IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 11, 2022, accepted April 19, 2022, date of publication April 22, 2022, date of current version May 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169512

Stop Oversampling for Class Imbalance Learning: A Review

AHMAD S. TARAWNEH[1], AHMAD B. HASSANAT[2], (Member, IEEE), GHADA AWAD ALTARAWNEH[3], AND ABDULLAH ALMUHAIMEED[4]

[1]Department of Algorithms and Their Applications, Eötvös Loránd University, 1053 Budapest, Hungary
[2]Faculty of Information Technology, Mutah University, Karak 61710, Jordan
[3]Department of Accounting, Mutah University, Karak 61710, Jordan
[4]The National Centre for Genomics and Bioinformatics, King Abdulaziz City for Science and Technology, Riyadh 11442, Saudi Arabia

Corresponding authors: Ahmad S. Tarawneh (ahmad.trwh@gmail.com) and Abdullah Almuhaimeed (muhaimed@kacst.edu.sa)

ABSTRACT For the last two decades, oversampling has been employed to overcome the challenge of learning from imbalanced datasets. Many approaches to solving this challenge have been offered in the literature. Oversampling, on the other hand, is a concern. That is, models trained on fictitious data may fail spectacularly when put to real-world problems. The fundamental difficulty with oversampling approaches is that, given a real-life population, the synthesized samples may not truly belong to the minority class. As a result, training a classifier on these samples while pretending they represent minority may result in incorrect predictions when the model is used in the real world. We analyzed a large number of oversampling methods in this paper and devised a new oversampling evaluation system based on hiding a number of majority examples and comparing them to those generated by the oversampling process. Based on our evaluation system, we ranked all these methods based on their incorrectly generated examples for comparison. Our experiments using more than 70 oversampling methods and nine imbalanced real-world datasets reveal that all oversampling methods studied generate minority samples that are most likely to be majority. Given data and methods in hand, we argue that oversampling in its current forms and methodologies is unreliable for learning from class imbalanced data and should be avoided in real-world applications.

INDEX TERMS Oversampling, SMOTE, imbalanced datasets, machine learning, Hassanat metric.

I. INTRODUCTION

When training a dataset with examples from one class greatly outnumbering those from the other, a phenomenon known as class imbalance emerges. The majority class is usually referred to as such, whereas the minority class is referred to as such. There may be more than one majority class and more than one minority class in a single dataset. The main cause of class imbalance is that classifiers trained on unequal training sets have a prediction bias, which is linked to poor performance in the minority class(es), Depending on the dataset utilized, the bias could range from a little imbalance to a severe imbalance [1]–[5]. This problem has grown and has become a significant difficulty since the minority class is frequently of critical importance, as it represents favorable examples that are rare in nature or expensive to obtain [6]. This is true when considering contexts such as Big Data analytics [7]–[13], Biometrics [14]–[22], gene profiling [23], credit card fraud detection [24], [25], face image retrieval [24], content-based image retrieval [26], [27], disease detection [28]–[32], internet of things [33]–[43], Natural Language Processing [44], [45], network security [46]–[52], image recognition [53]–[58], Anomaly Detection [59]–[69], etc.

In formal terms, a supervised machine learning dataset D with n instances belonging to m classes $C1, C2, C3, \ldots, Cm$ is said to be a class imbalanced dataset if and only if for any $Ci, Cj \exists |Ci| \gg |Cj|$, where i and j are indexes $1, 2, 3, \ldots, m$, and $i \neq j$.

There are several approaches to solving class imbalance problem before starting classification, such as:

- More samples from the minority class(es) should be acquired from the knowledge domain.
- Changing the loss function to give the failing minority class a higher cost [70].
- Oversampling the minority class.
- Undersampling the majority class.

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

- Any combination of previous approaches.

Each of the aforementioned approaches has its own set of benefits and drawbacks [71], [72]. Oversampling, on the other hand, is the most often used approach among them, as seen by the multitude of oversampling methods published in the last two decades. However, this does not necessarily imply that the oversampling approach is beneficial. Oversampling approaches boost the quantity of minority-class instances by creating new ones out of thin air based only on their similarity to one or more of the minority's examples. This is troublesome since such methods may raise the likelihood of the learning process being overfitted [73]–[75], [75], [76]. On paper, the overfitted synthetic datasets produce good machine learning results, however this is not always the case in practice. Another more critical problem of oversampling is that the fabricated examples could exist in the real world belonging to a different class, regardless of how similar it is to the minority's examples, as we always have examples from class A that are the closest to examples from a different class B. Therefore, we argue that, even if such synthesizing generates favorable outcomes on paper, negative results can be easily obtained in practice. The major goal of this study, in addition to reviewing a large number of oversampling methods, is to prove our counterclaim on the use of oversampling as a solution to the problem of class imbalance, which is as follows:

Oversampling in its current forms and methodologies is a misleading approach that should be avoided since it feeds the learning process with falsified instances that are pushed to be members of the minority class when they are most likely members of the majority.

To the best of our knowledge, the only methodology for proving an oversampling method's goodness is its classification accuracy metrics after the classification of the oversampled datasets, with no tests for the validity of the synthesized instances and if they are appropriate for training a model for real-world use. Therefore, we find oversampling practitioners are pleased with their machine learning outcomes in the lab, but they should consider how much harm could be done in practice outside of the lab, particularly in medical and other vital applications. The harm is exacerbated when we realize that several of these methods have become integral parts of APIs and machine learning packages, such as Python imbalanced-learn API [77] and Smote-Variants API [78]. We prove our counterclaim in this paper by using a number of typical oversampling methods on several benchmark datasets, concealing some of the majority examples, and then comparing the created examples to the hidden majority examples to determine if they approximately match. Finding such counter examples proves our counterclaim.

The following is the structure of this paper: The literature review of class imbalance problem is presented in the second section. The mythology of proving our counterclaim is illustrated in Section Three. And the experimental results are listed and discussed in section four.

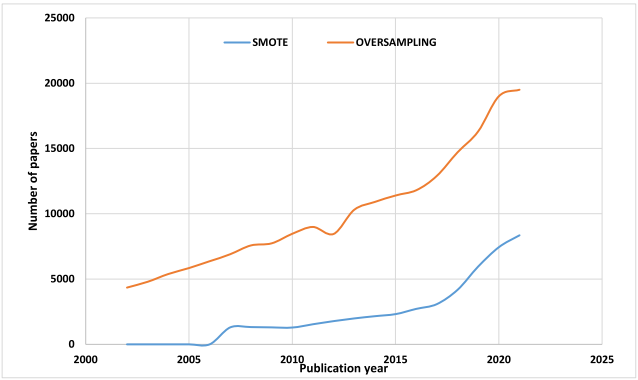


FIGURE 1. The number of publications that have the terms "oversampling" and/or "SMOTE."

II. LITERATURE REVIEW OF OVERSAMPLING METHODS

In the literature, there are various ways to machine learning from class imbalance data. One of the most prevalent ways is oversampling, particularly Synthetic Minority Oversampling Technique approaches (SMOTE). On January 26, 2022, a Google Scholar search for the term "SMOTE" yielded 77,300 results, while a search for "oversampling" yielded 297,000 results. This is merely a foreshadowing of the developing trend of oversampling. Figure 1 depicts the nearly exponential increase in the number of articles that dealt with, employed, or addressed oversampling and/or SMOTE.

The relevance of the well-defined class imbalance problem and the simplicity of oversampling solutions are the reasons for this abnormal surge in oversampling research. Anyone with a rudimentary understanding of machine learning can come up with a novel way to produce fresh similar examples given some minority examples. There could be an infinite number of such solutions.

Several studies, such as [1], [79], [80], have reviewed various oversampling approaches; nevertheless, they are not thorough and have not paid adequate attention to validating the oversampling approach to the problem of class imbalance.

One of the earliest and most extensively utilized approaches for class imbalance is the SMOTE method [81]. It interpolates synthetic examples between nearest neighbors from the training set's collection of minority class cases. As a result, by merging the properties of seed instances with randomly picked k-nearest neighbors, a synthetic sample is generated. The earliest version of the SMOTE algorithm relied solely on synthetic oversampling. They also used a combination of synthetic oversampling and undersampling, which might be useful [82]. SMOTE was tested on nine benchmark datasets and proven to improve classification performance.

SVMSMOTE [83], which is based on SMOTE, focuses on constructing SVM modifications to successfully handle the problem of class imbalance. Oversampling, cost-sensitive learning, and undersampling are some of the heuristics used in SVM modeling. This method produced promising results when compared to other oversampling methods.

Borderline-SMOTE [84] is an SMOTE-based minority oversampling method that only oversamples the minority examples around the borderline. In comparison to SMOTE and other random oversampling methods investigated, their findings show that this solution improves classification results for the minority class.

Oversampling by a synthetic inverse minority is used in Reverse-SMOTE (R-SMOTE) [85], a technique based on SMOTE and the inverse near-neighbor idea. R-SMOTE beats other over-sampling methods in terms of precision, F-measurement, and accuracy, according to this study that compared traditional sampling procedures to alternative methods, including SMOTE. In the comparison, eight benchmark datasets were employed.

Constrained Oversampling (CO) [86] is a technique for reducing noise in oversampling. This method is used to extract the overlapping regions in a dataset. Ant Colony Optimization is then used to define the boundaries of minority regions. Most significantly, in order to create a balanced dataset, fresh samples are synthesized via oversampling under constraints. This method varies from others in that it includes noise-reduction constraints in the oversampling process. CO outperforms a range of oversampling benchmarks, according to their results.

In addition, the Majority Weighted Minority Oversampling Technique (MWMOTE) [87] was offered as a solution to the problem of class-imbalance learning. MWMOTE finds and weights difficult-to-learn informative minority class samples based on their distance from nearby majority class samples. It then creates synthetic samples from the weighted informative minority class samples using a clustering algorithm. The primary premise of MWMOTE is that all generated samples must belong to one of the minority class clusters. In terms of numerous assessment measures, the provided results suggest that MWMOTE is superior than or similar to some other existing approaches.

Adaptive synthetic (ADASYN) [88] was given with the goal of eliminating bias and moving the classification decision boundary in the direction of the hard examples. The primary idea behind ADASYN is to use a weighted distribution for different minority class examples based on their learning difficulty, with more synthetic data created for more difficult minority class examples than for easier minority class examples. The efficacy of this method is proved by the results of experiments conducted on a variety of datasets using five different evaluation measures.

Synthetic Minority Over-Sampling Technique Based on Furthest Neighbor Algorithm (SOMTEFUNA) [6] is another exciting and recent method for machine learning from imbalanced datasets. To produce fresh synthetic minority examples, this method employs the farthest neighbor examples. SOMTEFUNA has a number of advantages over some other approaches, one of which being the lack of tuning parameters, which makes it easier to be used in real-world scenarios. Using Naive Bayes and Support Vector Machine classifiers, the method compared the benefits of resampling to common methods such as SMOTE and ADASYN. The reported findings show that SOMTEFUNA is a viable alternative to the other oversampling methods, according to its reported results.

Sampling WIth the Majority (SWIM) [89] is a synthetic oversampling method that is robust in cases of significant class imbalance. SWIM's fundamental feature is that it uses the density of the well-sampled majority class to direct the creation process. SWIM's model was built using both the radial basis function and the Mahalanobis distance. SWIM was put to the test on 25 benchmark datasets, and the findings show that it beats some of the most common oversampling methods.

Other ways of oversampling include, but are not limited to, the work of [78], [90]–[118].

The validation process is what all oversampling methods have in common, which is basically the evaluation of the classifier's performance employed to classify the oversampled datasets using one or more accuracy measures such as Accuracy, Precision, Recall, F-measure, G-mean, Specificity, Kappa, Matthews correlation coefficient (MCC), Area under the ROC Curve (AUC), True positive rate, False negative (FN), False positive (FP), True positive (TP), True negative (TN), and ROC curve. Table 1 lists 72 oversampling methods, including their known names, references, the number of datasets utilized, the number of classes in these datasets, the classifiers employed, and the performance metrics used to validate the classification results after oversampling.

