithm removes delnum samples; otherwise, it removes the remaining samples from S_overmaj, ensuring the majority class is reduced to the mean size. The rest are kept in *D′*.

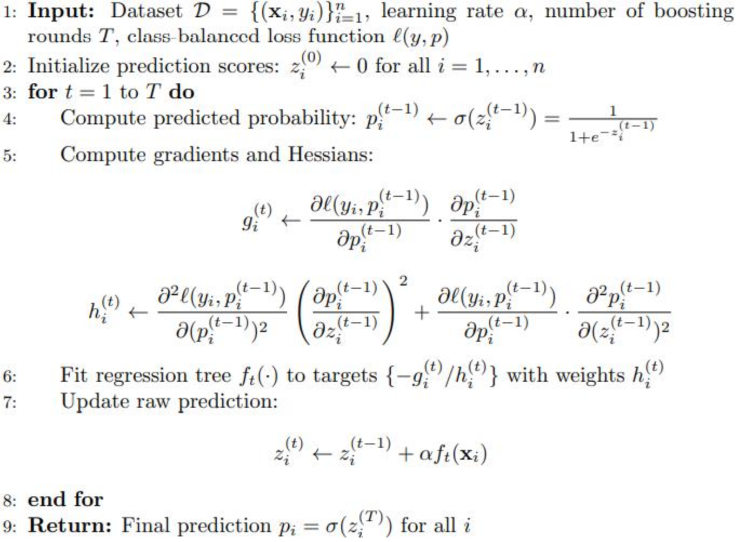
$$S_{Govermin} = \left\{x_j | x_j \in x_{maj}^{c\_maj} \wedge IM_{Peg}(X_j) > avg(IM_{Peg}(x_{maj})))\right\} \quad (8)$$

The third cleaning strategy, GDHS_BA (Balance-aware), also begins by computing IMPdeg(x) for all majority samples and identifying S_overmin using Equation (6). Like GDHS_GL, the algorithm calculates how many samples to delete and removes them from S_overmin and, if needed, from S_overmaj to balance the class size to *m*.

Finally, after processing all minority and majority classes with the selected oversampling and cleaning strategies, the algorithm returns the final balanced dataset *D′*, ready for use in training a classifier with reduced risk of bias toward the majority class.

2.2. Gradient Boosting Decision Trees

The pseudocode of GBDT is as follows.



```
Input: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, learning rate $\alpha$, number of boosting rounds $T$, class-balanced loss function $\ell(y, p)$
Initialize prediction scores: $z_i^{(0)} \leftarrow 0$ for all $i = 1, \ldots, n$
for $t = 1$ to $T$ do
    Compute predicted probability: $p_i^{(t-1)} \leftarrow \sigma(z_i^{(t-1)}) = \frac{1}{1+e^{-z_i^{(t-1)}}}$
    Compute gradients and Hessians:
        $g_i^{(t)} \leftarrow \frac{\partial \ell(y_i, p_i^{(t-1)})}{\partial p_i^{(t-1)}} \cdot \frac{\partial p_i^{(t-1)}}{\partial z_i^{(t-1)}}$
        $h_i^{(t)} \leftarrow \frac{\partial^2 \ell(y_i, p_i^{(t-1)})}{\partial (p_i^{(t-1)})^2} \left(\frac{\partial p_i^{(t-1)}}{\partial z_i^{(t-1)}}\right)^2 + \frac{\partial \ell(y_i, p_i^{(t-1)})}{\partial p_i^{(t-1)}} \cdot \frac{\partial^2 p_i^{(t-1)}}{\partial (z_i^{(t-1)})^2}$
    Fit regression tree $f_t(\cdot)$ to targets $\{-g_i^{(t)}/h_i^{(t)}\}$ with weights $h_i^{(t)}$
    Update raw prediction:
        $z_i^{(t)} \leftarrow z_i^{(t-1)} + \alpha f_t(\mathbf{x}_i)$
end for
Return: Final prediction $p_i = \sigma(z_i^{(T)})$ for all $i$
```

