HHO-SMOTe: Efficient Sampling Rate for Synthetic Minority Oversampling Technique Based on Harris Hawk Optimization

Khaled SH. Raslan, Almohammady S. Alsharkawy, K. R. Raslan
Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt

***Abstract*—Classifying imbalanced datasets presents a significant challenge in the field of machine learning, especially with big data, where instances are unevenly distributed among classes, leading to class imbalance issues that affect classifier performance. Synthetic Minority Over-sampling Technique (SMOTE) is an effective oversampling method that addresses this by generating new instances for the under-represented minority class. However, SMOTE's efficiency relies on the sampling rate for minority class instances, making optimal sampling rates crucial for solving class imbalance. In this paper, we introduce HHO-SMOTe, a novel hybrid approach that combines the Harris Hawk optimization (HHO) search algorithm with SMOTE to enhance classification accuracy by determining optimal sample rates for each dataset. We conducted extensive experiments across diverse datasets to comprehensively evaluate our binary classification model. The results demonstrated our model's exceptional performance, with an AUC score exceeding 0.96, a high G-means score of 0.95 highlighting its robustness, and an outstanding F1-score consistently exceeding 0.99. These findings collectively establish our proposed approach as a formidable contender in the domain of binary classification models.**

Keywords—Imbalanced data; machine learning; over-sampling; SMOTE; HHO

I. Introduction

The applications of Machine Learning (ML) have seen a growing trend in classification domains involving data for automating processes. However, the process of training presents difficulties due to inherent nature of algorithms, which typically learn from datasets with balanced distributions [1]. As a result, acquiring knowledge from datasets with uneven distributions can lead to reduced accuracy and dependability in the resulting model. This phenomenon is termed "imbalance" or "unbalance" [2].

In contemporary applications, addressing challenges posed by imbalanced data has emerged as a notable issue. This issue is particularly evident in various domains such as the detection of fraud telephone calls [3], text classification [4], and biomedical data analysis [5, 6]. The classification of imbalanced data stands as a significant concern within the realms of machine learning and data mining [7]. In the context of imbalanced datasets, a notable discrepancy exists, with one class containing notably fewer training instances (Minority class) than the other (Majority class). In dealing with imbalanced datasets, conventional machine learning and classification algorithms frequently exhibit a tendency to achieve very high accuracy rates in classifying the majority class, while attaining notably lower accuracy rates when classifying the minority class [8]. Therefore, the classifier's effectiveness suffers when it comes to diagnosing samples from the minority class. Consequently, the classification of imbalanced datasets presents a substantial hurdle in the realm of classification research. Conversely, in numerous practical scenarios, the emphasis is placed on recognizing minority class samples rather than their majority counterparts [9].

In this paper, we emphasize the critical nature of class imbalance and its adverse consequences on the performance of traditional classifiers in real-world applications, such as medical diagnosis, fraud detection, and anomaly detection. To overcome these problem and shortage, we present a unique hybrid binary classification method that integrates multiple algorithms, enhancing the overall robustness of the approach. The core of our methodology lies in the utilization of the Harris Hawk optimization search algorithm, which facilitates the calculation of optimal sample rates for each minority class, resulting in improved representation within the data set. By strategically adapting the SMOTE technique with Harris Hawk Search, we ensure more effective synthetic data generation, tailored to capture the specific characteristics of the imbalance data.

The SMOTE has emerged as a contender for effectively addressing the classification of imbalanced datasets [10]. This technique operates by generating new instances for the under-represented minority class, effectively re-balancing the dataset by augmenting the presence of minority class data points using SMOTE framework. These algorithms adopt a uniform sampling rate for all instances. Unfortunately, this uniform approach leads to suboptimal performance outcomes. This limitation becomes particularly pronounced when the dataset presents varying degrees of difficulty across different instances of the minority class. Instances that are inherently harder to classify may benefit from a different sampling strategy compared to instances that are relatively easier to classify. This nuanced variation is often not accounted for by the uniform sampling rate strategy, resulting in missed opportunities to improve the overall performance of the classification model. As a result, there exists a need for more sophisticated techniques that can deceptively adjust the sampling rates based on the inherent complexities within the minority class instances. By doing so, the resulting classification model could achieve more accurate and refined outcomes, effectively mitigating the limitations imposed by the current SMOTE-based methodologies.

Within our paper, we propose an innovative algorithm that builds upon the foundation of the SMOTE technique while incorporating the HHO [11] to enhance the efficacy of imbalanced data classification. The integration of the HHO algorithm introduces a dynamic approach wherein diverse sampling rates are generated for individual instances of the minority class. This process culminates in the identification of an optimal combination of these sampling rates. Subsequently, this amalgamation of optimal sampling rates is formulated and seamlessly integrated into the SMOTE Algorithm. The quest for these optimal sampling rates is executed with a high degree of intelligence, ensuring an insightful search process. Once these optimal rates are successfully pinpointed, over-sampling is carried out exclusively on the instances belonging to the minority class, with each instance benefiting from its corresponding optimal sampling rate.

The subsequent sections of this paper are structured as follows: In Section II introducing an overview of current methodologies utilized for handling imbalanced datasets. Section III describes the SMOTE technique and the HHO algorithm in some detail. Section IV delves into the intricacies of our novel HHO-SMOTe algorithm, presenting a detailed account of its design and functionality, Section V guides you through a comprehensive examination of outcomes, encompassing diverse datasets and a variety of algorithms. Section VI concludes this paper.

II. Related Work

A lot of research papers [2, 12, 13] have create a comprehensive examination of imbalanced datasets. These studies have not only conducted reviews but have also put forth various solutions aimed at effectively addressing the challenge of imbalanced data. Their objective is to determine the most optimal approach that exhibits superior performance in handling this issue. Ebenuwa et al. [12] introduced a feature selection approach for handling imbalanced datasets. They outlined the methodology and implementation steps, evaluating its effectiveness using machine learning algorithms like decision trees, logistic regression, and support vector machines. Their study aimed to identify the algorithm most suitable for addressing imbalanced data challenges through this ensemble of classifiers. The approach proposed in [13] involves the incorporation of an oversampling technique that meticulously incorporates all minority samples during the classification process within the training data. The Study conducted a comprehensive evaluation of this technique by comparing its performance against state-of-the-art ensemble learning methods. The objective behind this assessment was to ascertain the prowess of the oversampling technique in addressing imbalanced data scenarios.

Liu et al. [14] proposed advanced EasyEnsemble and BalanceCascade algorithms to address class imbalance issues more effectively than existing methods. Their research revealed that both algorithms outperformed established techniques, demonstrating their efficiency in tackling class imbalance challenges. Additionally, the authors in [15] devised the GASMOTE algorithm, which introduces a novel approach of employing distinct sampling rates tailored to individual instances within minority classes. This algorithm intelligently identifies the optimal combination of these sampling rates. Empirical evaluations performed on ten prototypical imbalanced datasets unveiled compelling outcomes. When juxtaposed against the SMOTE algorithm, GASMOTE exhibited an impressive enhancement. The empirical results derived from this application validate the GASMOTE algorithm's precision.

Nnamoko and Korkontzelos in [16] have taken strides in the realm of diabetes prediction by devising an optimized iteration of the SMOTE technique. This advanced algorithm integrates the InterQuartile Range technique to strategically oversample dispersed or extreme data prior to the application of SMOTE. This pre-processing step contributes significantly to enhancing the distribution of training samples, ultimately bolstering the efficacy of the diabetes prediction model. Liu st al. [17] brought forth a pioneering contribution in the arena of data balance within the context of spam detection. They proposed a sophisticated algorithm termed Fuzzy-based OverSampling, which revolves around the utilization of fuzzy logic principles to carefully harmonize the data distribution in synthetic sampling endeavors. This innovative methodology exhibited its prowess in not only rectifying the class imbalance but also in fine-tuning the distribution to be more representative of the real-world scenario. Notably, this enhancement manifested in elevated precision levels across a diverse array of ensemble learning models employed for the spam detection task.