As can be seen from the previous discussion and Table 1, all the aforementioned oversampling methods use the classification accuracy measures of the synthesized data to verify their goodness, assuming that the synthesised examples belong to the minority class. On paper, however, the accuracy measures appear to be good if the data is over-fitted, which is common when using Oversampling methods [71]–[76].

Another critical problem with the oversampling approach is the assumption that the synthetic examples belong to the minority class; do they truly belong to the minority class?

None of the previous literature has answered this critical question. This study aims to provide a validation system for oversampling methods, in order to determine to what degree these methods synthesize unrealistic examples; assuming they are belonging to the minority when they are not.

III. METHOD AND DATA

The proposed validation system for oversampling methods works by hiding a subset of the majority's examples, which is referred to as the hidden subset. Although the hidden majority examples are part of the population, we excluded them from the training dataset since we assumed they were not obtained from the real-world knowledge domain. Because all oversampling approaches do not access the entire real-world population, this assumption is correct.

It is important to make sure that the class imbalance problem still exists after concealing the hidden subset.

TABLE 1. Summary of the methods used in this study. In this table, C4.5 is Decision Tree C4.5, LR is Logistic Regression, LDA is linear discriminate analysis, NB is naive bayes, RF is random forest and ANN is artificial neural network.

ID	Method Name	Reference	No. Datasets	No. Classes	Classifiers	Performance measures
M1	SMOTE	[81]	6	Binary	C4.5 Ripper NB	ROC curve AUC
M2	SMOTE TomekLinks	[119]	13	Binary	C4.5	AUC
M3	SMOTE ENN	[119]	13	Binary	C4.5	AUC
M4	Borderline SMOTE1	[84]	4	Binary	C4.5	TP rate F-values
M5	Borderline SMOTE2	[84]	4	Binary	C4.5	TP rate F-values
M6	ADASYN	[89]	5	Binary	C4.5	Accuracy, Precision Recall, F-measure G-mean
M7	AHC	[121]	1	Binary	C4.5 KNN, SVM NB AdaBoost	Recall, Specificity Accuracy ROC, G-mean weighted accuracy
M8	distance SMOTE	[122]	10	Binary	Linear Regression	Precision Recall F-measure
M9	polynom fit SMOTE	[123]	1	Binary	SVM	TP rate, TN rate
M10	Stefanowski	[124]	9	Binary	C4.5 MODLEM	Accuracy Specificity Recall
M11	ADOMS	[125]	12	Binary	ANN	G-mean
M12	Safe Level SMOTE	[126]	2	Binary	NB, SVM	Precision, Recall F-values, AUC
M13	MSMOTE	[127]	3	Binary	C4.5 AdaBoost	Precision, Recall F-values
M14	DE oversampling	[128]	10	Binary	SVM	F-measure, AUC
M15	SMOBD	[129]	9	Binary	SVM	G-mean, AUC
M16	SUNDO	[130]	4	Binary	CART, SVM	Accuracy, TP TN, FP, FN
M17	MSYN	[131]	10	Binary	C4.5	AUC, F-measure
M18	SVM balance	[132]	1	Binary	ANN RF LR	Accuracy, Recall Specificity, AUC
M19	TRIM SMOTE	[133]	11	Binary	C4.5	AUC, F-measure
M20	SMOTE RSB	[134]	44	Binary	C4.5	AUC
M21	ProWSyn	[135]	10	Binary	ANN C4.5	F-measure G-mean, AUC
M22	SL graph SMOTE	[136]	8	Binary	KNN, RIPPER C4.5 NB Logistic Tree	F-measure, AUC
M23	LVQ SMOTE	[137]	8	Binary	ANN NB RF SVM, OLVQ3	Recall Specificity G-mean
M24	SOI CJ	[138]	20	Binary	C4.5	Precision, Recall F-measure, AUC
M25	ROSE	[139]	20	Binary	C4.5 LR	AUC

TABLE 1. *(Continued.)* **Summary of the methods used in this study. In this table, C4.5 is Decision Tree C4.5, LR is Logistic Regression, LDA is linear discriminate analysis, NB is naive bayes, RF is random forest and ANN is artificial neural network.**

M26	SMOTE OUT	[140]	18	Binary	SVM	F-measure
M27	SMOTE Cosine	[140]	18	Binary	SVM	F-measure
M28	Selected SMOTE	[140]	18	Binary	SVM	F-measure
M29	LN SMOTE	[141]	15	Binary	C4.5 NB	F-measure G-mean Recall
M30	MWMOTE	[88]	20	Binary	C4.5 AdaBoost, KNN ANN	G-mean, AUC
M31	PDFOS	[142]	6	Binary	RBF	F-measure G-mean, AUC
M32	RWO sampling	[143]				
M33	NEATER	[144]	22	Binary	C4.5 RF, SVM	G-mean, AUC
M34	DEAGO	[145]	2	Binary	ANN	AUC
M35	Gazzah	[146]	2	Multi	SVM	True Negative Rate True Positive Rate
M36	MCT	[147]	14	Binary	C4.5 KNN, SVM NB	FN, FP
M37	SMOTE IPF	[148]	9	Binary	C4.5	AUC
M38	KernelADASYN	[149]	7	Binary	C4.5	Precision Recall F-measure G-Mean
M39	MOT2LD	[150]	15	Binary	CART	G-mean F-measure Kappa
M40	V SYNTH	[151]	6	Binary	LR LDA	Recall Specificity Accuracy FPR, FNR
M41	OUPS	[152]	45	Binary	SVM KNN LDA LR	Recall Specificity G-mean
M42	SMOTE D	[153]	66	Binary	C4.5 KNN, SVM	F-measure
M43	SMOTE PSO	[154]	18	Multi	SVM	AUC and G-mean
M44	CURE SMOTE	[155]	12	Multi	CART RF	F-measure G-mean AUC
M45	CE SMOTE	[156]	10	Binary	C4.5	F-measure G-mean
M46	Edge Det SMOTE	[157]	9	Binary	SVM	F-measure G-mean accuracy
M47	CBSO	[158]	8	Binary	ANN C4.5	F-measure G-mean
M48	ASMOBD	[129]	9	Binary	SVM	G-mean and AUC
M49	Assembled SMOTE	[159]	4	Binary	SVM	Precision Recall F-score Accuracy

TABLE 1. *(Continued.)* **Summary of the methods used in this study. In this table, C4.5 is Decision Tree C4.5, LR is Logistic Regression, LDA is linear discriminate analysis, NB is naive bayes, RF is random forest and ANN is artificial neural network.**

M50	SDSMOTE	[160]	4	Binary	C4.5 AdaBoost Bagging	AUC F-measure
M51	DSMOTE	[161]	11	Binary	Naïve Bayes KNN, SVM	Accuracy Recall Precision F-measure G-mean
M52	G SMOTE	[162]	4	Binary	1NN and SVM	AUC
M53	NT SMOTE	[163]	3	Binary	SVM and C4.5	Accuracy Precision
M54	Lee	[164]	8	Binary	SVM	Recall G-mean
M55	SMOTE PSOBAT	[165]	30	Binary	ANN C4.5	Kappa Accuracy
M56	MDO	[166, 167]	20	Multi	C4.5, KNN RIPPER	AUC G-mean Precision Recall F-measure
M57	Random SMOTE	[168]	10	Multi	KNN	G-mean Accuracy
M58	VIS RST	[169]	6	Binary	AdaBoost C4.5	True positive rate true negatives rate F-measure, and AUC
M59	GASMOTE	[170]	10	Binary	C4.5	F-measure, G-mean
M60	SMOTE FRST 2T	[171]	1	Binary	C4.5	AUC, FP, FN
M61	AND SMOTE	[172]	12	Binary	CART	AUC
M62	NRAS	[173]	41	Binary	ANN SVM, LDA LR	Recall Specificity G-mean
M63	AMSCO	[174]	30	Binary	ANN	Precision Recall F-measure G-mean
M64	SSO	[175]	6	Binary	ANN	F-measure G-mean
M65	NDO sampling	[176]	1	Binary	LR Decision Tree C5 ANN, SVM	F-measure, Recall
M66	DSRBF	[177]	13	Multi	ANN	Accuracy Recall
M67	Gaussian SMOTE	[104]	1	Binary	SVM	Accuracy Recall Specificity
M68	Supervised SMOTE	[178]	2	Binary	SVM	Accuracy Recall Specificity MCC , AUC
M69	SN SMOTE	[179]	39	Binary	KNN, ANN C4.5	Accuracy Recall Specificity, AUC
M70	CCR	[180]	32	Binary	CART, KNN NB, SVM	F-measure G-mean, AUC
M71	ANS	[181]	14	Binary	ANN, KNN NB, SVM C4.5	F-measure, AUC
M72	cluster SMOTE	[182]	2	Binary	RIPPER	AUC

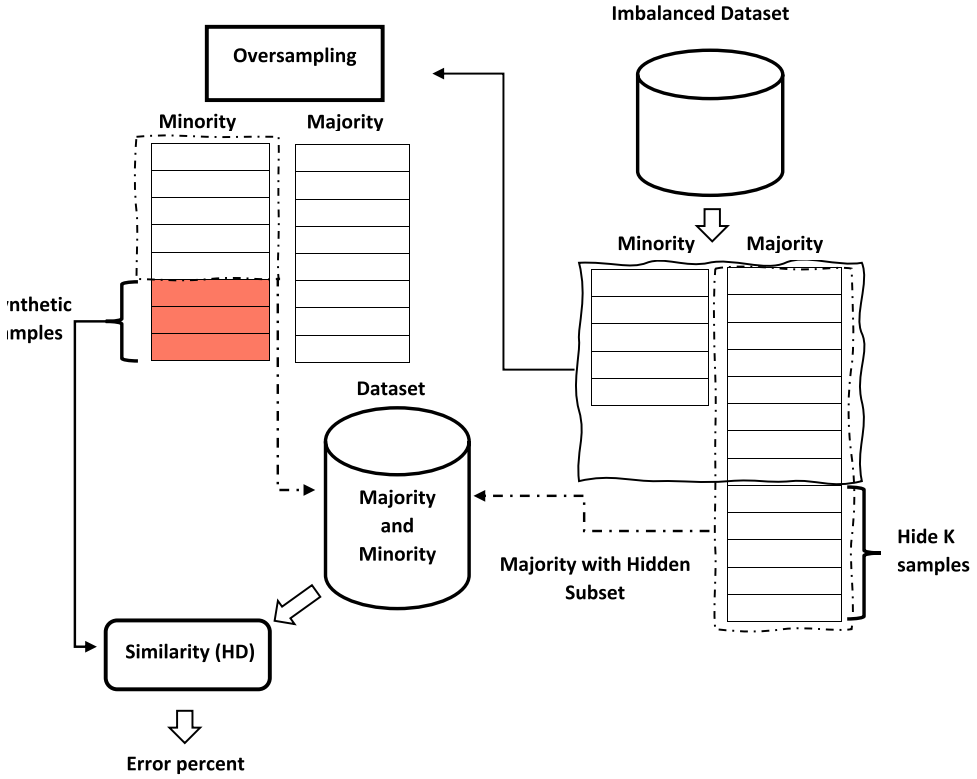


FIGURE 2. Flow diagram of the proposed validation system.

After that, we apply the oversampling method that needs to be validated on the remaining dataset in order to generate new examples, which is referred to as the synthetic subset. The hidden subset is then returned to the training set.

The generated examples in the synthetic subset are claimed to belong to the minority class by all oversampling methods. We compare the similarity between these examples (the synthetic subset) and all examples in the original training set before oversampling to see if these synthesized examples belong to the minority or the majority.

Figure 2 illustrates the proposed validation system.

In order to determine the degree of similarity, we need a similarity measure such as Euclidean distance (ED), Manhattan distance (MD), Hassanat distance (HD) [182], etc. In this paper, we opt for HD as being invariant to noise, outliers and data scale, since the nature of this metric prevents each feature from having a distance greater than one, regardless of the scale of the features in the targeted dataset. Furthermore, HD had been shown to outperform a wide range of machine learning similarity measures, including the most common ones like ED and MD [183]–[187].