2.3. Combination of GDHS and GBDT

The pseudocode of the combination of GDHS and GBDT is as follows.

```
// Step 1: GDHS Preprocessing
Calculate mean class size m from D using Eq. (1)
for each minority class c_i in X_min do
    for each sample x in c_i do
        Compute safe_factor(x) using Eq. (2)
        Compute gen_factor(x) using Eq. (3)
        Compute select_weight(x) using Eq. (4)
    end for
    syn_num ← m - |c_i|
    while syn_num > 0 do
        Generate synthetic sample using weighted interpolation
        syn_num ← syn_num - 1
    end while
    Add c_i (with synthetic samples) to D'
end for

for each majority class c_j in X_maj do
    for each sample x in c_j do
        Compute IMPdeg(x) using Eq. (5)
        Identify S_overmin and S_overmaj using Eq. (6-7)
    end for
    Perform cleaning (LC, GL, or BA strategy) to reduce |c_j| to m
    Add retained c_j samples to D'
end for

// Step 2: GBDT Classification on D'
Initialize z_i ← 0 for all samples in D'
for t = 1 to T do
    Compute predicted prob: p_i ← sigmoid(z_i)
    Compute gradient g_i and Hessian h_i using class-balanced loss
    Fit regression tree f_t to -g_i / h_i with weight h_i
    Update: z_i ← z_i + alpha · f_t(x_i)
end for

Compute final prediction:  p_i ← sigmoid(z_i)
Return p_i as the predicted probability for each x_i
```

2.3. Classifier Performance

Classifier Performance will be measured using the Matthews Correlation Coefficient (MCC), Precision, Recall, and F-Value. This classifier performance measurement is based on the confusion matrix, as shown in Table 1[26].

Table 1. Confusion Matrix

	Predictive Positive Class	**Predictive Negative Class**
Actual Positive Class	True Positive (TP)	False Negative (FN)
Actual Negative Class	False Positive (FP)	True Negative (TN)

The following Equation can be seen in the Matthews Correlation Coefficient (MCC), Precision, Recall, and F-Value calculations [27].

$$MCC = \frac{TP\ x\ TN - FP\ x\ FN}{\sqrt{(TN\ x\ FN)(TN\ x\ FP)(TN\ x\ FN)(TP\ x\ FP)}} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = TP \quad (11)$$

$$F - Value = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \quad (12)$$

3. Results and Discussion

3.1. Dataset

The dataset used in this study can be seen in Table 1.

Table 2. Dataset

Dataset	#Ex	#Atts	Distribution of Class	IR
Contraceptive	1473	9	629/333/511	1.89
Flare	1066	11	147/211/239/95/43/331	7.70
Car Evaluation	1728	6	384/69/1210/65	18.62
Thyroid Disease	720	21	17/37/666	39.18
Red Wine Quality	1599	11	10/53/681/638/199/18	68.10
Page-Blocks	5473	10	4913/329/28/88/115	188.72

Table 2 shows six commonly used datasets in machine learning, each with different sizes, features, and class distributions. Some datasets, like Page-Blocks and Red Wine Quality, have a high class imbalance (indicated by high IR values), while others, like Contraceptive, are more balanced. These datasets are often used to test algorithms' performance, especially in handling imbalanced data.

3.2. Testing for Classifier Performance

The results can be seen in Table 3.

Table 3. Classifier Performance

	GDHS+GDBT	SMOTE+XGBoost	CatBoost	Select-SMOTE+LightGBM
		Dataset Contraceptive		
MCC	**0.983**	0.910	0.921	0.965
Precision	**0.872**	0.791	0.803	0.802
Recall	0.875	0.825	0.817	**0.881**
F-Value	**0.881**	0.818	0.876	0.867
		Dataset Flare		
MCC	**0.967**	0.956	0.961	0.949
Precision	0.817	**0.856**	0.727	0.815
Recall	**0.854**	0.835	0.841	0.819
F-Value	0.873	0.816	0.849	**0.881**
		Dataset Car Evaluation		
MCC	**0.985**	0.912	0.879	0.926
Precision	0.879	0.865	0.874	**0.891**
Recall	0.865	**0.876**	0.861	0.875
F-Value	**0.867**	0.815	0.786	0.811
		Dataset Thyroid Disease		
MCC	**0.876**	0.816	0.798	0.817
Precision	**0.901**	0.814	0.807	0.897
Recall	0.859	0.845	**0.871**	0.819
F-Value	0.876	**0.881**	0.863	0.814
		Dataset Red Wine Quality		
MCC	**0.854**	0.813	0.703	0.811
Precision	**0.865**	0.854	0.711	0.813
Recall	**0.834**	0.811	0.787	0.838
F-Value	**0.871**	0.866	0.765	0.798
		Dataset Page-Blocks		
MCC	**0.818**	0.809	0.783	0.798
Precision	**0.816**	0.798	0.789	0.811
Recall	**0.789**	0.788	0.709	0.765
F-Value	**0.821**	0.719	0.799	0.789