The authors in [18] undertook a significant enhancement of the SL-SMOTE technique by incorporating an evolutionary optimization procedure to fine-tune its algorithmic parameters. This evolved rendition, aptly labeled Evolutionary SL-SMOTE, attained exemplary performance metrics when evaluated in the context of seminal quality prediction using AdaBoost. In the research conducted by Susan and Kumar [19], a comprehensive survey was undertaken to delve into the realm of preprocessing techniques within the domain of machine learning applications. The scholarly paper in question provides an in-depth exploration of various sampling methodologies, delving into the intricacies of how each of the scrutinized works tactically incorporated the suggested remedies. The culmination of this survey encompasses a thorough summary of the experimental protocols employed, encompassing intricate procedural insights as well as the comprehensive compilation of the outcomes that were documented.

To address more effectively the issue of how to determine the proper sample rate of the minority instances involved in the synthesis to avoid the generated minority instances decreasing the learning efficiency of the classification process, in this paper, we propose HHO-SMOTe which is also an improved variant of SMOTE based on a novel nature inspired algorithm call HHO. Nevertheless, HHO-SMOTe emphasis on determine the appropriate minority instances which increase the accuracy of the classification algorithmic.

III. Preliminaries

The SMOTE and exploratory and exploitative stages of the Harris Hawk Optimization algorithm are covered in this section. We explained the different procedures and steps used by each algorithm. In addition, we demonstrate how these various phases have been used to develop a novel algorithm. Due to the integration of the two algorithms, our method can dynamically adapt to a variety of datasets and consistently produce the best results with a high degree of efficiency.

A. SMOTE

SMOTE is commonly used when dealing with imbalanced datasets, where one class (minority class) has significantly fewer examples than the other class (majority class). In such cases, machine learning models may struggle to correctly classify the minority class because they tend to be biased towards the majority class. SMOTE helps address this imbalance by generating synthetic examples of the minority class to create a more balanced dataset for training.

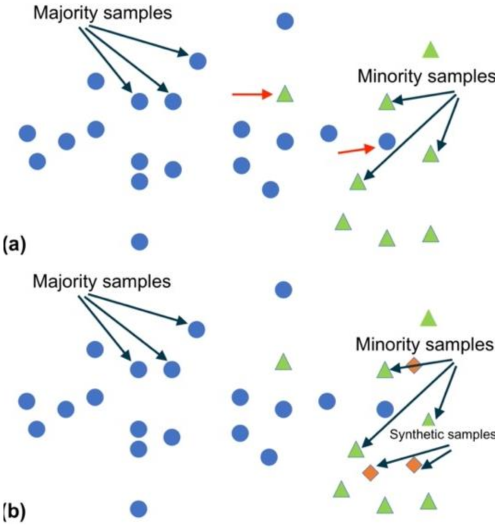

Fig. 1. The principle of the SMOTE.

We can observe an example of an imbalanced dataset in Fig. 1(a) above. Here, the majority class is represented by circular shapes, which stand in for the data's predominant occurrences, while the minority class is represented by triangular shapes, signifying the smaller number of data samples. Some examples from the minority and majority classes are in areas that do not naturally align with the opposite class, most notably with the red arrow. The SMOTE algorithm initiates the process of selecting synthetic samples, a crucial step in bolstering the minority class. The sampling rate specified for each category of occurrences serves as the basis for this selection process. The synthetic samples are presented as square forms in Fig. 1(b). Upon applying the SMOTE technique, the resultant effect is a reduction in the disparity between the Minority and Majority classes.

The SMOTE algorithm includes a sample rate parameter to control the extent of over-sampling. The sample rate determines how many synthetic examples are generated for each minority class instance. Here's an equation that includes the sample rate in the SMOTE algorithm:

$$C = A + s * (B - A) \quad (1)$$

where:

- C is the synthetic example being generated.
- A is a randomly selected instance from the minority class.
- B is one of the k nearest neighbors of A (also from the minority class) and is randomly chosen.
- s is the sample rate parameter, which influences how many synthetic examples are generated between A and B. The s parameter is a value between *0* and *1*, which allows you to control the density of synthetic examples to be generated. When s = *0.5*, one synthetic example is generated exactly. This can be seen as an average or balanced interpolation between the two instances. If s is less than *0.5*, the synthetic examples will be closer to A than B, otherwise the synthetic examples will be closer to B than A.

Impact of sample rate to balance Dataset:

The choice of s influences how many synthetic instances are generated and how they are distributed between A and B. By adjusting s, you can fine-tune the balance of your dataset. A smaller s may be suitable if you want a moderate increase in the minority class, while a larger s will result in a more substantial over-sampling. As Addressed class imbalance in datasets using the SMOTE algorithm is a common strategy in machine learning, but selecting the appropriate sample rate presents a challenging task. There are no universal guidelines for determining the ideal sample rate, as it hinges on various factors like dataset characteristics, machine learning algorithms, and problem-specific nuances. The primary goal of SMOTE is to balance class distribution, vital for training fair and effective models. However, selecting the wrong sample rate can lead to overfitting, underfitting, or suboptimal model performance.

Researchers in [20-23] often use SMOTE approaches to balance their datasets before staring work on the classification or feature selection, or cluster problems without working with the sample rate selection for the minority classes. Grid search involves trying out a range of predefined sample rates and selecting the one that optimizes evaluation metrics such as precision, recall, F1-score, or AUC. Cross-validation enhances this process by providing a more robust assessment across multiple data subsets. An iterative refinement process, where researchers gradually narrow down the optimal sample rate through experimentation and analysis, is common practice. Additionally, understanding the sensitivity of machine learning algorithms to different sample rates is crucial.

In summary, choosing the right sample rate in SMOTE is a nuanced decision that relies on empirical methods, domain expertise, and iterative exploration to strike the balance that suits the dataset and problem domain. We have put forth our solution for determining the most accurate sample rate, which will be applied when generating samples from the minority classes to achieve data set balance. This solution leverages the intelligence of the HHO algorithm, a sophisticated optimization technique.

B. Harris Hawks Optimizer (HHO)

The HHO has introduced by Ali Asghar Heidari in 2019, the HHO algorithm has garnered significant attention from the research community [11, 24]. HHO draws inspiration from the hunting behavior of Harris Hawks in nature, particularly their agile surprise pounce technique. Harris Hawks, known for their remarkable intelligence, exhibit various chasing styles based on different scenarios and the behavior of their prey. HHO is widely recognized as one of the most effective optimization algorithms, and it has been successfully applied to a variety of problems across different domains encompass energy and power flow analysis, engineering, medical applications, network optimization, and image processing. The comprehensive review [25-28] presents a survey of the existing body of work related to HHO.

Within this section, shows the modeling of both the exploratory and exploitative phases inherent in HHO methodology. The phases are done by three steps draw inspiration from the natural behaviors of Harris hawks, including their approaches to prey exploration, surprise pouncing, and the diverse attack strategies employed. HHO represents a population-based optimization approach devoid of gradients, rendering it adaptable to a wide array of optimization challenges, provided that they are appropriately formulated. The detailed explanations provided in the subsequent subsections.

1) Exploration phase: Hawks perch in specific locations and constantly monitor the surrounding environment to identify prey using two strategies, which are represented in Eq. (2). If $p < 0.5$, the hawks perch based on the position of the family members. If $p \geq 0.5$, the hawks perch in a random space within the population area.

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \mid X_{rand}(t) - 2r_2 X(t) \mid & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (2)$$

where $X(t + 1)$ is the position vector of hawks in the next iteration t, Xrabbit(t) is the position of rabbit, $X(t)$ is the current position vector of hawks, r_1, r_2, r_3, r_4, and q are random numbers inside *(0, 1)*, which are updated in each iteration, *LB* and *UB* show the upper and lower bounds of variables, *Xrand(t)* is a randomly selected hawk from the current population, and X_m is the average position of the current population of hawks.

