electronics

Article

RDPVR: Random Data Partitioning with Voting Rule for Machine Learning from Class-Imbalanced Datasets

Ahmad B. Hassanat [1], Ahmad S. Tarawneh [2,*], Samer Subhi Abed [1], Ghada Awad Altarawneh [3], Malek Alrashidi [4] and Mansoor Alghamdi [4]

[1] Faculty of Information Technology, Mutah University, Mutah, Karak 6171, Jordan; hasanat@Mutah.edu.jo (A.B.H.); alhashmi.samer@gmail.com (S.S.A.)
[2] Faculty of informatics, Eotvos Lorand University, 1117 Budapest, Hungary
[3] Department of Accounting, Mutah University, Mutah, Karak 6171, Jordan; Ghadaa@Mutah.edu.jo
[4] Department of Computer Science, Applied College, University of Tabuk, Tabuk 71491, Saudi Arabia; Mqalrashidi@ut.edu.sa (M.A.); malghamdi@ut.edu.sa (M.A.)
* Correspondence: Ahmad.trwh@gmail.com

Abstract: Since most classifiers are biased toward the dominant class, class imbalance is a challenging problem in machine learning. The most popular approaches to solving this problem include oversampling minority examples and undersampling majority examples. Oversampling may increase the probability of overfitting, whereas undersampling eliminates examples that may be crucial to the learning process. We present a linear time resampling method based on random data partitioning and a majority voting rule to address both concerns, where an imbalanced dataset is partitioned into a number of small subdatasets, each of which must be class balanced. After that, a specific classifier is trained for each subdataset, and the final classification result is established by applying the majority voting rule to the results of all of the trained models. We compared the performance of the proposed method to some of the most well-known oversampling and undersampling methods, employing a range of classifiers, on 33 benchmark machine learning class-imbalanced datasets. The classification results produced by the classifiers employed on the generated data by the proposed method were comparable to most of the resampling methods tested, with the exception of SMOTEFUNA, which is an oversampling method that increases the probability of overfitting. The proposed method produced results that were comparable to the Easy Ensemble (EE) undersampling method. As a result, for solving the challenge of machine learning from class-imbalanced datasets, we advocate using either EE or our method.

Keywords: classification; data mining; KNN; CART; SVM; SMOTE

Citation: Hassanat, A.B.; Tarawneh, A.S.; Abed, S.S.; Altarawneh, G.A.; Alrashidi, M.; Alghamdi, M. RDPVR: Random Data Partitioning with Voting Rule for Machine Learning from Class-Imbalanced Datasets. *Electronics* **2022**, *11*, 228. https://doi.org/10.3390/electronics11020228

Academic Editor: Stefano Ferilli

Received: 14 December 2021
Accepted: 7 January 2022
Published: 12 January 2022

1. Introduction

A class imbalance problem occurs when training a dataset that contains examples belonging to one class that significantly outnumber those belonging to the other class(es). The first class is normally referred to as the majority class, while the latter is referred to as the minority. In a single dataset, there may be more than one majority class and more than one minority class. The core problem with class imbalance is that classifiers trained on unequal training sets have a prediction bias that is associated with poor performance in the minority class(es). The bias might range from a minor imbalance to a major imbalance depending on the dataset used [1–4].

Since the minority class is frequently of crucial importance, as representing positive instances that are rare in nature or costly to acquire, this problem has grown and has become a substantial challenge [5]. This is true when considering contexts such as Biometrics [6–14], disease detection [15–19], credit card fraud detection [20,21], gene profiling [22], face image retrieval [23], content-based image retrieval [24,25], Internet of Things [23–33], Natural

Language Processing [34,35], Anomaly Detection [36–46], network security [47–53], image recognition [54–59], Big Data analytics [60–66], etc.

In formal terms, a supervised machine learning dataset D, containing n examples that belong to m classes C1, C2, C3, . . . , Cm, is said to be an imbalanced dataset if and only if any |Ci| >> |Cj|, where i and j are indexes ∈ {1, 2, 3, . . . , m}; and |Ci| is the cardinality of class i, i.e., the number of examples belonging to class i.

Before classification, there are several approaches that might be used to solve an imbalanced dataset problem, such as the following:

- More samples from the minority class(es) should be acquired from the discourse domain.
- Changing the loss function to give the failing minority class a higher cost [67].
- Oversampling of minority class examples.
- Undersampling of majority class examples.
- Any combination of the preceding methods.