The results above compare the performance of four classification method combinations across six different datasets using MCC, Precision, Recall, and F-Value metrics. Overall, the GDHS+GDBT method consistently outperforms the others across most datasets, especially regarding MCC and F-Value, indicating strong classification capability even on datasets with high class imbalance. For example, the Contraceptive dataset achieves an MCC of 0.983 and an F-Value of 0.881. The SMOTE+XGBoost and Select-SMOTE+LightGBM methods also perform well but slightly trail behind GDHS+GDBT, particularly on datasets like Thyroid and Red Wine. CatBoost shows competitive performance, especially regarding Recall on datasets like Flare and Thyroid, but generally has lower MCC values than the other methods. GDHS+GDBT is the most robust and accurate method across the datasets, making it a strong choice for handling imbalanced data classification tasks.

The results can be seen in Figure 1.

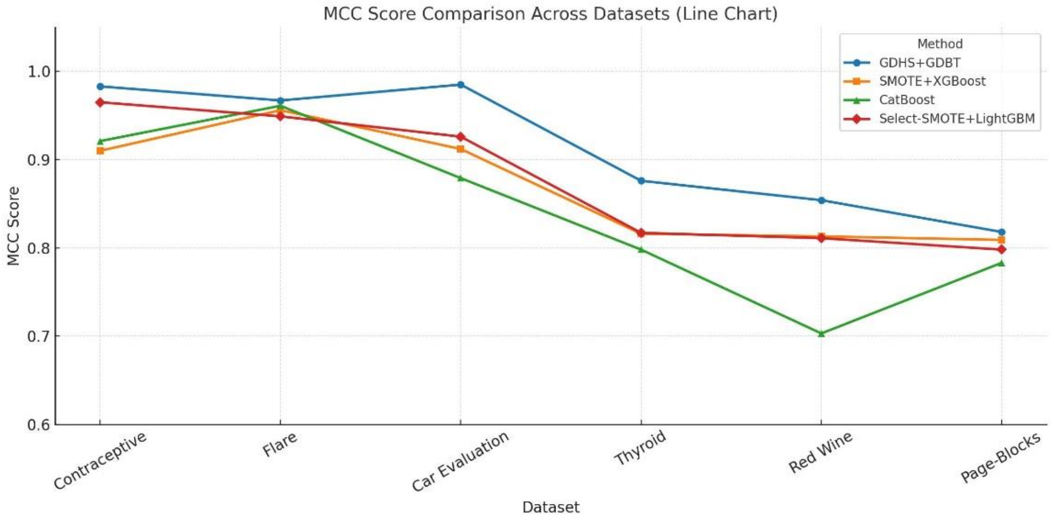


Fig 1. Classifier Performance

3.2. Discussion

The experimental results show that the GDHS+GDBT method consistently achieves the highest MCC values across most datasets, particularly those with high Imbalance Ratio (IR), such as Car Evaluation (IR = 18.62), Thyroid Disease (IR = 39.18), Red Wine Quality (IR = 68.10), and Page-Blocks (IR = 188.72). This indicates that GDHS+GDBT is more robust and effective in handling imbalanced data than other methods. The superior performance can be attributed to the hybrid design of GDHS (which enhances sample distribution) combined with GDBT's strong generalization ability, leading to improved minority class recognition. Therefore, GDHS+GDBT is particularly well-suited for classification tasks involving datasets with severe class imbalance.