The HHO utilized a simple model to generate random locations inside the group's home range *(LB, UB)*. The first rule generates solutions based on a random location and other hawks. In second rule of Eq. (2), we have the difference of the location of best so far and the average position of the group plus a randomly-scaled component based on range of variables, while r_3 is a scaling coefficient to further increase the random nature of rule once r_4 takes close values to 1 and similar distribution patterns may occur. Utilizing the simplest rule, which can mimic the behaviors of hawks. The average position of hawks is attained using Eq. (3):

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \quad (3)$$

where, $X_i(t)$ indicates the location of each hawk in iteration t and N denotes the total number of hawks.

2) Transition from exploration to exploitation: The HHO can transfer from exploration to exploitation and then, change between different exploitative behaviors based on the escaping energy of the prey. The energy of a prey decreases considerably during the escaping behavior. To model this fact, the energy of a prey is modeled as:

$$E = 2E_0(1 - \frac{t}{T}) \quad (4)$$

Where E indicates the escaping energy of the prey, T is the maximum number of iterations, and E_0 is the initial state of its energy. In HHO, E_0 randomly changes inside the interval $(-1, 1)$ at each iteration. When the value of E_0 decreases from 0 to -1, the rabbit is physically flagging, whilst when the value of E_0 increases from 0 to 1, it means that the rabbit is strengthening.

3) Exploitation phase: Which the hawks attack the targeted prey. Then, however, the prey tries to escape the attack. Based on hawk attacking behavior and escaping prey behavior, four scenarios will be described as below:

a) Soft Besiege: When $r \geq 0.5$ and $|E| \geq 0.5$, the rabbit still has enough energy and try to escape by some random misleading jumps but finally it cannot. During these attempts, the Harris' hawks encircle it softly to make the rabbit more exhausted and then perform the surprise pounce. This behavior is modeled by the following rules:

$$X(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \quad (5)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (6)$$

Where $\Delta X(t)$ is the difference between the position vector of the rabbit and the current location in iteration t, r_5 is a random number inside *(0, 1)*, and $J = 2(1 - r_5)$ represents the random jump strength of the rabbit throughout the escaping procedure. The J value changes randomly in each iteration to simulate the nature of rabbit motions.

b) Hard Besiege: When $r \geq 0.5$ *and* $|E| < 0.5$, the prey is so exhausted, and it has a low escaping energy. In addition, the Harris' hawks hardly encircle the intended prey to finally perform the surprise pounce. In this situation, the current positions are updated using:

$$X(t+1) = X_{rabbit}(t) - E \mid \Delta X(t) \mid \quad (7)$$

c) Soft Besiege with Progressive Rapid Dives: When still $|E| \geq 0.5$ *but* $r < 0.5$, the rabbit has enough energy to successfully escape and still a soft besiege is constructed before the surprise pounce. This procedure is more intelligent than the previous case, the final strategy for updating the positions of hawks in the soft besiege phase can be performed by:

$$X(t+1) = \begin{cases} Y \; if \; F(Y) < F(X(t)) \\ Z \; if \; F(Z) < F(X(t)) \end{cases} \quad (8)$$

where, Y and Z are obtained using Eq.9 and Eq.10. A simple illustration of this step for one hawk. Y is the hawks next move based on the following rule.

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X(t) \quad (9)$$

To mathematically model the escaping patterns of the prey and leapfrog movements (as called in [22]), the levy flight (LF) concept is utilized in the HHO algorithm. In HHO the hawks dive based on the LF-based patterns using the following rule:

$$Z = Y + S \times LF(D) \quad (10)$$

Where D is the dimension of problem and S is a random vector by size $1 \times D$ and LF is the levy flight function, which is calculated as follows.

$$LF(X) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\,\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}} \quad (11)$$

Where u, v are random values inside $(0, 1)$, β is a default constant set to 1.5.

d) Hard Besiege with Progressive Rapid Dives: When $|E| < 0.5$ *and* $r < 0.5$, the rabbit has not enough energy to escape and a hard besiege is constructed before the surprise pounce to catch and kill the prey. The situation of this step in the prey side is similar to that in the soft besiege, but this time, the hawks try to decrease the distance of their average location with the escaping prey. Therefore, the following rule is performed in hard besiege condition:

$$X(t+1) = \begin{cases} Y \; if \; F(Y) < F(X(t)) \\ Z \; if \; F(Z) < F(X(t)) \end{cases} \quad (12)$$

where Y and Z are obtained using rules in Eq. (13) and Eq. (14).

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X_m(t) \quad (13)$$

$$Z = Y + S \times LF(D) \quad (14)$$

IV. The Proposed HHO-SMOTe Algorithm

In this section, the proposed HHO-SMOTe approach is proposed for determining the efficient sample rate to be used in the SMOTE technique. The proposed HHO-SMOTe primary goal is to increase the accuracy of classification of the imbalanced datasets. We employed the HHO algorithm to find the optimum solution based on the KNN classification accuracy in order to get the best sampling rate of the synthetic minority class instances.

The proposed HHO-SMOTe initialized by determining its control parameters such as the population size N, the number of minority class instances n, and the maximum number of iterations. Then, the algorithm starts by generating a population X with the dimension $N \times n$ from the initial solution as an initial phase for the HHO-SMOTe approach. Each solution $x_i \in X$ represents a candidate sample rate for SMOTe and it is assessed by the value of dataset classification accuracy where the best sample rate (solution) has the highest classification accuracy based on KNN algorithm. The solution can be represented with a raw of n values, these values are 0 and the maximum number of samples for each minority class instance. The 0 value in the first position of x_i indicates that the current instance in the minority class have a sample rate 0 and will not be used in the generation of the synthetic data. Since, if the value is greater than 0, then the current minority class instance will be utilized in the generation of the synthetic data. For example, a solution x_i for generating a synthetic data which have 6 minority class instances can be represented as x_i = [1, 0, 2, 0, 3, 1]. This means that the sample rate to generate the synthetic date is 1 sample of the first minority class instance, 0 sample of the second minority class instance, two samples of the third minority class instance, and so on. The pseudocode of the HHO-SMOTe is showed in Algorithm 1.

Algorithm 1: Pseudo-code of HHO-SMOTe approach.

```
Inputs:
The population size $N$ and maximum number of iterations $T$
Outputs:
The location of rabbit and its fitness value Initialize the random population $X_i, i = 1, 2, ..., N$
while (stopping condition is not met) do
        Generate a synthetic data based on current sample rate (solution) using SMOTE alg., then calculate the fitness values of hawks using on KNN alg.
        Set $X_{rabbit}$ as the location of rabbit (highest accuracy)
        for (each hawk ($X_i$)) do
          Update the initial energy $E_0$ and jump strength J→ $E_0$ = 2rand() - 1, J = 2(1-rand())
          Update the $E$ using Eq. (4)
          if (|E| ≥ 1) then (Exploration phase)
                Update the location vector using Eq. (2)
          if (|E| < 1) then (Exploitation phase)
                if (r ≥ 0.5 and |E| ≥ 0.5) then (Soft besiege)
                  Update the location vector using Eq. (5)
                else if (r ≥ 0.5 and |E| < 0.5) then (Hard besiege)
                  Update the location vector using Eq. (7)
                else if (r < 0.5 and |E| ≥ 0.5) then (Soft besiege with progressive rapid dives)
                  Update the location vector using Eq. (8)
                else if (r < 0.5 and |E| < 0.5) then (Hard besiege with progressive rapid dives)
                  Update the location vector using Eq. (12)
Return Xrabbit
```

A. Performance Evaluation Measures

Performance evaluation metrics are critical for evaluating classification performance and guiding classifier design. In this step, the confusion matrix was used to get the results of the proposed HHO-SMOTe approach and to make the comparison between all the used SMOTE approaches. The confusion matrix Fig. 2 describes the performance of the classification models. True positive (TP): Observation is predicted positive and is actually positive. False positive (FP): Observation is predicted positive and is actually negative. True negative (TN): Observation is predicted negative and is actually negative. False negative (FN): Observation is predicted negative and is actually positive. From the confusion matrix, we can conclude the following measures:

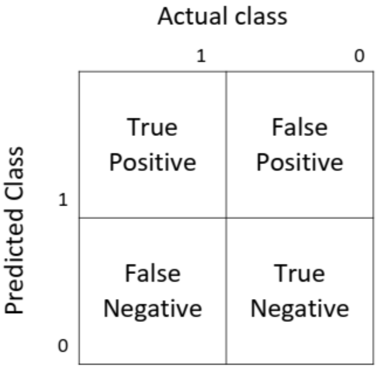

Fig. 2. Confusion matrix for the two-class classification problem.