HD can be expressed mathematically as in equation1.

$$D(p_i, q_i) = \begin{cases} 1 - \dfrac{1 + min(p_i, q_i)}{1 + max(p_i, q_i)}, & min(p_i, q_i) \geq 0 \\ 1 - \dfrac{1 + min(p_i, q_i) + |min(p_i, q_i)|}{1 + max(p_i, q_i) + |min(p_i, q_i)|}, & min(p_i, q_i) < 0 \end{cases} \tag{1}$$

and for the total distance between two examples is

$$HD(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{N} D(p_i, q_i) \tag{2}$$

where p and q are feature vectors and N is the number of features in each vector.

It is worth mentioning that we are proposing a validation system, not an evaluation system, the similarity measure using HD is meant to find the number of examples taken from the synthetic subset that are similar to the minority as the core of our validation system. i.e. HD is calculated between the generated examples and the original examples. Those generated examples, which are more similar/nearest to the majority indicate the error of the oversampling method validated. This error is calculated according to equation 3.

$$Error = \frac{CM}{SS} \tag{3}$$

where CM is the number of synthetic examples that are more similar to majority examples using HD and SS is the total number of examples in the synthetic subset. The number of incorrectly synthesized examples, CM, and the total number of synthesized examples, SS, are used to determine the oversampling error, i.e., each example belonging to CM is closer to one of the instances belonging to the Majority class than any example belonging to the Minority class, despite the fact that it should be the other way around.

TABLE 2. Description of the datasets used in this study.

ID	Name	No. Attributes	No. Classes	No. Minor	No. Major
1	Yeast4	10	2	51	1433
2	Yeast5	10	2	44	1440
3	Yeast6	10	2	35	1449
4	pima	8	2	268	500
5	car_good	6	2	69	1659
6	oil_spell	49	2	41	896
7	wisconsin	10	2	239	444
8	abalone-21_vs_8	8	2	14	567
9	Vehicle3	19	2	212	634

IV. DATASETS

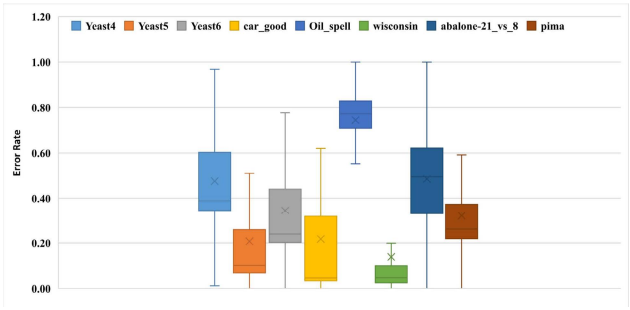
We employ nine real-life datasets to put our validation system to the test, such as Yeast4, Yeast5, and Yeast6, which are routinely used by many oversampling methods. On [188], all of the datasets are freely available. Table 2 contains information about these datasets.

Table 2 shows that the datasets have different minority and majority distributions, despite the fact that the number classes is the same. It is not necessary to address the problem with multi-class datasets to prove our counter claim, as most oversampling approaches only use binary class datasets, as shown in Table 1.

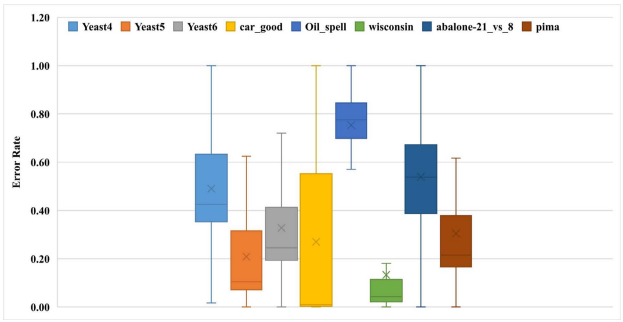
V. EXPERIMENTS AND RESULTS

In our experiments, we used all of the oversampling methods listed in Table 1 on each of the datasets listed in Table 2, after eliminating some majority examples at random. We employed varied numbers of hidden examples, namely 10%, 25%, and 50% of the majority examples of each dataset examined, to see the effect of the number of hidden examples on the validation process. Furthermore, each experiment is repeated five times, with the average of the results for each hidden ratio for each oversampling method on each dataset being reported. Table 3 shows the number of erroneous synthetic examples (NE), which are ones that are generated as minority examples but appear to be more comparable to majority examples, as the proposed validation system suggests. It also shows the number of synthetic examples (SE) generated by each oversampling method, in addition to the error rate (ER) which is calculated using Equation 3. All the result reported in Table 3 were obtained by hiding only 10% of the majority examples. Table 4 continues the results of Table 3 on the rest of the datasets.

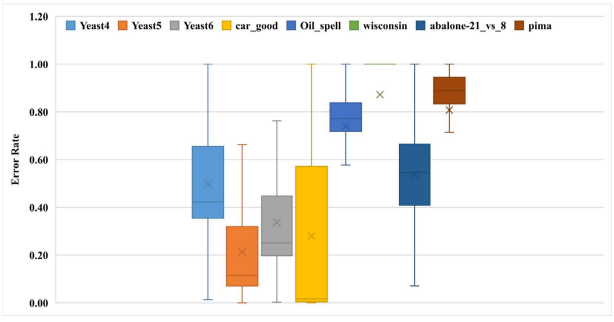
The averages of five trials on each dataset for each approach are provided in Table 3 and Table 4. In addition, for each approach, the average error is calculated for the error rates the eight datasets mentioned in Tables 3 and 4. The last column in Table 4 shows the average rank of each method based on the eight used datasets; the lower the rank, the better the oversampling performance; for example, rank 1 shall be awarded to the method with the smallest error rate.



(a) 10%



(b) 25%



(c) 50%

FIGURE 3. Box plot of the average error rates of all oversampling methods on different datasets with varied hidden percentages.

The thorough examination of Tables 3 and 4 demonstrates that all oversampling methods result in errors in the synthesized examples. That is, they generate examples that are

TABLE 3. Oversampling methods' validation results on the used datasets using 10% hidden percent.

	Yeast4			Yeast5			Yeast6			car_good		
Method	**NE**	**SE**	**ER**	**NE**	**SE**	**ER**	**NE**	**SE**	**ER**	**NE**	**NA**	**ER**
M1	463.2	1239	0.37	108	1252	0.09	294	1270	0.23	53	1425	0.04
M2	470	1235	0.38	117	1252	0.09	279	1270	0.22	46	1425	0.03
M3	414.2	1130.2	0.37	91	1222	0.07	266	1187	0.22	47	1406	0.03
M4	270.6	1239	0.22	277	1252	0.22	391	1270	0.31	60	1425	0.04
M5	471	1239	0.38	597	1252	0.48	757	1270	0.60	368	1425	0.26
M6	568.8	1239	0.46	219	1252	0.17	534	1270	0.42	53	1425	0.04
M7	50	50	1.00	42	43	0.98	34	34	1.00	66	68	0.97
M8	482.6	1239	0.39	145	1252	0.12	258	1270	0.20	11	1425	0.01
M9	765	1224	0.63	204	1232	0.17	244	1260	0.19	873	1449	0.60
M10	53	59	0.90	9	11	0.82	17	17	1.00	36	204	0.18
M11	523.4	1239	0.42	165	1252	0.13	372	1270	0.29	190	1425	0.13
M12	109.8	125.2	0.88	-	-	-	55	55	1.00	-	-	-
M13	363	1239	0.29	261	1252	0.21	308	1270	0.24	884	1425	0.62
M14	425.8	771.4	0.55	-	-	-	250	648	0.39	237	992	0.24
M15	479	1239	0.39	116	1252	0.09	357	1270	0.28	57	1425	0.04
M16	285	286	1.00	289	292	0.99	298	300	0.99	304	306	0.99
M17	309.8	1239	0.25	45	1252	0.04	79	1270	0.06	23	1425	0.02
M18	478.8	1239	0.39	103	1252	0.08	269	1270	0.21	59	1425	0.04
M19	329	1239	0.27	26	1252	0.02	124	1270	0.10	64	1425	0.04
M20	938.8	2478	0.38	0	43	0.00	575	2540	0.23	0	359	0.00
M21	729.4	1239	0.59	221	1252	0.18	385	1270	0.30	513	1425	0.36
M22	282.6	1239	0.23	-	-	-	406	1270	0.32	54	1425	0.04
M23	775.8	1239	0.63	330	1252	0.26	621	1270	0.49	290	1425	0.20
M24	52	1239	0.04	120	1252	0.10	41	1270	0.03	48	1425	0.03
M25	867	1239	0.70	456	1252	0.36	707	1270	0.56	744	1425	0.52
M26	442	1239	0.36	76	1252	0.06	246	1270	0.19	41	1425	0.03
M27	475.4	1239	0.38	86	1252	0.07	265	1270	0.21	60	1425	0.04
M28	470.2	1239	0.38	121	1252	0.10	303	1270	0.24	49	1425	0.03
M29	413.8	1239	0.33	167	1252	0.13	272	1270	0.21	163	1425	0.11
M30	644.8	1239	0.52	223	1252	0.18	369	1270	0.29	81	1425	0.06
M31	872.2	1239	0.70	395	1252	0.32	599	1270	0.47	791	1425	0.56
M32	927.8	1239	0.75	698	1252	0.56	987	1270	0.78	467	1425	0.33
M33	1041.6	2478	0.42	315	2504	0.13	813	2540	0.32	96	2850	0.03
M34	1236	1239	1.00	523	1252	0.42	36	1270	0.03	1425	1425	1.00
M35	504	791	0.64	148	807	0.18	159	841	0.19	369	668	0.55
M36	602	1239	0.49	191	1252	0.15	414	1270	0.33	248	1425	0.17
M37	518	1239	0.42	110	1252	0.09	292	1270	0.23	54	1425	0.04
M38	1200	1239	0.97	855	1252	0.68	1264	1270	1.00	1283	1425	0.90
M39	307	1226	0.25	129	1248	0.10	93	1262	0.07	467	1409	0.33
M40	1199	1239	0.97	639	1252	0.51	1262	1270	0.99	-	-	-
M41	938	1264	0.74	547	1268	0.43	711	1272	0.56	713	1430	0.50
M42	1239	1239	1.00	1010	1251	0.81	1271	1271	1.00	1424	1424	1.00
M43	78	153	0.51	32	132	0.24	47	105	0.45	72	207	0.35
M44	288	1239	0.23	188	1252	0.15	89	1270	0.07	14	1425	0.01
M45	710	1239	0.57	72	1252	0.06	139	1270	0.11	70	1425	0.05
M46	440	1239	0.36	85	1252	0.07	278	1270	0.22	68	1425	0.05
M47	753	1239	0.61	363	1252	0.29	633	1270	0.50	120	1425	0.08
M48	302	1239	0.24	35	1252	0.03	32	1270	0.03	23	1425	0.02
M49	476	1239	0.38	123	1252	0.10	308	1270	0.24	49	1425	0.03
M50	451	1239	0.36	85	1252	0.07	256	1270	0.20	49	1425	0.03
M51	14	1239	0.01	2	1252	0.00	0	1270	0.00	0	1425	0.00

TABLE 3. *(Continued.)* Oversampling methods' validation results on the used datasets using 10% hidden percent.