In fact, the ability to collect more data is constrained by time and expense. Furthermore, in some fields, the minority class is extremely rare, making it impossible to acquire enough samples at any cost. E.g., in the case of recognizing an abnormal case vs. a normal case, normal cases are easy to come by, whereas abnormal cases are difficult to obtain the same quantity.

Cost adjustment, on the other hand, has been proven to be a successful method of dealing with imbalance; however, it usually requires changes to the classification algorithm and/or prior knowledge of acceptable misclassification costs.

As a result, we have two options for re-sampling: oversampling or undersampling. The number of examples belonging to the minority class increases when oversampling is used, while undersampling methods reduce the number of examples from the majority class.

However, both approaches, in our opinion, have their own set of problems. For example, oversampling creates new examples out of nothingness based only on their similarity to one or more of the minority's examples. This is problematic because such methods may increase the probability of overfitting the learning process [68,69].

Undersampling, on the other hand, eliminates examples that may be critical to the learning process, making it even worse. This likewise produces positive results on paper, but the opposite is true in practice.

In this paper, we present a Random Data Partitioning with Voting Rule (RDPVR) method for Machine Learning from Class-Imbalanced Datasets to avoid the drawbacks of both oversampling and undersampling. Here, we partition an imbalanced dataset into smaller balanced subdatasets to establish a fair learning process for imbalanced datasets. This is done by randomly selecting several majority's examples equal to the number of the minority's examples, discarding all the extra majority's examples, and keeping all of minority's examples for each subdataset. Thus, we ensure that each subdataset was balanced. Then, we utilize each of these subdatasets to train a classifier independently, resulting in several trained models that are all used to classify using a simple voting rule. Naïve Bayes (NB), K-Nearest Neighbors (KNN), Classification and Regression Tree (CART), and Support Vector Machine (SVM) are among the classifiers utilized to evaluate our RDPVR method.

The following is the structure of this paper: The related work on imbalanced datasets in machine learning is presented in the second section. The RDPVR method is illustrated in Section 3, and the experimental results are listed and discussed in Section 4.

2. Related Work

In the literature, there is a plethora of ways for machine learning from imbalanced data. However, we focus on oversampling and undersampling approaches in this study because they are the most related approaches to our RDPVR method. For a more comprehensive and extensive assessment, we refer the readers to [70–72].

2.1. Oversampling

One of the most extensively utilized approaches for mitigating the detrimental consequences of class imbalance is the Synthetic Minority Oversampling Technique (SMOTE) [73]. It interpolates synthetic examples in the training set's collection of minority class instances between nearest neighbors. As a result, a synthetic sample is created by combining the characteristics of seed instances with randomly selected k-nearest neighbors. The user must specify the k parameter. The SMOTE algorithm's first version exclusively used synthetic oversampling. They also combined synthetic oversampling and undersampling, which can be effective [74]. SMOTE was empirically examined on nine benchmark datasets, and it was found to boost the classification process.

Borderline-SMOTE [75] is a minority oversampling method based on SMOTE, in which only the minority examples close to the borderline are oversampled. Their experiments demonstrate that this solution improves classification results for the minority class compared to SMOTE and other random oversampling methods tested.

SVMSMOTE [76] is also based on SMOTE and focuses on building SVM modifications to address the problem of class imbalance effectively. SVM modeling uses a variety of heuristics, such as oversampling, cost-sensitive learning, and undersampling. When compared to other oversampling methods, this method yielded promising results.

Reverse-SMOTE (R-SMOTE) [77], an approach based on SMOTE and the inverse near-neighbor notion, relies on oversampling by a synthetic inverse minority. After comparing conventional sampling approaches to some methods including SMOTE, it was found that R-SMOTE outperforms other oversampling methods in terms of precision, F-measurement, and accuracy. Three benchmark datasets were used in the comparison.

Constrained Oversampling (CO) [78] is developed in order to limit noise creation in oversampling. The overlapping regions in the dataset are initially extracted by this method. Then, to define the limits of minority regions, Ant Colony Optimization is employed. Most importantly, in order to provide a balanced dataset, oversampling under limitations is used to synthesis fresh samples. This method differs from others in that it includes limitations in the oversampling process to reduce noise creation. The reported results indicate that CO outperforms a variety of oversampling benchmarks.

In addition, the Majority Weighted Minority Oversampling Technique (MWMOTE) [79] was proposed as a solution to the problem of class-imbalance learning. MWMOTE finds and weights difficult-to-learn significant minority class samples based on their distance from neighboring majority class samples. Then, it creates synthetic samples from the weighted significant 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 reported results suggest that MWMOTE is better than or similar to some of the other existing approaches.