1) G-mean: The geometric mean is the root of the product of class-wise sensitivity. This measure tries to maximize the accuracy on each of the classes while keeping these accuracies balanced. For binary classification G-mean is the squared root of the product of the sensitivity and specificity. For multi-class problems it is a higher root of the product of sensitivity for each class.

$$G - mean = \sqrt{Sensitivity \times Specificity} \quad (15)$$

2) F1 score: The $F1$ score, F score, or F measure is the harmonic mean of precision and sensitivity it gives importance to both factors:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \quad (16)$$

3) AUC: The receiver operating characteristics (ROC) curve is the plot between sensitivity and the FP rate for various threshold values. The area under curve (AUC) is the area under this ROC curve; it is used to measure the quality of a classification model. The larger the area, the better the performance. The ROC curve is a two-dimensional coordinate graph in which the X-axis represents the false positive rate (FPR) and Y-axis represent the true positive rate (TPR). The AUC can be calculated as:

$$AUC = \frac{1 + TPR - FPR}{2} \quad (17)$$

V. Experiments and Evaluation

In this section, the experiments were done on different datasets. The following subsections will demonstrate the results and analyze these results. The experiments were conducted on Google Colaboratory, which provides a free Jupyter notebook environment with GPU support for running machine learning experiments [29].

TABLE I. Summary Description of Imbalanced Datasets

Dataset	#Att.	Org dataset	#Min.	#Maj.	IR
abalone9-18	8	731	42	689	16.40
ada	47	4147	1029	3118	3.03
cleveland-0	14	177	13	164	12.62
ecoli2	7	336	52	284	5.46
ecoli3	7	336	35	301	8.60
ecoli4	7	336	20	316	15.80
german	29	1000	300	700	2.33
glass0	9	214	70	144	2.06
glass1	9	214	76	138	1.82
glass2	9	214	17	197	11.59
habarman	3	306	81	225	2.78
hypothyroid	25	3163	151	3012	19.95
kc1	20	2109	326	1783	5.47
new-thyroid1	5	215	35	180	5.14
page-blocks0	10	5472	559	4913	8.79
pc1	21	1109	77	1032	13.40
Pima	8	768	268	500	1.87
vehicle0	18	846	199	647	3.25
vehicle1	18	846	217	629	2.90
vehicle2	18	846	218	628	2.88
vehicle3	18	846	212	634	2.99
yeast3	10	1484	163	1321	8.10
yeast4	10	1484	51	1433	28.10
yeast5	10	1484	44	1440	32.73
yeast6	10	1484	35	1449	41.40

In our research, we utilized over 25 diverse datasets in different industries and attributes to evaluate the proposed technique. We maintained the original class distribution with five-fold cross-validation and conducted each experiment five times to obtain average metrics. Table I summarizes dataset details, including the dataset name, the number of attributes, the number of samples for the minority class, the original dataset record numbers, the number of samples in the majority class, and the corresponding imbalance ratio.

Table II presents the outcomes of our experimentation of 19 SMOTe variants approach and the proposed HHO-SMOTe approach with KNN algorithm as the application of SMOTE techniques for oversampling the dataset.The 19 methods are ADASYN [30], AND-SMOTE [31], ANS [32], Borderline-SMOTE1 [33], Borderline-SMOTE2 [33], distance-SMOTE [34], G-SMOTE [35], GASMOTE [15] , Gaussian-SMOTE [36], KernelADASYN [37], kmeans-SMOTE [38], Random-SMOTE [39], Safe-Level-SMOTE [40], SDSMOTE [41], SMOTE [10], SOMO [42], SVM-balance[43], SYMPROD [44], ASN-SMOTE [45]. Notably, we have highlighted in bold the distinctive optimal values achieved for the average G-mean, F1-score, and AUC within the KNN results. This highlighting underscores the noteworthy observation that the combination of HHO-SMOTe consistently yields optimal results across a diverse array of datasets. The classification performance comparison results for the selected seven approaches applied on twelve datasets presented in Fig. 3, 4, and 5 are obtained using data from Table II.

TABLE II. RESULTS OBTAINED BY KNN ON DATASETS OVERSAMPLED BY DIFFERENT SMOTE TECHNIQUES

Dataset	abalone9-18			ada			cleveland-0			ecoli2			ecoli3		
Method	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC
ADASYN	0.952	0.952	0.952	0.857	0.857	0.857	0.888	0.888	0.894	0.948	0.949	0.949	0.993	**0.994**	0.993
AND-SMOTE	0.969	0.969	0.969	0.866	0.866	0.866	0.95	0.949	0.95	0.931	0.93	0.932	0.988	0.988	0.988
ANS	0.961	0.961	0.961	0.856	0.856	0.856	0.979	0.981	0.982	0.923	0.923	0.923	0.971	0.971	0.971
Borderline-SMOTE1	0.972	0.975	0.975	0.864	0.864	0.864	0.969	0.97	0.969	0.967	0.968	0.967	**0.996**	0.994	**0.995**
Borderline-SMOTE2	0.963	0.964	0.963	0.874	0.874	0.874	0.928	0.929	0.93	0.968	0.968	0.968	0.989	0.988	0.989
distance-SMOTE	0.974	0.973	0.974	0.858	0.858	0.858	0.96	0.96	0.961	0.953	0.955	0.954	0.989	0.988	0.989
G-SMOTE	0.961	0.961	0.961	0.857	0.857	0.857	0.963	0.96	0.964	0.941	0.942	0.941	0.989	0.987	0.991
GASMOTE	0.634	0.650	0.668	0.503	0.446	0.507	0.850	0.845	0.851	0.900	0.894	0.900	0.494	0.375	0.510
Gaussian-SMOTE	0.91	0.912	0.914	0.716	0.722	0.73	0.96	0.96	0.96	0.926	0.929	0.927	0.983	0.982	0.983
KernelADASYN	0.954	0.954	0.955	0.856	0.857	0.857	0.972	0.97	0.973	0.975	0.974	0.975	0.976	0.977	0.976
kmeans-SMOTE	0.447	0.928	0.6	0.863	0.863	0.863	0.816	0.98	0.833	0.943	0.942	0.943	0.988	0.988	0.988
Random-SMOTE	0.948	0.947	0.948	0.854	0.854	0.854	0.939	0.939	0.941	0.955	0.955	0.955	0.982	0.982	0.982
Safe-Level-SMOTE	0.954	0.954	0.954	0.804	0.804	0.804	0.861	0.858	0.864	0.962	0.962	0.962	0.983	0.982	0.983
SDSMOTE	0.957	0.957	0.957	0.85	0.85	0.85	0.928	0.92	0.931	0.927	0.929	0.928	0.969	0.971	0.969
SMOTE	**0.976**	**0.976**	**0.976**	0.848	0.848	0.848	0.931	0.939	0.933	0.955	0.955	0.955	0.983	0.982	0.983
SOMO	0.258	0.91	0.533	0.699	0.811	0.72	0.707	0.957	0.75	0.893	0.94	0.896	0.955	0.971	0.955
SVM-balance	0.955	0.954	0.955	0.921	0.921	0.921	0.968	0.97	0.968	0.96	0.961	0.96	0.983	0.982	0.983
SYMPROD	0.973	0.973	0.973	0.853	0.852	0.853	0.707	0.978	0.75	0.944	0.942	0.944	0.988	0.988	0.988
ASN-SMOTE	0.717	0.46857	0.749	0.4782	0.20965	0.562	0.48	0.439	0.595	0.9128	0.79798	0.914	0.8965	0.62918	0.899
HHO-SMOTe	0.9383	0.93533	0.94	**0.941**	**0.934**	**0.935**	**0.981**	**0.99**	**0.989**	**0.987**	**0.983**	**0.985**	0.982	0.982	0.982