M52	534	1239	0.43	85	1252	0.07	296	1270	0.23	15	1425	0.01
M53	413	1239	0.33	18	1252	0.01	285	1270	0.22	12	1425	0.01
M54	452	1239	0.36	103	1252	0.08	320	1270	0.25	55	1425	0.04
M55	225	592	0.38	76	869	0.09	169	724	0.23	26	979	0.03
M56	592	1239	0.48	453	1252	0.36	441	1270	0.35	211	1425	0.15
M57	470	1239	0.38	116	1252	0.09	304	1270	0.24	48	1425	0.03
M58	232	411	0.56	69	224	0.31	95	148	0.64	85	346	0.25
M59	64	165	0.39	12	133	0.09	18	116	0.16	5	234	0.02
M60	552	1482	0.37	115	1500	0.08	279	1270	0.22	59	1425	0.04
M61	212	1239	0.17	73	1252	0.06	128	1270	0.10	28	1425	0.02
M62	140	1239	0.11	84	1252	0.07	28	1270	0.02	241	1425	0.17
M63	275	805	0.34	97	1101	0.09	182	789	0.23	282	5700	0.05
M64	579	1235	0.47	236	1250	0.19	257	1270	0.20	1373	1425	0.96
M65	558	1239	0.45	143	1252	0.11	307	1270	0.24	413	1425	0.29
M66	521	1239	0.42	101	1252	0.08	275	1270	0.22	58	1425	0.04
M67	918	1239	0.74	727	1252	0.58	1025	1270	0.81	1260	1425	0.88
M68	365	1239	0.29	140	1252	0.11	100	1270	0.08	238	1425	0.17
M69	496	1239	0.40	104	1252	0.08	326	1270	0.26	67	1425	0.05
M70	1048	1240	0.85	898	1253	0.72	1076	1270	0.85	851	1449	0.59
M71	381	1239	0.31	-	-	-	105	1270	0.08	18	1425	0.01
M72	411	1239	0.33	46	1252	0.04	498	1270	0.39	22	1425	0.02

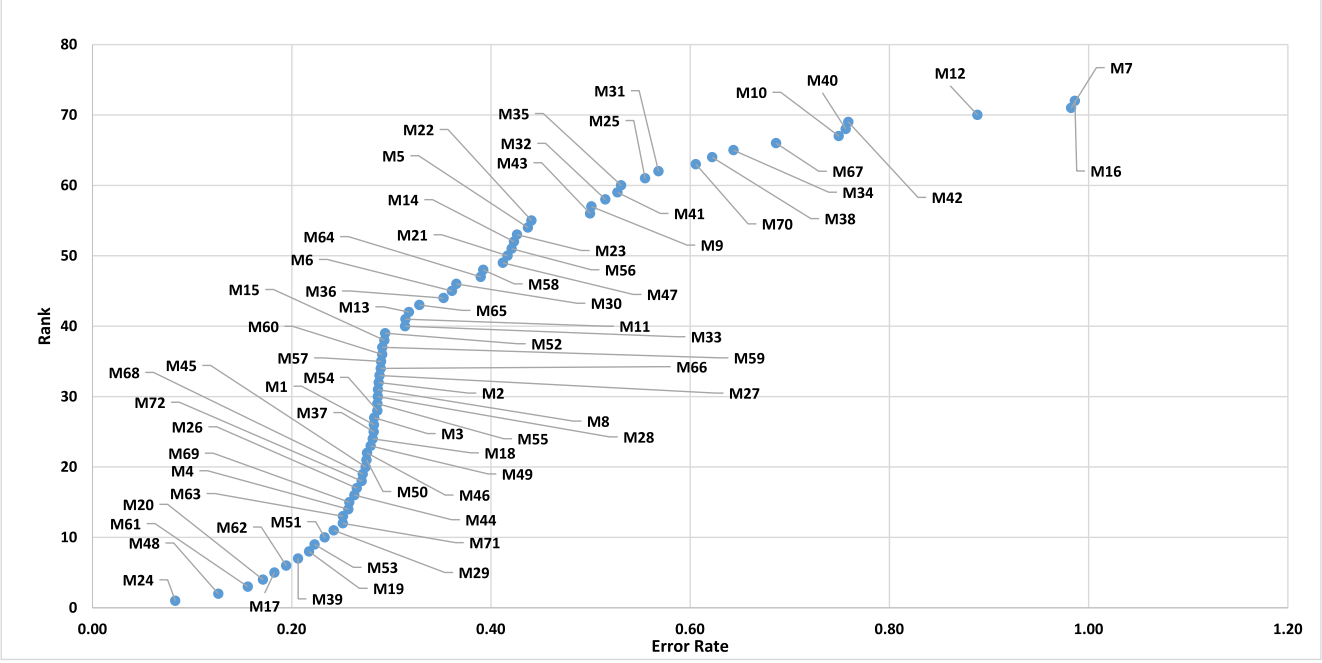


FIGURE 4. Methods ranking based on their average error rates on the datasets mentioned in Tables 3 and 4.

meant to be minority, yet are similar to the majority or fall within the majority class's decision boundary. Despite the fact that all methods generate such examples, the quantity of fake examples generated differs from one method to another. On the Yeast4 dataset, for example, the oversampling method (M51) generates 14 incorrect examples, whereas other methods, such as M70, generate more than 1K incorrect examples. That is why M51 is ranked first, whereas M70 is ranked much higher. Similar findings were achieved when 25% and 50% of the majority examples were used as hidden examples, thus there is no need to include them in tables; however, we show them in Figure 3.

The average error rate of all oversampling methods increases somewhat as the hidden percent increases, as seen

TABLE 4. Oversampling methods' validation results on the used datasets using 10% hidden percent.

	oil			wisconsin			abalone-21_vs_8			pima				
Method	**NE**	**NA**	**ER**	**NE**	**NA**	**ER**	**NE**	**NA**	**ER**	**NE**	**NA**	**ER**	**Avg. ER**	**Rank**
M1	601	766	0.78	4	161	0.02	250	497	0.50	40	182	0.22	0.28	**26**
M2	512	668	0.77	6	157	0.04	239	493	0.48	34	120	0.28	0.29	**32**
M3	446	584	0.76	5	139	0.04	230	478	0.48	-	-	-	0.28	**27**
M4	443	766	0.58	7	161	0.04	164	497	0.33	57	182	0.31	0.26	**14**
M5	437	766	0.57	59	161	0.37	250	497	0.50	63	182	0.35	0.44	**54**
M6	632	766	0.83	16	161	0.10	277	497	0.56	57	182	0.31	0.36	**45**
M7	40	40	1.00	235	238	0.99	13	13	1.00	247	267	0.93	0.98	**71**
M8	578	766	0.75	3	161	0.02	271	497	0.55	47	182	0.26	0.29	**31**
M9	758	779	0.97	12	239	0.05	326	504	0.65	201	268	0.75	0.50	**57**
M10	66	71	0.93	23	35	0.66	24	26	0.92	26	44	0.59	0.75	**67**
M11	591	766	0.77	3	161	0.02	238	497	0.48	48	182	0.26	0.31	**41**
M12	79	79	1.00	16	23	0.70	-	-	-	110	127	0.87	0.89	**70**
M13	449	766	0.59	4	161	0.02	164	497	0.33	43	182	0.24	0.32	**42**
M14	557	714	0.78	8	103	0.08	209	293	0.71	35	164	0.21	0.42	**52**
M15	578	766	0.75	3	161	0.02	252	497	0.51	48	182	0.26	0.29	**38**
M16	199	199	1.00	23	23	1.00	116	116	1.00	22	24	0.92	0.99	**72**
M17	513	766	0.67	0	161	0.00	144	497	0.29	25	182	0.14	0.18	**5**
M18	592	766	0.77	1	161	0.01	239	497	0.48	49	182	0.27	0.28	**24**
M19	568	766	0.74	6	161	0.04	155	497	0.31	40	182	0.22	0.22	**8**
M20	116	160	0.73	2	263	0.01	5	160	0.03	0	3	0.00	0.17	**4**
M21	673	766	0.88	18	161	0.11	269	497	0.54	68	182	0.37	0.42	**50**
M22	489	766	0.64	13	17	0.76	153	497	0.31	104	132	0.79	0.44	**55**
M23	631	766	0.82	16	161	0.10	321	497	0.65	47	182	0.26	0.43	**53**
M24	103	766	0.13	11	161	0.07	19	497	0.04	40	182	0.22	0.08	**1**
M25	734	766	0.96	41	161	0.25	284	497	0.57	93	182	0.51	0.55	**61**
M26	582	766	0.76	2	161	0.01	252	497	0.51	37	182	0.20	0.27	**17**
M27	585	766	0.76	3	161	0.02	263	497	0.53	53	182	0.29	0.29	**33**
M28	585	766	0.76	8	161	0.05	242	497	0.49	44	182	0.24	0.29	**30**
M29	458	766	0.60	5	161	0.03	121	497	0.24	49	182	0.27	0.24	**11**
M30	636	766	0.83	14	161	0.09	272	497	0.55	75	182	0.41	0.37	**46**
M31	730	766	0.95	57	161	0.35	336	497	0.68	94	182	0.52	0.57	**62**
M32	559	766	0.73	8	161	0.05	322	497	0.65	51	182	0.28	0.51	**58**
M33	1238	1532	0.81	9	322	0.03	502	994	0.51	98	364	0.27	0.31	**40**
M34	733	766	0.96	0	161	0.00	497	497	1.00	136	182	0.75	0.64	**65**
M35	706	724	0.98	-	-	-	152	235	0.65	-	-	-	0.53	**60**
M36	564	766	0.74	11	161	0.07	291	497	0.59	53	182	0.29	0.35	**44**
M37	582	766	0.76	5	161	0.03	236	497	0.47	40	182	0.22	0.28	**25**
M38	-	-	-	4	161	0.02	389	497	0.78	0	182	0.00	0.62	**64**
M39	354	748	0.47	14	159	0.09	42	491	0.09	39	159	0.25	0.21	**7**
M40	614	766	0.80	74	161	0.46	445	497	0.90	121	182	0.66	0.76	**68**
M41	701	768	0.91	0	162	0.00	391	510	0.77	56	183	0.31	0.53	**59**
M42	494	763	0.65	40	163	0.25	497	497	1.00	71	192	0.37	0.76	**69**
M43	107	123	0.87	70	120	0.58	16	42	0.38	446	726	0.61	0.50	**56**
M44	688	766	0.90	16	161	0.10	216	497	0.43	38	182	0.21	0.26	**16**
M45	524	766	0.68	6	161	0.04	209	497	0.42	48	182	0.26	0.27	**20**
M46	595	766	0.78	8	161	0.05	222	497	0.45	44	182	0.24	0.28	**22**
M47	646	766	0.84	16	161	0.10	240	497	0.48	71	182	0.39	0.41	**49**
M48	-	-	-	32	161	0.20	125	497	0.25	22	182	0.12	0.13	**2**
M49	598	766	0.78	3	161	0.02	243	497	0.49	34	182	0.19	0.28	**23**
M50	604	766	0.79	11	161	0.07	246	497	0.49	33	182	0.18	0.28	**21**
M51	765	766	1.00	0	161	0.00	356	497	0.72	25	182	0.14	0.23	**10**

TABLE 4. *(Continued.)* **Oversampling methods' validation results on the used datasets using 10% hidden percent.**

M52	572	766	0.75	5	161	0.03	287	497	0.58	46	182	0.25	0.29	**39**
M53	542	766	0.71	0	161	0.00	129	497	0.26	43	182	0.24	0.22	**9**
M54	608	766	0.79	4	161	0.02	257	497	0.52	39	182	0.21	0.29	**28**
M55	514	688	0.75	2	72	0.03	71	129	0.55	75	319	0.24	0.29	**29**
M56	719	766	0.94	82	161	0.51	88	497	0.18	74	182	0.41	0.42	**51**
M57	597	766	0.78	5	161	0.03	259	497	0.52	44	182	0.24	0.29	**35**
M58	123	151	0.81	5	148	0.03	3	26	0.12	46	111	0.41	0.39	**48**
M59	125	162	0.77	13	639	0.02	27	42	0.64	179	750	0.24	0.29	**37**
M60	650	805	0.81	12	168	0.07	281	571	0.49	53	216	0.25	0.29	**36**
M61	423	766	0.55	5	161	0.03	55	497	0.11	37	182	0.20	0.16	**3**
M62	546	766	0.71	10	161	0.06	119	497	0.24	31	182	0.17	0.19	**6**
M63	2492	3064	0.81	9	491	0.02	-	-	-	148	678	0.22	0.25	**13**
M64	-	-	-	14	160	0.09	256	495	0.52	54	180	0.30	0.39	**47**
M65	535	766	0.70	13	161	0.08	247	497	0.50	46	182	0.25	0.33	**43**
M66	618	766	0.81	9	161	0.06	225	497	0.45	44	182	0.24	0.29	**34**
M67	739	766	0.96	60	161	0.37	315	497	0.63	92	182	0.51	0.69	**66**
M68	604	766	0.79	22	161	0.14	109	497	0.22	68	182	0.37	0.27	**19**
M69	506	766	0.66	4	161	0.02	206	497	0.41	32	182	0.18	0.26	**15**
M70	570	779	0.73	14	239	0.06	389	497	0.78	74	268	0.28	0.61	**63**
M71	624	766	0.81	6	161	0.04	24	497	0.05	83	182	0.46	0.25	**12**
M72	511	766	0.67	6	161	0.04	243	497	0.49	35	182	0.19	0.27	**18**

in Figure 3. This is logical since when oversampling methods synthesize their minority examples, they become unaware of some majority examples; in fact, we expected a significant error rise as the size of the hidden subset grew larger. In terms of the effect of the dataset on the average oversampling error, we can observe in the same figure that some datasets, such as (Yeast5), are easier to be oversampled than others, such as (Yeast6) and (Yeast4). However, the difference is not substantial, and more importantly, as the Box plots show, the standard deviation of the error rates produced by all oversampling methods on each dataset is extremely high.