4. Conclusion

This study confirms that combining the Generalization potential and learning Difficulty-based Hybrid Sampling (GDHS) with Gradient Boosting Decision Tree (GBDT) yields a highly effective classification framework for multiclass imbalanced datasets. Through extensive experiments on six benchmark datasets, the GDHS+GBDT approach consistently outperformed other baseline method, including SMOTE+XGBoost, CatBoost, and Select-SMOTE+LightGBM, across key performance metrics such as Matthews Correlation Coefficient (MCC), Precision, Recall, and F-Value. A particularly notable advantage of GDHS+GBDT is its robustness in handling datasets with high Imbalance Ratios (IR), such as Red Wine Quality (IR = 68.10) and Page-Blocks (IR = 188.72). In these challenging conditions, most conventional classifiers tend to be biased toward the majority class, resulting in poor recognition of minority classes. However, the GDHS sampling method intelligently amplifies the learning potential of minority class instances while simultaneously reducing overlap and noise from majority classes. When combined with GBDT's adaptive learning capabilities, this approach significantly improves minority class recall without compromising overall accuracy. The experimental results suggest that GDHS+GBDT is particularly suitable for real-world classification tasks where severe class imbalance and performance on minority instances are critical. Future work may explore the extension of this hybrid approach to real-time and high-dimensional data and integration with deep learning models for enhanced generalization.

Acknowledgement

The authors would like to express their deepest gratitude to the Rector of Universitas Medan Area for the continuous support, encouragement, and research facilities provided throughout this study. We also sincerely thank the Ministry of Higher Education, Science, and Technology for the funding and support that made this research possible.

References

[1] P. Gupta, A. Varshney, M. R. Khan, R. Ahmed, M. Shuaib, and S. Alam, “Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques,” *Procedia Computer Science*, vol. 218, pp. 2575–2584, Jan. 2023, doi: 10.1016/j.procs.2023.01.231.

[2] Y.-C. Wang and C.-H. Cheng, “A multiple combined method for rebalancing medical data with class imbalances,” *Computers in Biology and Medicine*, vol. 134, p. 104527, Jul. 2021, doi: 10.1016/j.compbiomed.2021.104527.

[3] B. Alabduallah *et al.*, “Class imbalanced data handling with cyberattack classification using Hybrid Salp Swarm Algorithm with deep learning approach,” *Alexandria Engineering Journal*, vol. 106, pp. 654–663, Nov. 2024, doi: 10.1016/j.aej.2024.08.061.

[4] M. Błaszczyk and J. Jędrzejowicz, “Framework for imbalanced data classification,” *Procedia Computer Science*, vol. 192, pp. 3477–3486, Jan. 2021, doi: 10.1016/j.procs.2021.09.121.

[5] M. Lango and J. Stefanowski, “What makes multiclass imbalanced problems difficult? An experimental study,” *Expert Systems with Applications*, vol. 199, p. 116962, Aug. 2022, doi: 10.1016/j.eswa.2022.116962.

[6] M. S. Santos, P. H. Abreu, N. Japkowicz, A. Fernández, and J. Santos, “A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research,” *Information Fusion*, vol. 89, pp. 228–253, Jan. 2023, doi: 10.1016/j.inffus.2022.08.017.

[7] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, “A survey on imbalanced learning: latest research, applications and future directions,” *Artif Intell Rev*, vol. 57, no. 6, p. 137, May 2024, doi: 10.1007/s10462-024-10759-6.

[8] Q. Li, Y. Song, J. Zhang, and V. S. Sheng, “Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering,” *Expert Systems with Applications*, vol. 147, p. 113152, Jun. 2020, doi: 10.1016/j.eswa.2019.113152.

[9] F. Grina, Z. Elouedi, and E. Lefevre, “Resampling of multiclass imbalanced data using belief function theory and ensemble learning,” *International Journal of Approximate Reasoning*, vol. 156, pp. 1–15, May 2023, doi: 10.1016/j.ijar.2023.02.006.

[10] H. Hartono, Y. Risyani, E. Ongko, and D. Abdullah, “HAR-MI method for multiclass imbalanced datasets,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, Art. no. 2, Apr. 2020, doi: 10.12928/telkomnika.v18i2.14818.

[11] H. Hartono and E. Ongko, “Combining Hybrid Approach Redefinition-Multiclass Imbalance (HAR-MI) and Hybrid Sampling in Handling Multiclass Imbalance and Overlapping,” *JOIV : International Journal on Informatics Visualization*, vol. 5, no. 1, pp. 22–26, Mar. 2021, doi: 10.30630/joiv.5.1.420.