Dataset	ecoli4			german			glass0			glass1			glass2		
Method	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC
ADASYN	0.938	0.939	0.938	0.791	0.794	0.794	0.883	0.885	0.886	0.866	0.878	0.87	0.938	0.941	0.939
AND-SMOTE	0.961	0.961	0.961	0.803	0.804	0.809	0.943	0.943	0.943	0.844	0.843	0.844	0.916	0.916	0.916
ANS	0.938	0.939	0.938	0.799	0.8	0.8	0.931	0.931	0.931	**0.942**	**0.939**	**0.943**	0.967	0.966	0.967
Borderline-SMOTE1	0.961	0.961	0.961	0.793	0.793	0.793	0.932	0.931	0.932	0.85	0.855	0.851	0.908	0.908	0.909
Borderline-SMOTE2	0.961	0.961	0.961	0.781	0.781	0.781	0.936	0.931	0.937	0.859	0.856	0.859	0.966	0.965	0.966
distance-SMOTE	0.956	0.956	0.956	0.776	0.778	0.778	0.953	0.954	0.953	0.819	0.819	0.819	0.919	0.916	0.919
G-SMOTE	0.956	0.956	0.956	0.793	0.793	0.793	0.932	0.931	0.932	0.834	0.833	0.834	0.941	0.941	0.941
GASMOTE	0.790	0.785	0.811	0.547	0.527	0.555	0.852	0.865	0.855	0.796	0.793	0.800	0.515	0.458	0.568
Gaussian-SMOTE	0.93	0.928	0.931	0.688	0.701	0.716	0.886	0.885	0.887	0.814	0.818	0.817	0.786	0.793	0.806
KernelADASYN	0.944	0.945	0.945	0.45	0.626	0.531	0.892	0.886	0.894	0.905	0.904	0.905	0.874	0.874	0.874
kmeans-SMOTE	0.956	0.956	0.95	0.796	0.797	0.79	0.96	0.961	0.96	0.929	0.928	0.92	0.401	0.87	0.56

			6			7			3			9		6	6
Random-SMOTE	0.945	0.945	0.946	0.789	0.79	0.789	0.915	0.919	0.915	0.833	0.831	0.834	0.924	0.924	0.924
Safe-Level-SMOTE	0.927	0.928	0.927	0.688	0.688	0.688	**0.966**	**0.966**	**0.966**	0.83	0.831	0.83	0.958	0.958	0.958
SDSMOTE	0.966	0.966	0.966	0.792	0.792	0.795	0.92	0.919	0.921	0.853	0.855	0.853	0.95	0.95	0.952
SMOTE	0.927	0.928	0.928	0.769	0.769	0.769	0.909	0.908	0.91	0.806	0.807	0.806	0.95	0.95	0.95
SOMO	0.823	0.941	0.834	0.502	0.668	0.596	0.935	0.938	0.935	0.782	0.799	0.785	0	0.886	0.5
SVM-balance	0.957	0.956	0.957	0.825	0.822	0.827	0.917	0.919	0.919	0.889	0.903	0.891	0.922	0.924	0.923
SYMPROD	**0.967**	**0.967**	**0.967**	0.48	0.67	0.575	0.95	0.954	0.95	0.909	0.908	0.91	0	0.834	0.491
ASN-SMOTE	0.931	0.934	0.938	0.6485	0.47041	0.682	0.5402	0.42331	0.53	0.2924	0.345	0.574	0.2924	0.23	0.574
HHO-SMOTe	0.956	0.956	0.956	**0.889**	**0.897**	**0.889**	0.959	0.956	0.951	0.851	0.853	0.854	**0.977**	**0.974**	**0.973**

Dataset	habarman			hypothyroid			kc1			new-thyroid1			page-blocks0		
Method	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC
ADASYN	0.807	0.807	0.807	0.976	0.976	0.976	0.921	0.921	0.921	0.991	0.991	0.991	0.978	0.978	0.978
AND-SMOTE	0.754	0.755	0.757	0.977	0.977	0.977	0.945	0.945	0.945	0.963	0.963	0.963	0.983	0.983	0.983
ANS	0.79	0.791	0.794	0.976	0.976	0.976	0.915	0.916	0.916	0.674	0.893	0.727	0.979	0.979	0.979
Borderline-SMOTE1	0.778	0.778	0.778	0.981	0.981	0.981	0.926	0.926	0.926	0.991	0.991	0.991	**0.984**	**0.984**	**0.984**
Borderline-SMOTE2	0.765	0.763	0.766	0.975	0.975	0.975	0.917	0.917	0.917	0.981	0.981	0.981	0.978	0.978	0.978
distance-SMOTE	0.782	0.785	0.784	0.978	0.978	0.978	0.937	0.937	0.937	0.954	0.954	0.954	0.982	0.982	0.982
G-SMOTE	0.781	0.778	0.781	0.975	0.975	0.975	0.941	0.941	0.941	0.991	0.991	0.991	0.978	0.978	0.978
GASMOTE	0.487	0.445	0.495	0.863	0.844	0.871	0.827	0.817	0.827	0.744	0.717	0.745	0.627	0.597	0.638
Gaussian-SMOTE	0.777	0.777	0.783	0.75	0.773	0.78	0.776	0.79	0.792	0.95	0.953	0.951	0.971	0.971	0.971
KernelADASYN	0.794	0.792	0.801	0.987	0.987	0.987	0.968	0.981	0.968	0.972	0.972	0.972	0.975	0.975	0.975
kmeans-SMOTE	0.787	0.797	0.794	0.768	0.972	0.793	0.967	0.967	0.967	0.953	0.954	0.955	0.981	0.981	0.981
Random-SMOTE	0.787	0.791	0.79	0.979	0.979	0.979	0.922	0.922	0.923	0.99	0.991	0.99	0.977	0.977	0.977
Safe-Level-SMOTE	0.807	0.814	0.809	0.957	0.957	0.957	0.922	0.922	0.922	0.972	0.972	0.973	0.978	0.978	0.978
SDSMOTE	0.794	0.799	0.797	0.975	0.975	0.975	0.933	0.933	0.933	0.984	0.98	0.988	0.979	0.979	0.979
SMOTE	0.764	0.763	0.766	0.979	0.979	0.979	0.942	0.943	0.942	0.991	0.991	0.991	0.982	0.982	0.982
SOMO	0.441	0.652	0.56	0.656	0.965	0.714	0.957	0.957	0.957	0.905	0.968	0.909	0.98	0.98	0.98
SVM-balance	0.932	0.933	0.932	0.972	0.972	0.972	0.952	0.951	0.952	0.983	0.982	0.983	0.983	0.983	0.983
SYMPROD	0.78	0.785	0.781	0.978	0.979	0.979	0.933	0.933	0.933	0.962	0.963	0.962	0.982	0.982	0.982
ASN-SMOTE	**0.997**	**0.994**	**0.992**	0.6809	0.45673	0.696	0.6807	0.45604	0.696	0.5894	0.34384	0.662	0.700	0.618	0.707
HHO-SMOTe	0.7873	0.791	0.793	**0.989**	**0.98833**	**0.989**	**0.9806**	**0.98287**	**0.981**	**0.998**	**0.99733**	**0.998**	0.973	0.973	0.973

Dataset	pc1			pima			vehicle0			vehicle1			vehicle2		
Method	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC
ADASYN	0.958	0.958	0.958	0.781	0.78	0.781	0.939	0.938	0.939	0.84	0.841	0.841	0.962	0.963	0.962
AND-SMOTE	0.965	0.965	0.965	0.783	0.783	0.783	0.949	0.949	0.949	0.833	0.833	0.833	0.969	0.968	0.969