In order to compare oversampling methods, we ranked them according to their average error across eight datasets. The average ranks of the methods are plotted against the average errors they produced on all used datasets as shown in Figure 4.

Despite the fact that all of the methods discussed generate false examples, Figure 4 indicates that certain methods do better than others in avoiding the formation of false examples. As a result, if oversampling is unavoidable, the method's reliability should be verified using a validation tools such as ours. Methods like M24, for example, have error rates close to 0%, whereas others like M7 and M16 have error rates near to 99%.

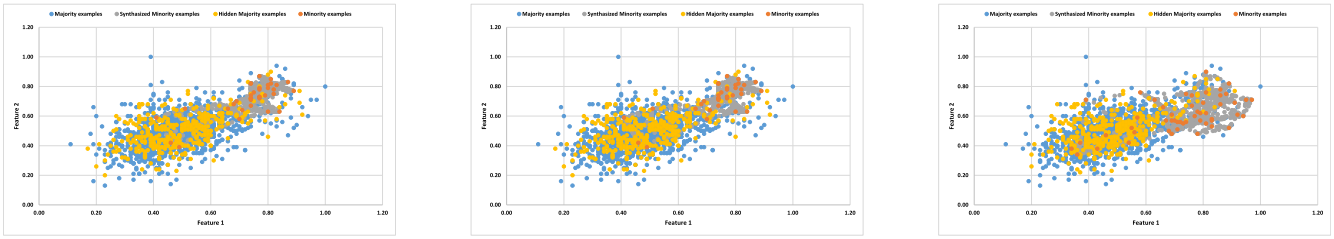
As a result, all Oversampling methods validated produce misleading examples, regardless of the hidden percentage or dataset used. Figure 5 visualizes the Oversampling results on eight datasets using M1, displaying various sorts of examples, including hidden, minority, majority, and synthesized examples while hiding 10% of the majority examples. For the sake of illustration, we limited the data to only two features.

As seen in Figure 5, many synthetic examples are created on the basis of or near hidden examples, producing almost identical feature values. Even in higher dimensional feature space, such a situation has the potential to occur. The common mistake that all oversampling methods make is to feed such data to a classifier, assuming that all of the examples are realistic and labeled based on reality. The classifier has no other knowledge and learns based on the false assumption, which produces excellent results in labs but unexpected behavior in real-world scenarios.

The results shown thus far do not necessarily imply that incorrect example synthesis occurs just when the majority examples are hidden from the oversampling method. Even though the majority examples are completely visible to the methods, some methods generate false examples. The published findings of all of the oversampling methods demonstrate this, as none of them claimed to be an accurate method with no errors.

We validated some of the best performers on a ninth machine learning dataset, Vehicle3, because some oversampling approaches passed our validation test by presenting a relatively small number of unrealistic examples, and to further support our counterclaim against the validity of the oversampling approach in general. The validation results some of the good performers are shown in Table 5.

As can be seen in Table 5, when we changed the dataset, the errors of the ''best'' oversampling methods increased significantly, demonstrating once again that these oversampling methods fill in the features space gap without considering whether the generated examples are truly belong to the minority, and falsely consider them as such. This makes the training



(a) Yeast4 (b) Yeast5 (c) Yeast6

FIGURE 5. Visualization of validating the SMOTE method on three datasets. When the figure is scaled up, more information is obtained.

TABLE 5. ER on vehicle3 dataset using 25% hidden data and some of the methods with the least ER.

Method	NE	SE	ER
M51	168	264	0.64
M24	88	264	0.33
M62	122	264	0.46
M61	61	264	0.23
M19	91	264	0.34
M39	102	236	0.43
M44	194	264	0.73
M68	149	264	0.56

of these examples deceptive, and it could lead to the classifier being overfitted on incorrect data if robust generalization techniques are not used. As a result, when applied to real-world tasks, it is possible that the entire machine learning system fails spectacularly, particularly in critical applications such as security, autonomous driving, aviation safety and medical applications, where even one unrealistic false synthesized example could do catastrophic harm.

VI. CONCLUSION

Oversampling methods have been used and developed for decades to handle the problem of class imbalance learning, and there is a near exponential growing trend for such type of research. The main question of this research is oversampling approach in its current form and methods provide applicable and viable solution for learning from class imbalance data? We claim that the current oversampling approach is deceptive and could lead to severe failures in real-world applications. In order to answer the main question and to prove our counterclaim, we reviewed a large number of oversampling methods and analyzed their performance in terms of providing unrealistic examples, for this purpose we propose a new validation system for oversampling methods, which we utilized to validate over 70 different oversampling methods. Our validation results on nine real-world common datasets reveal that all of the oversampling methods investigated generate false examples, assuming that they are minorities when they are not, causing classifiers to perform well in labs but more likely fail in practice.

The Oversampling methods investigated in this paper are ranked according to how many incorrect examples they generate. When used to solve real-life problems, the ranking shows that some methods are less harmful than others. When the datasets were changed, however, they were found to produce intolerable number of errors. Therefore, we recommend avoiding such methods when dealing with sensitive applications such as security, autonomous driving, aviation safety, and medical applications that use machine learning from class imbalanced data. Instead, we seriously encourage using ensemble approaches to problems of class imbalance, such as Easy Ensemble [189], Random Data Partitioning [71], etc. Because these methods do not create data out of thin air and do not, as the Undersampling approach suggests, deny the learning process from critical data.

More research should be done in the future to confirm the validity or invalidity of oversampling approach, investigating more methods and incorporating more data. Furthermore, we recommend that additional research be conducted on real-world applications, including measurements of incorrect predictions made with and without the use of oversampling methods, as well as comparisons with ensemble methods.

REFERENCES

[1] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.

[2] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 42, no. 4, pp. 463–484, Jul. 2011.

[3] G. M. Weiss, "Foundations of imbalanced learning," in *Imbalanced Learning: Foundations, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley, 2013, pp. 13–41.

[4] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Learning from class-imbalanced data with a model-agnostic framework for machine intelligent diagnosis," *Rel. Eng. Syst. Saf.*, vol. 216, Dec. 2021, Art. no. 107934.

[5] M. Peng, Q. Zhang, X. Xing, T. Gui, X. Huang, Y.-G. Jiang, K. Ding, and Z. Chen, "Trainable undersampling for class-imbalance learning," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4707–4714.

[6] A. S. Tarawneh, A. B. A. Hassanat, K. Almohammadi, D. Chetverikov, and C. Bellinger, "SMOTEFUNA: Synthetic minority over-sampling technique based on furthest neighbour algorithm," *IEEE Access*, vol. 8, pp. 59069–59082, 2020.

[7] R. F. Mansour, S. Abdel-Khalek, I. Hilali-Jaghdam, J. Nebhen, W. Cho, and G. P. Joshi, "An intelligent outlier detection with machine learning empowered big data analytics for mobile edge computing," *Cluster Comput.*, pp. 1–13, Nov. 2021.

[8] N. O. Aljehane and R. F. Mansour, "Optimal allocation of renewable energy source and charging station for PHEVs," *Sustain. Energy Technol. Assessments*, vol. 49, Feb. 2022, Art. no. 101669.

[9] R. F. Mansour, J. Escorcia-Gutierrez, M. Gamarra, V. G. Díaz, D. Gupta, and S. Kumar, "Artificial intelligence with big data analytics-based brain intracranial hemorrhage E-diagnosis using CT images," *Neural Comput. Appl.*, pp. 1–13, Jun. 2021.