[12] S. Nouas, L. Oukid, and F. Boumahdi, “Enhancing imbalanced text classification: an overlap-based refinement approach,” *Data Science and Management*, Mar. 2025, doi: 10.1016/j.dsm.2025.03.001.

[13] A. Jiménez-Macías, P. J. Muñoz-Merino, P. M. Moreno-Marcos, and C. D. Kloos, “Evaluation of traditional machine learning algorithms for featuring educational exercises,” *Appl Intell*, vol. 55, no. 7, p. 501, Mar. 2025, doi: 10.1007/s10489-025-06386-5.

[14] J. Yun and J.-S. Lee, “Learning from class-imbalanced data using misclassification-focusing generative adversarial networks,” *Expert Systems with Applications*, vol. 240, p. 122288, Apr. 2024, doi: 10.1016/j.eswa.2023.122288.

[15] J. Chen, C. Chen, W. Huang, J. Zhang, K. Debattista, and J. Han, “Dynamic contrastive learning guided by class confidence and confusion degree for medical image segmentation,” *Pattern Recognition*, vol. 145, p. 109881, Jan. 2024, doi: 10.1016/j.patcog.2023.109881.

[16] P. Thölke *et al.*, “Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data,” *NeuroImage*, vol. 277, p. 120253, Aug. 2023, doi: 10.1016/j.neuroimage.2023.120253.

[17] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, “A survey on imbalanced learning: latest research, applications and future directions,” *Artif Intell Rev*, vol. 57, no. 6, p. 137, May 2024, doi: 10.1007/s10462-024-10759-6.

[18] Y. Yan, Y. Lv, S. Han, C. Yu, and P. Zhou, “GDHS: An efficient hybrid sampling method for multiclass imbalanced data classification,” *Neurocomputing*, vol. 637, p. 130088, Jul. 2025, doi: 10.1016/j.neucom.2025.130088.

[19] Y. Yan, Y. Jiang, Z. Zheng, C. Yu, Y. Zhang, and Y. Zhang, “LDAS: Local density-based adaptive sampling for imbalanced data classification,” *Expert Systems with Applications*, vol. 191, p. 116213, Apr. 2022, doi: 10.1016/j.eswa.2021.116213.

[20] B. Krawczyk, M. Koziarski, and M. Woźniak, “Radial-Based Oversampling for Multiclass Imbalanced Data Classification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 2818–2831, Aug. 2020, doi: 10.1109/TNNLS.2019.2913673.

[21] M. Koziarski, M. Woźniak, and B. Krawczyk, “Combined Cleaning and Resampling algorithm for multiclass imbalanced data with label noise,” *Knowledge-Based Systems*, vol. 204, p. 106223, Sep. 2020, doi: 10.1016/j.knosys.2020.106223.

[22] J. Luo, Y. Yuan, and S. Xu, “Improving GBDT performance on imbalanced datasets: An empirical study of class-balanced loss functions,” *Neurocomputing*, vol. 634, p. 129896, Jun. 2025, doi: 10.1016/j.neucom.2025.129896.

[23] L. Han *et al.*, “An explainable XGBoost model improved by SMOTE-ENN technique for maize lodging detection based on multi-source unmanned aerial vehicle images,” *Computers and Electronics in Agriculture*, vol. 194, p. 106804, Mar. 2022, doi: 10.1016/j.compag.2022.106804.

[24] S. B. Jabeur, C. Gharib, S. Mefteh-Wali, and W. B. Arfi, “CatBoost model and artificial intelligence techniques for corporate failure prediction,” *Technological Forecasting and Social Change*, vol. 166, p. 120658, May 2021, doi: 10.1016/j.techfore.2021.120658.

[25] C. Zhao, Z. Yan, X. Sun, and M. Wu, “Enhancing aspect category detection in imbalanced online reviews: An integrated approach using Select-SMOTE and LightGBM,” *International Journal of Intelligent Networks*, vol. 5, pp. 364–372, Jan. 2024, doi: 10.1016/j.ijin.2024.10.002.

[26] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” *Pattern Recognition*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.

[27] Z. Xu, D. Shen, T. Nie, and Y. Kou, “A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data,” *Journal of Biomedical Informatics*, vol. 107, p. 103465, Jul. 2020, doi: 10.1016/j.jbi.2020.103465.