ANS	0.944	0.944	0.944	0.808	0.807	0.809	0.971	0.972	0.971	0.817	0.821	0.822	0.952	0.952	0.952
Borderline-SMOTE1	0.968	0.968	0.968	0.769	0.77	0.769	0.932	0.933	0.933	0.846	0.846	0.846	0.972	0.973	0.972
Borderline-SMOTE2	0.961	0.961	0.961	0.788	0.79	0.788	0.951	0.951	0.952	0.817	0.82	0.818	0.959	0.96	0.959
distance-SMOTE	0.972	0.973	0.972	0.809	0.81	0.81	0.955	0.954	0.955	0.858	0.86	0.859	0.946	0.944	0.947
G-SMOTE	0.979	0.979	0.979	0.794	0.793	0.795	0.948	0.949	0.948	0.828	0.83	0.83	0.974	0.973	0.974
GASMOTE	0.680	0.791	0.690	0.529	0.544	0.547	0.669	0.629	0.670	0.519	0.466	0.522	0.586	0.543	0.588
Gaussian-SMOTE	0.821	0.834	0.835	0.723	0.741	0.729	0.866	0.866	0.872	0.697	0.711	0.717	0.883	0.885	0.888
KernelADASYN	0.746	0.957	0.777	0.8	0.8	0.8	0.928	0.928	0.928	0.787	0.791	0.788	0.96	0.96	0.96
kmeans-SMOTE	0.965	0.965	0.965	0.79	0.79	0.79	0.946	0.946	0.946	0.833	0.833	0.833	0.964	0.963	0.964
Random-SMOTE	0.956	0.956	0.956	0.743	0.743	0.743	0.959	0.959	0.959	0.847	0.847	0.848	0.973	0.973	0.974
Safe-Level-SMOTE	0.935	0.935	0.936	0.77	0.77	0.77	0.967	0.969	0.967	0.819	0.818	0.82	0.968	0.968	0.969
SDSMOTE	0.96	0.96	0.96	0.761	0.763	0.761	0.941	0.941	0.941	0.849	0.847	0.85	0.966	0.966	0.966
SMOTE	0.974	0.974	0.974	0.764	0.764	0.764	0.948	0.951	0.948	0.828	0.828	0.828	0.963	0.963	0.963
SOMO	0.733	0.951	0.764	0.814	0.81	0.818	0.902	0.93	0.903	0.592	0.724	0.644	0.915	0.941	0.916
SVM-balance	0.937	0.939	0.937	0.842	0.843	0.843	0.959	0.959	0.959	0.863	0.867	**0.863**	0.981	0.981	0.981
SYMPROD	0.973	0.973	0.973	0.755	0.754	0.755	0.952	0.951	0.953	0.827	0.828	0.827	0.975	0.976	0.976
ASN-SMOTE	0.825	0.55594	0.838	**0.983**	**0.957**	**0.984**	0.793	0.625	0.815	0.6642	0.505	0.666	0.7895	0.64697	0.799
HHO-SMOTe	**0.982**	**0.985**	**0.986**	0.772	0.774	0.777	**0.986**	**0.992**	**0.987**	**0.8637**	**0.89833**	0.847	**0.9863**	**0.98767**	**0.988**

Dataset	vehicle3			yeast3			yeast4			yeast5			yeast6		
Method	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC	Gmean	F1-score	AUC
ADASYN	0.856	0.856	0.856	0.794	0.794	0.795	0.974	0.974	0.974	0.974	0.974	0.974	0.991	0.991	0.991
AND-SMOTE	0.863	0.864	0.863	0.794	0.795	0.794	0.976	0.976	0.976	0.981	0.981	0.981	0.988	0.987	0.988
ANS	0.865	0.866	0.866	0.822	0.822	0.822	0.975	0.975	0.975	0.975	0.974	0.975	0.66	0.972	0.716
Borderline-SMOTE1	0.873	0.874	0.874	0.826	0.826	0.826	0.974	0.974	0.974	0.984	0.984	0.984	0.99	0.99	0.99
Borderline-SMOTE2	0.859	0.858	0.86	0.828	0.829	0.829	0.958	0.958	0.958	**0.986**	**0.986**	**0.986**	0.987	0.987	0.987
distance-SMOTE	0.844	0.847	0.845	0.811	0.811	0.812	0.975	0.975	0.975	0.976	0.977	0.976	0.987	0.987	0.987
G-SMOTE	0.845	0.845	0.845	0.805	0.804	0.805	0.972	0.971	0.972	0.977	0.977	0.977	0.987	0.987	0.987
GASMOTE	0.527	0.476	0.530	0.464	0.318	0.479	0.498	0.522	0.566	0.476	0.481	0.555	0.408	0.407	0.520
Gaussian-SMOTE	0.629	0.648	0.658	0.789	0.79	0.793	0.946	0.947	0.947	0.945	0.946	0.946	0.976	0.976	0.976
KernelADASYN	0.827	0.829	0.827	0.751	0.754	0.755	0.911	0.914	0.913	0.978	0.979	0.979	0.981	0.98	0.981
kmeans-SMOTE	0.842	0.842	0.842	0.827	0.827	0.829	0.966	0.966	0.966	0.377	0.959	0.568	0.796	0.976	0.814
Random-SMOTE	0.829	0.829	0.829	0.854	0.853	0.854	0.975	0.975	0.975	0.971	0.971	0.971	0.986	0.986	0.986
Safe-Level-SMOTE	0.808	0.808	0.808	0.802	0.802	0.802	0.956	0.956	0.956	0.972	0.972	0.972	0.984	0.984	0.984
SDSMOTE	0.847	0.848	0.848	0.823	0.823	0.823	0.981	**0.981**	0.981	0.979	0.979	0.979	0.986	0.986	0.986
SMOTE	0.865	0.866	0.865	0.796	0.796	0.796	0.971	0.971	0.971	0.979	0.979	0.979	0.994	0.994	0.994
SOMO	0.608	0.771	0.65	0.647	0.737	0.67	0.795	0.932	0.81	0.354	0.957	0.56	0.67	0.975	0.72

			6						1			2			2
SVM-balance	**0.891**	**0.897**	**0.892**	0.903	0.907	0.904	0.975	0.976	0.975	0.967	0.967	0.967	0.987	0.988	0.977
SYMPROD	0.852	0.853	0.853	0.833	0.831	0.834	0.972	0.972	0.972	0.984	0.984	0.984	0.989	0.988	0.989
ASN-SMOTE	0.6692	0.50726	0.672	0.9104	0.756	0.911	0.7818	0.29784	0.792	0.9678	0.50278	0.968	0.8753	0.36762	0.88
HHO-SMOTe	0.807	0.804	0.809	**0.968**	**0.968**	**0.968**	**0.989**	0.973	**0.996**	0.971	0.974	0.979	**0.997**	**0.998**	**0.997**

In Fig. 3, we assessed G-mean values across 12 data sources using seven SMOTE techniques. A higher G-mean indicates a model's proficiency in both positive and negative class identification, a valuable metric for imbalanced classification. ANS-SMOTE and GASMOTE ranked lower, while ADASYN, SMOTE, RANDOM-SMOTE, and Borderline-SMOTE performed similarly. ADASYN had slightly lower G-mean for "cleveland-0." HHO-SMOTe consistently excelled across various datasets, demonstrating its robustness in imbalanced classification tasks.

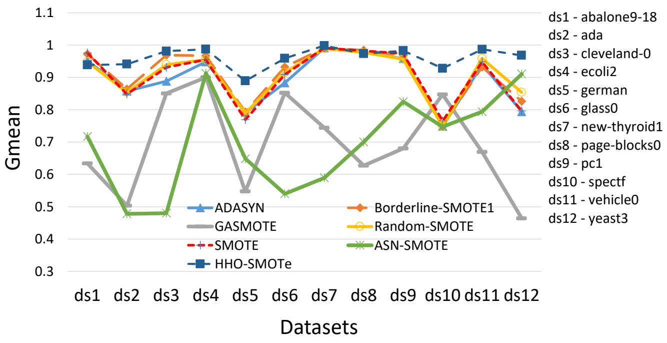

Fig. 3. Comparison of G-mean of seven SMOTE techniques.