[10] A. B. A. Hassanat, "Two-point-based binary search trees for accelerating big data classification using KNN," *PLoS ONE*, vol. 13, no. 11, Nov. 2018, Art. no. e0207772.
[11] A. Hassanat, "Norm-based binary search trees for speeding up KNN big data classification," *Computers*, vol. 7, no. 4, p. 54, Oct. 2018.
[12] A. Hassanat, "Furthest-pair-based decision trees: Experimental results on big data classification," *Information*, vol. 9, no. 11, p. 284, Nov. 2018.
[13] A. B. A. Hassanat, "Furthest-pair-based binary search tree for speeding big data classification using K-nearest neighbors," *Big Data*, vol. 6, no. 3, pp. 225–235, Sep. 2018.
[14] A. B. Hassanat and S. Jassim, "Visual words for lip-reading," *Proc. SPIE*, vol. 7708, Apr. 2010, Art. no. 77080B.
[15] A. B. Hassanat, "Visual speech recognition," *Speech Lang. Technol.*, vol. 1, pp. 279–303, Jan. 2011.
[16] A. B. A. Hassanat, E. Btoush, M. A. Abbadi, B. M. Al-Mahadeen, M. Al-Awadi, K. I. A. Mseidein, A. M. Almseden, A. S. Tarawneh, M. B. Alhasanat, V. B. S. Prasath, and F. A. Al-alem, "Victory sign biometrie for terrorists identification: Preliminary results," in *Proc. 8th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2017, pp. 182–187.
[17] A. B. A. Hassanat, "On identifying terrorists using their victory signs," *Data Sci. J.*, vol. 17, p. 27, Oct. 2018.
[18] A. S. Tarawneh, D. Chetverikov, C. Verma, and A. B. Hassanat, "Stability and reduction of statistical features for image classification and retrieval: Preliminary results," in *Proc. 9th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2018, pp. 117–121.
[19] M. Z. Al-Shamaileh, A. B. Hassanat, A. S. Tarawneh, M. Sohel Rahman, C. Celik, and M. Jawthari, "New online/offline text-dependent Arabic handwriting dataset for writer authentication and identification," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jun. 2019, pp. 116–121.
[20] A. Hassanat, M. Al-Awadi, E. Btoush, A. Al-Btoush, E. Alhasanat, and G. Altarawneh, "New mobile phone and webcam hand images databases for personal authentication and identification," *Procedia Manuf.*, vol. 3, pp. 4060–4067, Mar. 2015.
[21] A. I. Al-Btoush, M. A. Abbadi, A. B. Hassanat, A. S. Tarawneh, A. Hasanat, and V. B. S. Prasath, "New features for eye-tracking systems: Preliminary results," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jun. 2019, pp. 179–184.
[22] A. B. Hassanat, V. S. Prasath, B. M. Al-Mahadeen, and S. M. M. Alhasanat, "Classification and gender recognition from veiled-faces," *Int. J. Biometrics*, vol. 9, no. 4, pp. 347–364, 2017.
[23] H. Xu, C. Zhang, G. S. Hong, J. Zhou, J. Hong, and K. S. Woon, "Gated recurrent units based neural network for tool condition monitoring," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.
[24] F. Ugo, D. S. Alfredo, P. Francesca, Z. Paolo, and P. Francesco, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Inf. Sci.*, vol. 479, pp. 448–455, Apr. 2019.
[25] N. Ghatasheh, H. Faris, R. Abukhurma, P. A. Castillo, N. Al-Madi, A. M. Mora, A. M. Al-Zoubi, and A. Hassanat, "Cost-sensitive ensemble methods for bankruptcy prediction in a highly imbalanced data distribution: A real case from the Spanish market," *Prog. Artif. Intell.*, vol. 9, no. 4, pp. 361–375, Dec. 2020.
[26] A. S. Tarawneh, A. B. Hassanat, C. Celik, D. Chetverikov, M. S. Rahman, and C. Verma, "Deep face image retrieval: A comparative study with dictionary learning," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jun. 2019, pp. 185–192.
[27] A. S. Tarawneh, C. Celik, A. B. Hassanat, and D. Chetverikov, "Detailed investigation of deep features with sparse representation and dimensionality reduction in CBIR: A comparative study," *Intell. Data Anal.*, vol. 24, no. 1, pp. 47–68, 2020.
[28] M. Hammad, M. H. Alkinani, B. B. Gupta, and A. A. Abd El-Latif, "Myocardial infarction detection based on deep neural network on imbalanced data," *Multimedia Syst.*, pp. 1–13, Jan. 2021.
[29] M. Fatima and M. Pasha, "Survey of machine learning algorithms for disease diagnostic," *J. Intell. Learn. Syst. Appl.*, vol. 9, no. 1, pp. 1–16, 2017.
[30] A. Alqatawneh, R. Alhalaseh, A. Hassanat, and M. Abbadi, "Statistical-hypothesis-aided tests for epilepsy classification," *Computers*, vol. 8, no. 4, p. 84, Nov. 2019.
[31] M. Aseeri, A. B. Hassanat, and S. Mnasri, "Modelling-based simulator for forecasting the spread of COVID-19: A case study of Saudi Arabia," *Int. J. Comput. Sci. Netw. Secur.*, vol. 20, pp. 114–125, 2020.
[32] A. B. Hassanat, S. Mnasri, M. A. Aseeri, K. Alhazmi, O. Cheikhrouhou, G. Altarawneh, M. Alrashidi, A. S. Tarawneh, K. S. Almohammadi, and H. Almoamari, "A simulation model for forecasting COVID-19 pandemic spread: Analytical results based on the current Saudi COVID-19 data," *Sustainability*, vol. 13, no. 9, p. 4888, Apr. 2021.
[33] S. Mnasri, A. Van Den Bossche, N. Nasri, and T. Val, "The 3D redeployment of nodes in wireless sensor networks with real testbed prototyping," in *Proc. Int. Conf. Ad-Hoc Netw. Wireless*. Cham, Switzerland: Springer, 2017, pp. 18–24.
[34] S. Mnasri, N. Nasri, and T. Val, "The 3D indoor deployment in DL-IoT with experimental validation using a particle swarm algorithm based on the dialects of songs," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 928–933.
[35] S. Mnasri, A. Van den Bossche, N. Nasri, and T. Val, "The 3D deployment multi-objective problem in mobile WSN: Optimizing coverage and localization," *Int. Res. J. Innov. Eng.*, vol. 1, no. 5, pp. 1–15, 2015.
[36] S. Mnasri, N. Nasri, M. Alrashidi, A. van den Bossche, and T. Val, "IoT networks 3D deployment using hybrid many-objective optimization algorithms," *J. Heuristics*, vol. 26, no. 5, pp. 663–709, Oct. 2020.
[37] W. Abdallah and T. Val, "Genetic-Voronoi algorithm for coverage of IoT data collection networks," in *Proc. 30th Int. Conf. Comput. Theory Appl. (ICCTA)*, Dec. 2020, pp. 16–22.
[38] W. Abdallah, S. Mnasri, N. Nasri, and T. Val, "Emergent IoT wireless technologies beyond the year 2020: A comprehensive comparative analysis," in *Proc. Int. Conf. Comput. Inf. Technol. (ICCIT)*, Sep. 2020, pp. 1–5.
[39] S. Mnasri, N. Nasri, A. van den Bossche, and T. Val, "A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem," *ISA Trans.*, vol. 91, pp. 262–280, Aug. 2019.
[40] S. Mnasri, F. Abbes, K. Zidi, and K. Ghedira, "A multi-objective hybrid BCRC-NSGAII algorithm to solve the VRPTW," in *Proc. 13th Int. Conf. Hybrid Intell. Syst. (HIS)*, Dec. 2013, pp. 60–65.
[41] S. Tlili, S. Mnasri, and T. Val, "A multi-objective gray wolf algorithm for routing in IoT collection networks with real experiments," in *Proc. Nat. Comput. Colleges Conf. (NCCC)*, Mar. 2021, pp. 1–5.
[42] S. Mnasri, N. Nasri, A. Van Den Bossche, and T. Val, "A hybrid ant-genetic algorithm to solve a real deployment problem: A case study with experimental validation," in *Proc. Int. Conf. Ad-Hoc Netw. Wireless*. Cham, Switzerland: Springer, 2017, pp. 367–381.
[43] S. Mnasri, N. Nasri, A. Van Den Bossche, and T. Val, "A comparative analysis with validation of NSGA-III and MOEA/D in resolving the 3D indoor redeployment problem in DL-IoT," in *Proc. Int. Conf. Internet Things, Embedded Syst. Commun. (IINTEC)*, Oct. 2017, pp. 15–20.
[44] M. Alghamdi and W. Teahan, "Experimental evaluation of Arabic OCR systems," *PSU Res. Rev.*, vol. 1, no. 3, pp. 229–241, Nov. 2017.
[45] A. B. A. Hassanat and G. Awad Altarawneh, "Rule-and dictionary-based solution for variations in written Arabic names in social networks, big data, accounting systems and large databases," *Res. J. Appl. Sci., Eng. Technol.*, vol. 8, no. 14, pp. 1630–1638, Oct. 2014.
[46] E. Hamadaqa, A. Abadleh, A. Mars, and W. Adi, "Highly secured implantable medical devices," in *Proc. Int. Conf. Innov. Inf. Technol. (IIT)*, Nov. 2018, pp. 7–12.
[47] S. Mulhem, A. Abadleh, and W. Adi, "Accelerometer-based joint user-device clone-resistant identity," in *Proc. 2nd World Conf. Smart Trends Syst., Secur. Sustainability (WorldS4)*, Oct. 2018, pp. 230–237.
[48] A. Mars, A. Abadleh, and W. Adi, "Operator and manufacturer independent D2D private link for future 5G networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr./May 2019, pp. 1–6.
[49] A. Alabadleh, S. Aljaafreh, A. Aljaafreh, and K. Alawasa, "A RSS-based localization method using HMM-based error correction," *J. Location Based Services*, vol. 12, nos. 3–4, pp. 273–285, Oct. 2018.
[50] A. Aljaafreh, K. Alawasa, S. Alja'afreh, and A. Abadleh, "Fuzzy inference system for speed bumps detection using smart phone accelerometer sensor," *J. Telecommun., Electron. Comput. Eng. (JTEC)*, vol. 9, nos. 2–7, pp. 133–136, 2017.
[51] A. Abadleh, E. Al-Hawari, E. Alkafaween, and H. Al-Sawalqah, "Step detection algorithm for accurate distance estimation using dynamic step length," in *Proc. 18th IEEE Int. Conf. Mobile Data Manage. (MDM)*, May 2017, pp. 324–327.
[52] A. Abadleh, S. Han, S. J. Hyun, B. Lee, and M. Kim, "Construction of indoor floor plan and localization," *Wireless Netw.*, vol. 22, no. 1, pp. 175–191, Jan. 2016.
[53] A. B. Hassanat, V. S. Prasath, K. I. Mseidein, M. Al-awadi, and A. M. Hammouri, "A hybridwavelet-shearlet approach to robust digital imagewatermarking," *Informatica*, vol. 41, no. 1, pp. 3–24, 2017.

[54] A. B. Hassanat and S. Jassim, "Color-based lip localization method," *Proc. SPIE*, vol. 7708, Apr. 2010, Art. no. 77080Y.

[55] A. B. A. Hassanat, M. Alkasassbeh, M. Al-awadi, and E. A. A. Alhasanat, "Color-based object segmentation method using artificial neural network," *Simul. Model. Pract. Theory*, vol. 64, pp. 3–17, May 2016.

[56] P. Narloch, A. Hassanat, A. S. Tarawneh, H. Anysz, J. Kotowski, and K. Almohammadi, "Predicting compressive strength of cement-stabilized rammed Earth based on SEM images using computer vision and deep learning," *Appl. Sci.*, vol. 9, no. 23, p. 5131, Nov. 2019.

[57] A. B. A. Hassanat, V. B. S. Prasath, M. Al-kasassbeh, A. S. Tarawneh, and A. J. Al-shamailh, "Magnetic energy-based feature extraction for low-quality fingerprint images," *Signal, Image Video Process.*, vol. 12, no. 8, pp. 1471–1478, Nov. 2018.

[58] A. B. A. Hassanat, M. Alkasassbeh, M. Al-awadi, and E. A. A. Alhasanat, "Colour-based lips segmentation method using artificial neural networks," in *Proc. 6th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2015, pp. 188–193.

[59] M. Al-kasassbeh and T. Khairallah, "Winning tactics with DNS tunnelling," *Netw. Secur.*, vol. 2019, no. 12, pp. 12–19, Dec. 2019.

[60] G. Al-Naymat, M. Al-Kasassbeh, and E. Al-Harwari, "Using machine learning methods for detecting network anomalies within SNMP-MIB dataset," *Int. J. Wireless Mobile Comput.*, vol. 15, no. 1, pp. 67–76, 2018.

[61] A. A. Zuraiq and M. Alkasassbeh, "Phishing detection approaches," in *Proc. 2nd Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2019, pp. 1–6.

[62] M. Almseidin, A. Abu Zuraiq, M. Al-kasassbeh, and N. Alnidami, "Phishing detection based on machine learning and feature selection methods," *Int. J. Interact. Mobile Technol. (iJIM)*, vol. 13, no. 12, p. 171, Dec. 2019.

[63] A. Abuzuraiq, M. Alkasassbeh, and M. Almseidin, "Intelligent methods for accurately detecting phishing websites," in *Proc. 11th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2020, pp. 085–090.

[64] M. Almseidin, I. Piller, M. Al-Kasassbeh, and S. Kovacs, "Fuzzy automaton as a detection mechanism for the multi-step attack," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 9, no. 2, pp. 575–586, 2019.

[65] M. Al-Kasassbeh, S. Mohammed, M. Alauthman, and A. Almomani, "Feature selection using a machine learning to classify a malware," in *Handbook of Computer Networks and Cyber Security* Cham, Switzerland: Springer, 2020, pp. 889–904.

[66] M. Almseidin, M. Al-Kasassbeh, and S. Kovacs, "Detecting slow port scan using fuzzy rule interpolation," in *Proc. 2nd Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2019, pp. 1–6.

[67] Z. Alothman, M. Alkasassbeh, and S. Al-Haj Baddar, "An efficient approach to detect IoT botnet attacks using machine learning," *J. High Speed Netw.*, vol. 26, no. 3, pp. 241–254, Nov. 2020.

[68] A. Rawashdeh, M. Alkasassbeh, and M. Al-Hawawreh, "An anomaly-based approach for DDoS attack detection in cloud environment," *Int. J. Comput. Appl. Technol.*, vol. 57, no. 4, pp. 312–324, 2018.

[69] M. Alkasassbeh, "A novel hybrid method for network anomaly detection based on traffic prediction and change point detection," 2018, *arXiv:1801.05309*.

[70] S. Wang, W. Jiang, and K.-L. Tsui, "Adjusted support vector machines based on a new loss function," *Ann. Oper. Res.*, vol. 174, no. 1, pp. 83–101, Feb. 2010.

[71] A. B. Hassanat, A. S. Tarawneh, S. S. Abed, G. A. Altarawneh, M. Alrashidi, and M. Alghamdi, "RDPVR: Random data partitioning with voting rule for machine learning from class-imbalanced datasets," *Electronics*, vol. 11, no. 2, p. 228, Jan. 2022.

[72] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning From Imbalanced Data Sets*, vol. 10. Springer, 2018.

[73] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 1–50, Nov. 2016.

[74] K. K. Hauner, R. E. Zinbarg, and W. Revelle, "A latent variable model approach to estimating systematic bias in the oversampling method," *Behav. Res. Methods*, vol. 46, no. 3, pp. 786–797, Sep. 2014.

[75] M. Al-Nashashibi, W. Hadi, N. El-Khalili, G. Issa, and A. A. AlBanna, "A new two-step ensemble learning model for improving stress prediction of automobile drivers," *Int. Arab J. Inf. Technol.*, vol. 18, no. 6, pp. 819–829, 2021.

[76] P. Fergus, M. Selvaraj, and C. Chalmers, "Machine learning ensemble modelling to classify caesarean section and vaginal delivery types using cardiotocography traces," *Comput. Biol. Med.*, vol. 93, pp. 7–16, Feb. 2018.

[77] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 559–563, 2017.

[78] G. Kovács, "Smote-variants: A Python implementation of 85 minority oversampling techniques," *Neurocomputing*, vol. 366, pp. 352–354, Nov. 2019.

[79] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, Jun. 2009.

[80] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: An experimental review," *J. Big Data*, vol. 7, no. 1, pp. 1–47, Dec. 2020.

[81] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jul. 2018.

[82] C. Drummond and R. C. Holte, "C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *Proc. Workshop Learn. Imbalanced Datasets II*, vol. 11, 2003, pp. 1–8.

[83] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 39, no. 1, pp. 281–288, Feb. 2009.

[84] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2005, pp. 878–887.

[85] R. Das, S. K. Biswas, D. Devi, and B. Sarma, "An oversampling technique by integrating reverse nearest neighbor in SMOTE: Reverse-SMOTE," in *Proc. Int. Conf. Smart Electron. Commun. (ICOSEC)*, Sep. 2020, pp. 1239–1244.

[86] C. Liu, S. Jin, D. Wang, Z. Luo, J. Yu, B. Zhou, and C. Yang, "Constrained oversampling: An oversampling approach to reduce noise generation in imbalanced datasets with class overlapping," *IEEE Access*, early access, Aug. 28, 2020, doi: 10.1109/ACCESS.2020.3018911.

[87] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2013.

[88] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.

[89] C. Bellinger, S. Sharma, N. Japkowicz, and O. R. Zaïane, "Framework for extreme imbalance classification: SWIM—Sampling with the majority class," *Knowl. Inf. Syst.*, vol. 62, pp. 841–866, Jul. 2019.

[90] C. Tian, L. Zhou, S. Zhang, and Y. Zhao, "A new majority weighted minority oversampling technique for classification of imbalanced datasets," in *Proc. Int. Conf. Big Data, Artif. Intell. Internet Things Eng. (ICBAIE)*, Jun. 2020, pp. 154–157.

[91] P. Domingos, "MetaCost: A general method for making classifiers cost-sensitive," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 1999, pp. 155–164.

[92] Y. E. Kurniawati, A. E. Permanasari, and S. Fauziati, "Adaptive synthetic-nominal (ADASYN-N) and adaptive synthetic-KNN (ADASYN-KNN) for multiclass imbalance learning on laboratory test data," in *Proc. 4th Int. Conf. Sci. Technol. (ICST)*, Aug. 2018, pp. 1–6.

[93] W. Zhang, R. Ramezani, and A. Naeim, "WOTBoost: Weighted oversampling technique in boosting for imbalanced learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 2523–2531.

[94] B. S. Raghuwanshi and S. Shukla, "SMOTE based class-specific extreme learning machine for imbalanced learning," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104814.

[95] G. Douzas and F. Bacao, "Self-organizing map oversampling (SOMO) for imbalanced data set learning," *Expert Syst. Appl.*, vol. 82, pp. 40–52, Oct. 2017.

[96] G. A. Pradipta, R. Wardoyo, A. Musdholifah, and I. N. H. Sanjaya, "Radius-SMOTE: A new oversampling technique of minority samples based on radius distance for learning from imbalanced data," *IEEE Access*, vol. 9, pp. 74763–74777, 2021.

[97] B. Krawczyk, M. Koziarski, and M. Wozniak, "Radial-based oversampling for multiclass imbalanced data classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2818–2831, Aug. 2020.

[98] J. Hong, H. Kang, and T. Hong, "Oversampling-based prediction of environmental complaints related to construction projects with imbalanced empirical-data learning," *Renew. Sustain. Energy Rev.*, vol. 134, Dec. 2020, Art. no. 110402.

[99] M. H. Ibrahim, "ODBOT: Outlier detection-based oversampling technique for imbalanced datasets learning," *Neural Comput. Appl.*, vol. 33, pp. 15781–15806, Jun. 2021.
[100] L. Wang, H. Wang, and G. Fu, "Multiple kernel learning with minority oversampling for classifying imbalanced data," *IEEE Access*, vol. 9, pp. 565–580, 2021.
[101] S. Bej, N. Davtyan, M. Wolfien, M. Nassar, and O. Wolkenhauer, "LoRAS: An oversampling approach for imbalanced datasets," *Mach. Learn.*, vol. 110, no. 2, pp. 279–301, Feb. 2021.
[102] T. Zhu, Y. Lin, and Y. Liu, "Improving interpolation-based oversampling for imbalanced data learning," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104826.
[103] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on K-means and smote," *Inf. Sci.*, vol. 465, pp. 1–20, Jun. 2018.
[104] H. Faris, R. Abukhurma, W. Almanaseer, M. Saadeh, A. M. Mora, P. A. Castillo, and I. Aljarah, "Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: A case from the Spanish market," *Prog. Artif. Intell.*, vol. 9, no. 1, pp. 31–53, Mar. 2020.
[105] Z. Jiang, J. Yang, and Y. Liu, "Imbalanced learning with oversampling based on classification contribution degree," *Adv. Theory Simul.*, vol. 4, no. 5, May 2021, Art. no. 2100031.
[106] G. Douzas, F. Bacao, J. Fonseca, and M. Khudinyan, "Imbalanced learning in land cover classification: Improving minority classes' prediction accuracy using the geometric SMOTE algorithm," *Remote Sens.*, vol. 11, no. 24, p. 3040, 2019.
[107] Y. Zhang, X. Li, L. Gao, L. Wang, and L. Wen, "Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning," *J. Manuf. Syst.*, vol. 48, pp. 34–50, Jul. 2018.
[108] Z. Wang and H. Wang, "Global data distribution weighted synthetic oversampling technique for imbalanced learning," *IEEE Access*, vol. 9, pp. 44770–44783, 2021.
[109] G. Liu, Y. Yang, and B. Li, "Fuzzy rule-based oversampling technique for imbalanced and incomplete data learning," *Knowl.-Based Syst.*, vol. 158, pp. 154–174, Oct. 2018.
[110] X. Wu, Y. Yang, and L. Ren, "Entropy difference and kernel-based oversampling technique for imbalanced data learning," *Intell. Data Anal.*, vol. 24, no. 6, pp. 1239–1255, Dec. 2020.
[111] J. Engelmann and S. Lessmann, "Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning," 2020, *arXiv:2008.09202*.
[112] Q. Li, G. Li, W. Niu, Y. Cao, L. Chang, J. Tan, and L. Guo, "Boosting imbalanced data learning with Wiener process oversampling," *Frontiers Comput. Sci.*, vol. 11, no. 5, pp. 836–851, Oct. 2017.
[113] C.-R. Wang and X.-H. Shao, "An improving majority weighted minority oversampling technique for imbalanced classification problem," *IEEE Access*, vol. 9, pp. 5069–5082, 2021.
[114] R. Malhotra and S. Kamal, "An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data," *Neurocomputing*, vol. 343, pp. 120–140, May 2019.
[115] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105662.
[116] M. R. K. Dhurjad and M. Banait, "A survey on oversampling techniques for imbalanced learning," *Int. J. Appl. Innov. Eng. Manage.*, vol. 3, no. 1, pp. 279–284, 2014.
[117] J. Li, Q. Zhu, Q. Wu, and Z. Fan, "A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors," *Inf. Sci.*, vol. 565, pp. 438–455, Jul. 2021.
[118] Z. Jiang, T. Pan, C. Zhang, and J. Yang, "A new oversampling method based on the classification contribution degree," *Symmetry*, vol. 13, no. 2, p. 194, Jan. 2021.
[119] G. E. Batista, R. C. Prati, and M. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 20–29, 2004.
[120] J. Wang, M. Xu, H. Wang, and J. Zhang, "Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding," in *Proc. 8th Int. Conf. Signal Process.*, vol. 3, 2006, pp. 1–4.
[121] J. De La Calleja and O. Fuentes, "A distance-based over-sampling method for learning from imbalanced data sets," in *Proc. FLAIRS Conf.*, 2007, pp. 634–635.
[122] S. Gazzah and N. E. B. Amara, "New oversampling approaches based on polynomial fitting for imbalanced data sets," in *Proc. 8th IAPR Int. Workshop Document Anal. Syst.*, Sep. 2008, pp. 677–684.
[123] J. Stefanowski and S. Wilk, "Selective pre-processing of imbalanced data for improving classification performance," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*. Berlin, Germany: Springer, 2008, pp. 283–292.
[124] S. Tang and S.-P. Chen, "The generation mechanism of synthetic minority class examples," in *Proc. Int. Conf. Technol. Appl. Biomed.*, May 2008, pp. 444–447.
[125] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2009, pp. 475–482.
[126] S. Hu, Y. Liang, L. Ma, and Y. He, "MSMOTE: Improving classification performance when training data is imbalanced," in *Proc. 2nd Int. Workshop Comput. Sci. Eng.*, vol. 2, Oct. 2009, pp. 13–17.
[127] L. Chen, Z. Cai, L. Chen, and Q. Gu, "A novel differential evolution-clustering hybrid resampling algorithm on imbalanced datasets," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining*, Jan. 2010, pp. 81–85.
[128] S. Wang, Z. Li, W. Chao, and Q. Cao, "Applying adaptive over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–8.
[129] S. Cateni, V. Colla, and M. Vannucci, "Novel resampling method for the classification of imbalanced datasets for industrial and other real-world problems," in *Proc. 11th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2011, pp. 402–407.
[130] X. Fan, K. Tang, and T. Weise, "Margin-based over-sampling method for learning from imbalanced datasets," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2011, pp. 309–320.
[131] M. A. H. Farquad and I. Bose, "Preprocessing unbalanced data using support vector machine," *Decis. Support Syst.*, vol. 53, no. 1, pp. 226–233, Apr. 2012.
[132] K. Puntumapon and K. Waiyamai, "A pruning-based approach for searching precise and generalized region for synthetic minority over-sampling," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2012, pp. 371–382.
[133] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RS B*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowl. Inf. Syst.*, vol. 33, no. 2, pp. 245–265, 2012.
[134] S. Barua, M. M. Islam, and K. Murase, "ProWSyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2013, pp. 317–328.
[135] C. Bunkhumpornpat and S. Subpaiboonkit, "Safe level graph for synthetic minority over-sampling techniques," in *Proc. 13th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Sep. 2013, pp. 570–575.
[136] M. Nakamura, Y. Kajiwara, A. Otsuka, and H. Kimura, "LVQ-SMOTE–learning vector quantization based synthetic minority over–sampling technique for biomedical data," *BioData Mining*, vol. 6, no. 1, pp. 1–10, 2013.
[137] A. I. Sánchez, E. F. Morales, and J. A. Gonzalez, "Synthetic oversampling of instances using clustering," *Int. J. Artif. Intell. Tools*, vol. 22, no. 2, Apr. 2013, Art. no. 1350008.
[138] G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 92–122, Jan. 2014.
[139] F. Koto, "SMOTE-out, SMOTE-cosine, and selected-SMOTE: An enhancement strategy to handle imbalance in data level," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst.*, Oct. 2014, pp. 280–284.
[140] T. Maciejewski and J. Stefanowski, "Local neighbourhood extension of SMOTE for mining imbalanced data," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Apr. 2011, pp. 104–111.
[141] M. Gao, X. Hong, S. Chen, C. J. Harris, and E. Khalaf, "PDFOS: PDF estimation based over-sampling for imbalanced two-class problems," *Neurocomputing*, vol. 138, pp. 248–259, Aug. 2014.
[142] H. Zhang and M. Li, "RWO-sampling: A random walk over-sampling approach to imbalanced data classification," *Inf. Fusion*, vol. 20, pp. 99–116, Nov. 2014.