In Fig. 4, we compare classification results using F1-score values for various SMOTE algorithms. The F1-score combines precision and recall, indicating a model's ability to balance false positives and false negatives. ANS-SMOTE and GASMOTE performed poorly compared to ADASYN, SMOTE, RANDOM-SMOTE, and Borderline-SMOTE. Conversely, HHO-SMOTe consistently achieved near-perfect F1-Scores (0.9 to 1) across datasets, showing its stability and reliability in diverse classification tasks.

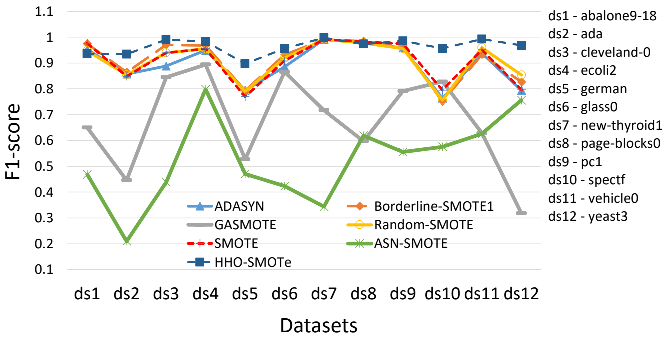

Fig. 4. Comparison of F1-score of seven SMOTE techniques.

In Fig. 5, we conducted a fresh evaluation of our classification studies, focusing on AUC (Area Under the Receiver Operating Characteristic Curve). AUC gauges a binary classification model's overall discrimination ability, considering true positive and false positive rates across different thresholds. The results show ANS-SMOTE and GASMOTE underperformed compared to ADASYN, SMOTE, RANDOM-SMOTe, and Borderline-SMOTE in AUC. In contrast, HHO-SMOTe consistently achieved high AUC values (typically 0.9 to 1), showcasing its adaptability across diverse datasets and confirming its effectiveness in classification tasks, especially when class separation is crucial.

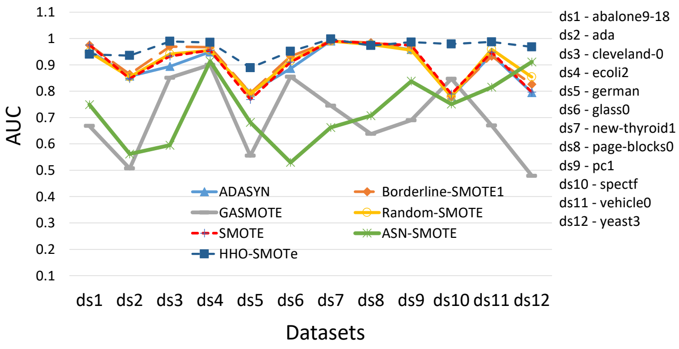

Fig. 5. Comparison of AUC of seven SMOTE techniques.

This research employs the of the well-known credit card fraud detection dataset [46]. The dataset was prepared by the ULB Machine Learning Group, which specializes in big data mining and fraud detection [47]. The dataset covers credit card transactions made by European credit card clients within two days in September 2013. Dataset have 492 fraudulent transactions out of 284807 total. Meanwhile, all attributes except ‘‘Time’’ and ‘‘Amount’’ are numerical due to transformation carried out on dataset using dimensionality reduction technique called principal component analysis (PCA). ‘‘Amount’’ attribute is the cost of the transaction, and ‘‘Time’’ attribute is the seconds that elapsed between a transaction and the first transaction in the dataset. ‘‘Class’’ is the dependent variable, has a value of 1 for fraudulent and 0 for legitimate.

In Fig. 6, we conducted extensive comparison using credit card fraud dataset known for its vast transaction volume. The goal was to thoroughly evaluate the stability and accuracy of our method within the realm of big data challenges, compared to other techniques. As depicted in the figure, HHO-SMOTe achieved highest AUC score, an impressive 0.96, surpassing other methods with scores below 0.94. These methods ranked in descending order as borderline-2, SMOTE, ADASYN, Borderline1, ASN-SMOTE, GASMOTE, and Random SMOTE. In terms of the F1-Score, all algorithms consistently scored above 0.99, even reaching a perfect score of 1. Regarding the G-mean metric, HHO-SMOTe demonstrated its superiority with a score exceeding 0.95, while its counterparts fell short with scores below 0.94.

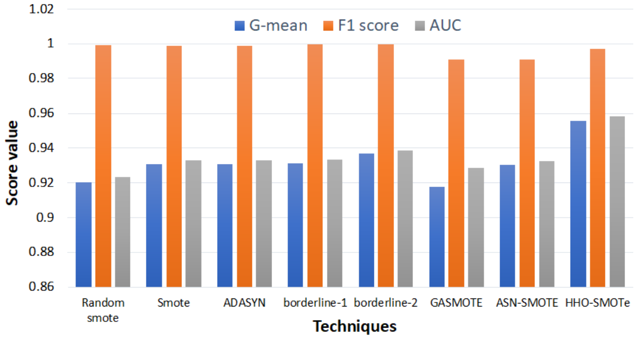

Fig. 6. Comparison of different SMOTE techniques and HHO-SMOTe using fraud detection dataset.

VI. Conclusion

In summary, the HHO-SMOTe approach represents a significant advancement in effectively addressing complexities of imbalanced datasets in classification tasks. By seamlessly integrating various classifiers with the Harris Hawk search optimization algorithm and SMOTE, we have established a robust framework capable of producing precise and reliable predictions for imbalanced data scenarios. These results hold substantial implications for a wide range of real-world applications where improved classification accuracy and data balance correction play pivotal roles in informed decision-making. Furthermore, our research contributes significantly to the field of imbalanced data handling by shedding light on a potent methodology that enhances the performance of classification models across diverse domains. This amalgamation of state-of-the-art techniques has the potential to mitigate challenges posed by skewed data distributions, ultimately enabling more accurate and trustworthy predictions.

References

[1] Z Zhang, Chunkai, Ying Zhou, and Yepeng Deng. "VCOS: A novel synergistic oversampling algorithm in binary imbalance classification" IEEE Access vol7 2019,p145435-145443.

[2] Fotouhi, Sara, Shahrokh Asadi, and Michael W. Kattan. "A comprehensive data level analysis for cancer diagnosis on imbalanced data" Journal of biomedical informatics vol90, 2019,103089.

[3] Phua, Clifton, Damminda Alahakoon, and Vincent Lee. "Minority report in fraud detection: classification of skewed data" Acm sigkdd explorations newsletter .vol6,no1, 2004, p50-59.

[4] Castillo, M. Dolores, and José Ignacio Serrano. "A multistrategy approach for digital text categorization from imbalanced documents" ACM SIGKDD Explorations Newsletter vol6,no1, 2004.

[5] Liu, Liang, et al. "Prediction of protein–protein interactions based on PseAA composition and hybrid feature selection" Biochemical and biophysical research communications vol380,no.2,2009318-322.

[6] He, Haibo, and Xiaoping Shen. "A Ranked Subspace Learning Method for Gene Expression Data Classification." IC-AI, vol1,2007, 58-364,.

[7] Soda, Paolo. "A multi-objective optimisation approach for class imbalance learning." Pattern Recognition vol44,no8, ,2011,1801-1810.

[8] He, Haibo, and Edwardo A. Garcia. "Learning from imbalanced data." IEEE Trans. on knowledge and data engineerin vol21,no.9, 2009,1263-1284.

[9] Qiong, G. U., et al. "A comparative study of cost-sensitive learning algorithm based on imbalanced data sets.". Microelectron. Comput. vol28,no.8,2009,146–149.

[10] Chawla, N., et al. "SMOTE: synthetic minority over-sampling technique." Journal of AI research vol16,2002,321-357.

[11] Heidari, Ali Asghar, et al. "Harris hawks optimization: Algorithm and applications." Future generation computer systems vol97,2019,849-872.

[12] S. Ebenuwa, M. Sharif, M. Alazab and A. Al-Nemrat, "Variance Ranking Attributes Selection Techniques for Binary Classification Problem in Imbalance Data," in IEEE Access.vol7,2019,24649-24666.

[13] G. Rekha, A. Tyagi, and R. Krishna, Solving Class Imbalance Problem Using Bagging, Boosting Techniques, with and Without Using Noise Filtering Method. International Journal of Hybrid Intelligent Systems15,no2, 2019, 67–76.

[14] X. Liu, J. Wu, and Z. Zhou, Exploratory Undersampling for Class-Imbalance Learning. IEEE Transactions on Syst, Man, & Cybernetics, vol. 39, no.2,2009,539-550.

[15] Jiang, K., Lu, J. & Xia, K. A Novel Algorithm for Imbalance Data Classification Based on Genetic Algorithm Improved SMOTE. Arab J Sci Eng vol41, 2016,3255–3266.

[16] Nnamoko, Nonso, and Ioannis Korkontzelos. "Efficient treatment of outliers and class imbalance for diabetes prediction." Artificial intelligence in medicine 104 ,2020, 101815.

[17] Liu, Shigang, et al. "Addressing the class imbalance problem in twitter spam detection using ensemble learning." Computers & Security vol69, 2017, 35-49.

[18] Ma, J., Afolabi, D.O., Ren, J. et al. "Predicting Seminal Quality via Imbalanced Learning with Evolutionary Safe-Level Synthetic Minority Over-Sampling Technique". Cogn Comput vol.13, 2021, 833–844.

[19] Susan, S, Kumar, A. "The balancing trick:Optimized sampling of imbalanced datasets A brief survey of the recent State of the Art". Engineering Reports. 2021; 3: e12298.

[20] Tanapol Kosolwattana, Chenang Liu, Renjie Hu, Shizhong Han, Hua Chen and Ying Lin. "A self-inspected adaptive SMOTE algorithm (SASMOTE) for highly imbalanced data classification in healthcare". BioData Mining16, 15. 2023.

[21] J. Wang; M. Xu; H. Wang; J. Zhang. "Classification of Imbalanced Data by Using the SMOTE Algorithm and Locally Linear Embedding", 2006 8th international Conference on Signal Processing,2006, 9505808.

[22] P. Jeatrakul, K.W. Wong and C.C. Fung ,"Classification of Imbalanced Data by Combining the Complementary Neural Network and SMOTE Algorithm", Neural Information Processing Models &App, Vol6444,no.2,2010,152-159.

[23] Aimin Zhang, Hualong Yu, Zhangjun Huan, Xibei Yang,."SMOTE-RkNN:A hybrid re-sampling method based on SMOTE and reverse k-nearest neighbors";Info. Sci.Vol.595,2022,p 70-88.

[24] M. Shehab, I. Mashal, Z. Momani, M. Shambour, A. AL-Badareen, S. Al-Dabet, N. Bataina, A. Ratib Alsoud,"Harris Hawks Optimization Algorithm: Variants and Applications",Archives of Computational Methods in Engineering vol. 29, 2022, p5579–5603.

[25] B. Tripathy, P. Maddikunta, Q. Pham,T. Kapal Dev, S. Pandya, and B. ElHalawany."Harris Hawk Optimization:A Survey on Variants and Applications" , Vol 2022 | Article ID 2218594 |.

[26] J.C. Bednarz, "Cooperative hunting in harris' hawks (parabuteo unicinctus", Science vol239 ,1988, 1525.

[27] Fathimathul Rajeena,Walaa N. Ismail, Mona A.S. A Metaheuristic Harris Hawks Optimization Algorithm for Weed Detection Using Drone Images, (ISSN 2076-3417). 2023 , 13(12), 7083.

[28] Su. Muruganandam, Vij. Natarajan, Raja Soos., Raj. Murugesan HHO-ACO hybridized load balancing technique in cloud computing, Feb 2023 Inter. Jour. of Info. Tech. Vol 15, P 1357–1365.

[29] Bisong, E.: Google Colaboratory, pp. 5964. Apress, Berkeley, CA 2019.

[30] He, H. and Bai, Y. and Garcia, E. A. and Li, S., "ADASYN: adaptive synthetic sampling approach for imbalanced learning", Proceedings of IJCNN, 2008, p. 1322—1328.

[31] J. Yun, J. Ha, and J. Lee, "Automatic Determination of Neighborhood Size in SMOTE", Proceedings of 10th International Conf. on Ubiquitous Info. Management and Comm.,2016, p. 100z1—100:8

[32] W Siriseriwan, and K. Sinapiromsaran, "Adaptive neighbor synthetic minority oversampling technique under 1NN outcast handling",Songklanakarin Jour of Science and Tech. vol39,no5,2017, p. 565-576

[33] Hui Han, Wen-Yuan Wang & Bing-Huan Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning", Advances in Intelligent Computing, vol6,2005, p. 878-887.

[34] De La Calleja, J. and Fuentes, O., "A distance-based over-sampling method for learning from imbalanced datasets", Proceedings of the 20th International Florida Artificial Intelligence, 2007, p. 634-635

[35] Sandhan, T. and Choi, J. Y., "Handling Imbalanced Datasets by Partially Guided Hybrid Sampling for Pattern Recognition", 2014 22nd International Conference on Pattern Recognition, 2014, p. 1449-1453

[36] Hansoo Lee and Jonggeun Kim and Sungshin Kim, "Gaussian-Based SMOTE Algorithm for Solving Skewed Class Distributions", Int. J. Fuzzy Logic and Intelligent Systems,Vol 17,no4, 2017, p. 229-234

[37] Tang, B. and He, H., "KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning", 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, p. 664-671

[38] G. Douzas and F. Bacao and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE", Information Sciences,Vol 465, 2018, p. 1-20.

[39] Don, "A New Over-Sampling Approach: Random-SMOTE for Learning from Imbalanced Data Sets", Knowledge Scienc,Vol 7091, 2011 p. 43-352.

[40] Bunkhumpornpat, C. and Sinapiromsaran, K. and Lursinsap, C., "Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem", Proceedings 13th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining, vol volume 5476, 2009, p. 475-482.

[41] Li, K. and Zhang, W. and Lu, Q. and Fang, X., "An Improved SMOTE Imbalanced Data Classification Method Based on Support Degree", Inter. Conf .on Identification, Info. and Knowl. in ITO, 2014, p. 34-38

[42] G. Douzas and F. Bacao, "Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning", Expert Systems with Applications, Vol.42,2017, p. 40-52

[43] Farquad, M., Bose, I., "Preprocessing Unbalanced Data Using Support Vector Machine", Decis Support Syst.,Vol53,no1,2012,p. 226-233.

[44] Kunakorntum, I. and Hinthong, W. and Phunchongharn, P.,"A Synthetic Minority Based on Probabilistic Distribution (SyMProD) Oversampling for Imbalanced Datasets", IEEE Access, 2020, p. 114692-114704.

[45] X. Yi, Y. Xu, Q. Hu, S. Krishnamoorthy, W. Li, Z. Tang. "ASN-SMOTE: a synthetic minority oversampling method with adaptive qualified synthesizer selection". Complex & Intel. Systems, Vol.8, 2022, p.2247–2272.

[46] Credit Card Fraud Detection. Accessed: Oct. 2021, 26. [Online]. Available: https://kaggle.com/mlg-ulb/creditcardfraud.

[47] H. Patel, D. S. Rajput, G. T. Reddy, C. Iwendi, A. K. Bashir, and O. Jo, ''A review on classification of imbalanced data for wireless sensor networks,'' Int. J. Distrib. Sensor Netw., vol. 16, no. 4, 2022.