[143] B. A. Almogahed and I. A. Kakadiaris, "NEATER: Filtering of over-sampled data using non-cooperative game theory," *Soft Comput.*, vol. 19, no. 11, pp. 3301–3322, Nov. 2015.

[144] C. Bellinger, N. Japkowicz, and C. Drummond, "Synthetic oversampling for advanced radioactive threat detection," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 948–953.

[145] S. Gazzah, A. Hechkel, and N. Essoukri Ben Amara, "A hybrid sampling method for imbalanced data," in *Proc. IEEE 12th Int. Multi-Conference Syst., Signals Devices (SSD15)*, Mar. 2015, pp. 1–6.

[146] L. Jiang, C. Qiu, and C. Li, "A novel minority cloning technique for cost-sensitive learning," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 29, no. 4, Jun. 2015, Art. no. 1551004.

[147] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Inf. Sci.*, vol. 291, pp. 184–203, Jan. 2015.

[148] B. Tang and H. He, "KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 664–671.

[149] Z. Xie, L. Jiang, T. Ye, and X. Li, "A synthetic minority oversampling method based on local densities in low-dimensional space for imbalanced learning," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2015, pp. 3–18.

[150] W. A. Young, S. L. Nykl, G. R. Weckman, and D. M. Chelberg, "Using Voronoi diagrams to improve classification performances when modeling imbalanced datasets," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1041–1054, Jul. 2015.

[151] W. A. Rivera and P. Xanthopoulos, "A priori synthetic over-sampling methods for increasing classification sensitivity in imbalanced data sets," *Expert Syst. Appl.*, vol. 66, pp. 124–135, Dec. 2016.

[152] F. R. Torres, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "SMOTE-D a deterministic version of SMOTE," in *Proc. Mex. Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2016, pp. 177–188.

[153] J. Cervantes, F. Garcia-Lamont, L. Rodriguez, A. López, J. R. Castilla, and A. Trueba, "PSO-based method for SVM classification on skewed data sets," *Neurocomputing*, vol. 228, pp. 187–197, Mar. 2017.

[154] L. Ma and S. Fan, "CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests," *BMC Bioinf.*, vol. 18, no. 1, pp. 1–18, Dec. 2017.

[155] S. Chen, G. Guo, and L. Chen, "A new over-sampling method based on cluster ensembles," in *Proc. IEEE 24th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Apr. 2010, pp. 599–604.

[156] Y.-I. Kang and S. Won, "Weight decision algorithm for oversampling technique on class-imbalanced learning," in *Proc. ICCAS*, Oct. 2010, pp. 182–186.

[157] S. Barua, M. M. Islam, and K. Murase, "A novel synthetic minority oversampling technique for imbalanced data set learning," in *Proc. Int. Conf. Neural Inf. Process.* Berlin, Germany: Springer, 2011, pp. 735–744.

[158] B. Zhou, C. Yang, H. Guo, and J. Hu, "A quasi-linear SVM combined with assembled SMOTE for imbalanced data classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–7.

[159] K. Li, W. Zhang, Q. Lu, and X. Fang, "An improved SMOTE imbalanced data classification method based on support degree," in *Proc. Int. Conf. Identificat., Inf. Knowl. Internet Things*, Oct. 2014, pp. 34–38.

[160] S. Mahmoudi, P. Moradi, F. Akhlaghian, and R. Moradi, "Diversity and separable metrics in over-sampling technique for imbalanced data classification," in *Proc. 4th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Oct. 2014, pp. 152–158.

[161] T. Sandhan and J. Y. Choi, "Handling imbalanced datasets by partially guided hybrid sampling for pattern recognition," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 1449–1453.

[162] Y. H. Xu, H. Li, L. P. Le, and X. Y. Tian, "Neighborhood triangular synthetic minority over-sampling technique for imbalanced prediction on small samples of Chinese tourism and hospitality firms," in *Proc. 7th Int. Joint Conf. Comput. Sci. Optim.*, Jul. 2014, pp. 534–538.

[163] J. Lee, N.-R. Kim, and J.-H. Lee, "An over-sampling technique with rejection for imbalanced class learning," in *Proc. 9th Int. Conf. Ubiquitous Inf. Manage. Commun.*, Jan. 2015, pp. 1–6.

[164] J. Li, S. Fong, and Y. Zhuang, "Optimizing SMOTE by metaheuristics with neural network and decision tree," in *Proc. 3rd Int. Symp. Comput. Bus. Intell. (ISCBI)*, Dec. 2015, pp. 26–32.

[165] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[166] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling and boosting techniques," *Soft Comput.*, vol. 19, no. 12, pp. 3369–3385, Dec. 2015.

[167] Y. Dong and X. Wang, "A new over-sampling approach: Random-SMOTE for learning from imbalanced data sets," in *Proc. Int. Conf. Knowl. Sci., Eng. Manage.* Berlin, Germany: Springer, 2011, pp. 343–352.

[168] K. Borowska and J. Stepaniuk, "Imbalanced data classification: A novel re-sampling approach combining versatile improved SMOTE and rough sets," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.* Cham, Switzerland: Springer, 2016, pp. 31–42.

[169] K. Jiang, J. Lu, and K. Xia, "A novel algorithm for imbalance data classification based on genetic algorithm improved SMOTE," *Arabian J. Sci. Eng.*, vol. 41, no. 8, pp. 3255–3266, Aug. 2016.

[170] E. Ramentol, I. Gondres, S. Lajes, R. Bello, Y. Caballero, C. Cornelis, and F. Herrera, "Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: The SMOTE-FRST-2T algorithm," *Eng. Appl. Artif. Intell.*, vol. 48, pp. 134–139, Feb. 2016.

[171] J. Yun, J. Ha, and J.-S. Lee, "Automatic determination of neighborhood size in SMOTE," in *Proc. 10th Int. Conf. Ubiquitous Inf. Manage. Commun.*, Jan. 2016, pp. 1–8.

[172] W. A. Rivera, "Noise reduction *a priori* synthetic over-sampling for class imbalanced data sets," *Inf. Sci.*, vol. 408, pp. 146–161, Oct. 2017.

[173] J. Li, S. Fong, R. K. Wong, and V. W. Chu, "Adaptive multi-objective swarm fusion for imbalanced data classification," *Inf. Fusion*, vol. 39, pp. 1–24, Jan. 2018.

[174] T. Rong, H. Gong, and W. W. Ng, "Stochastic sensitivity oversampling technique for imbalanced data," in *Proc. Int. Conf. Mach. Learn. Cybern.* Berlin, Germany: Springer, 2014, pp. 161–171.

[175] L. Zhang and W. Wang, "A re-sampling method for class imbalance learning with credit data," in *Proc. Int. Conf. Inf. Technol., Comput. Eng. Manage. Sci.*, vol. 1, Sep. 2011, pp. 393–397.

[176] F. Fernández-Navarro, C. Hervás-Martínez, and P. A. Gutiérrez, "A dynamic over-sampling procedure based on sensitivity for multi-class problems," *Pattern Recognit.*, vol. 44, no. 8, pp. 1821–1833, 2011.

[177] J. Hu, X. He, D.-J. Yu, X.-B. Yang, J.-Y. Yang, and H.-B. Shen, "A new supervised over-sampling algorithm with application to protein-nucleotide binding residue prediction," *PLoS ONE*, vol. 9, no. 9, Sep. 2014, Art. no. e107676.

[178] V. García, J. S. Sánchez, R. Martìn-Félez, and R. A. Mollineda, "Surrounding neighborhood-based SMOTE for learning from imbalanced data sets," *Prog. Artif. Intell.*, vol. 1, no. 4, pp. 347–362, 2012.

[179] M. Koziarski and M. Woźniak, "CCR: A combined cleaning and resampling algorithm for imbalanced data classification," *Int. J. Appl. Math. Comput. Sci.*, vol. 27, no. 4, pp. 727–736, Dec. 2017.

[180] W. Siriseriwan and K. Sinapiromsaran, "Adaptive neighbor synthetic minority oversampling technique under 1NN outcast handling," *Songklanakarin J. Sci. Technol.*, vol. 39, no. 5, pp. 565–576, Sep. 2017.

[181] D. A. Cieslak, N. V. Chawla, and A. Striegel, "Combating imbalance in network intrusion datasets," in *Proc. GrC*, 2006, pp. 732–737.

[182] A. B. Hassanat, "Dimensionality invariant similarity measure," *Comput. Sci.*, vol. 10, no. 8, pp. 221–226, Aug. 2014.

[183] H. A. Abu Alfeilat, A. B. Hassanat, O. Lasassmeh, A. S. Tarawneh, M. B. Alhasanat, H. S. Eyal Salman, and V. S. Prasath, "Effects of distance measure choice on K-nearest neighbor classifier performance: A review," *Big Data*, vol. 7, no. 4, pp. 221–248, Dec. 2019.

[184] R. Ehsani and F. Drabløs, "Robust distance measures for kNN classification of cancer data," *Cancer Inform.*, vol. 19, Oct. 2020, Art. no. 1176935120965542.

[185] C. R. Kancharla, J. Vankeirsbilck, D. Vanoost, J. Boydens, and H. Hallez, "Latent dimensions of auto-encoder as robust features for inter-conditional bearing fault diagnosis," *Appl. Sci.*, vol. 12, no. 3, p. 965, Jan. 2022.

[186] R. Veerachamy and R. Ramar, "Agricultural irrigation recommendation and alert (AIRA) system using optimization and machine learning in Hadoop for sustainable agriculture," *Environ. Sci. Pollut. Res.*, vol. 29, pp. 19955–19974, Mar. 2021.

[187] M. Farooq, S. Sarfraz, C. Chesneau, M. Ul Hassan, M. A. Raza, R. A. K. Sherwani, and F. Jamal, "Computing expectiles using K-nearest neighbours approach," *Symmetry*, vol. 13, no. 4, p. 645, Apr. 2021.

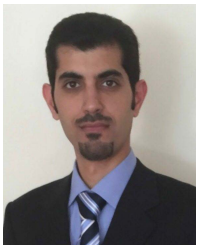
[188] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[189] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 39, no. 2, pp. 539–550, Apr. 2008.

AHMAD S. TARAWNEH was born in Karak, Jordan. He received the B.S. and M.S. degrees in computer science from Mutah University, in 2013 and 2015, respectively, and the Ph.D. degree in computer science from Eötvös Loránd University (ELTE), Hungary, Budapest. He worked on the project of EFOP (image and video processing), which is sponsored by Hungarian government and co-financed by the European Social Fund. His main research interests include computer vision, deep learning, machine learning, and data mining.

AHMAD B. HASSANAT (Member, IEEE) was born in Jordan. He received the B.S. degree in computer science from Mutah University, Jordan, in 1995, the M.S. degree in computer science from Al al-Bayt University, Jordan, in 2004, and the Ph.D. degree in computer science from The University of Buckingham, U.K., in 2010. He has been a Faculty Member of the Faculty of Information Technology, Mutah University, since 2010. His main research interests include computer vision, machine learning, big data, and pattern recognition.

GHADA AWAD ALTARAWNEH received the Ph.D. degree in accounting from The University of Buckingham, U.K., in 2011. She is currently an Associate Professor with Mutah University. Her main interests include managerial accounting, auditing, and business intelligence.



ABDULLAH ALMUHAIMEED received the bachelor's degree in computer science from Imam Muhammad Ibn Saud Islamic University, in 2007, and the M.Sc. and Ph.D. degrees in computer science from the University of Essex, U.K., in 2011 and 2016, respectively. He is currently an Assistant Research Professor of computer science with the King Abdulaziz City for Science and Technology (KACST).