In order to minimize bias, and pushing the classification decision border in the direction of the hard examples, adaptive synthetic (ADASYN) [80] was presented. The primary idea behind ADASYN is to use weighted values for different minority class examples based on how difficult they are to learn, with more synthetic data generated for minority class examples that are more difficult to learn than minority class examples that are easier to learn. The viability of this technique is proved by the results of experiments performed on a variety of datasets using five different evaluation methods.

Synthetic Minority Oversampling Technique Based on Furthest Neighbor Algorithm (SOMTEFUNA) [5] 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. Utilizing Support Vector Machine and Naïve Bayes classifiers, SOMTEFUNA 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.

Sampling WIth the Majority (SWIM) [81] is a synthetic oversampling method that is robust in extreme class imbalance scenarios. SWIM's main characteristic is that it guides the generation process using the density of the well-sampled majority class. Both the radial basis function and Mahalanobis distance were employed to build SWIM's model. SWIM was tested on 25 benchmark datasets, and the reported results reveal that SWIM outperforms some of the common oversampling approaches.

Other ways of oversampling include, but are not limited to, the work of [82–111].

2.2. Undersampling

The undersampling classification algorithm based on mean shift clustering for imbalanced data (UECMS) [112] was presented to accomplish the undersampling by using mean shift clustering and instance selection for the samples of majority classes. A fresh balanced dataset is created by combining the selected samples with all of the minority samples from the original dataset. The balanced datasets are also classified using bagging-based ensemble learning methods. The UECMS approach enhances the classification accuracy for imbalanced data, according to the findings of the study.

Another undersampling method based on meta-learning was presented to solve class-imbalance learning [4]. This method's fundamental idea is to parameterize and train the data sampler in order to enhance classification performance over the evaluation measure. For training the data sampler, they applied reinforcement learning to solve the non-differentiable optimization problem. By including evaluation metric optimization into the data sampling process, their technique may learn which instances should be discarded for a particular classifier and evaluation measure. As a data-level approach, this method can be easily applied to any evaluation measure and classifier. Experiments on both synthetic and realistic datasets confirmed the efficacy of their method.

DBMIST-US [113] is a two-stage undersampling method that combines the DBSCAN [114] clustering algorithm with a minimal spanning tree algorithm to reduce noisy samples and clear the decision boundary, allowing classifiers to handle imbalance and class overlap at the same time. The results show that DBMIST-US outperforms a total of 12 undersampling methods.

The Easy Ensemble and Balance Cascade (EE&BC) [115] is maybe one of the most interesting undersampling approaches we found in the literature. This approach consists of two methods: Easy Ensemble (EE) and Balance Cascade (BC). EE takes numerous subgroups of the majority class, trains a learner with each of them, and then combines their outputs. BC trains the learners in stages, with the majority of class examples properly classified by the present trained learners being eliminated from consideration at each stage. Both EE and BC outperform other methods in terms of accuracy and time consumed, according to the results of balancing a total of 16 datasets.

Other ways of undersampling include, but are not limited to, the work of [116–142].

Oversampling and undersampling both have advantages and disadvantages, and there is no one-size-fits-all method for fair machine learning from a class-imbalanced dataset, as previously noted. EE&BC, on the other hand, was the most akin to our views and thus to our proposed method, because both EE&BC and the proposed RDPVR aim to benefit from every single example, regardless of whether it belongs to the majority or minority group.

Since the proposed RDPVR removes the excessive majority examples, it can be regarded as an undersampling method; nevertheless, the eliminated ones in one subdataset may exist in another, avoiding the aforementioned major problem of undersampling.

3. The Proposed Method

When it comes to data classification, class imbalance occurs when one majority class's training samples vastly outnumber those of the other minority class. The fundamental issue with class imbalance is that classifiers trained on an imbalanced dataset have a prediction bias, resulting in poor performance in the minority class.

We opt for randomly partitioning the dataset into smaller balanced subdatasets to mitigate the effect of class imbalance in a machine learning dataset and build a fair learning process while avoiding the downsides of both oversampling and undersampling. This is accomplished by selecting a number of majority examples equal to the number of minority examples at random, removing any excess majority examples, and maintaining all minority examples for each subdataset. As a result, we make certain that each subdataset is balanced. Then, using each of these subdatasets separately, we train a classifier, yielding a number of trained models that are all utilized to classify using a simple voting rule, as shown in Figures 1 and 2. One important factor that needs to be considered here is the number of subsets (partitions). The number of subsets needed by the algorithm is a hyper parameter that needs to be tuned for each dataset.

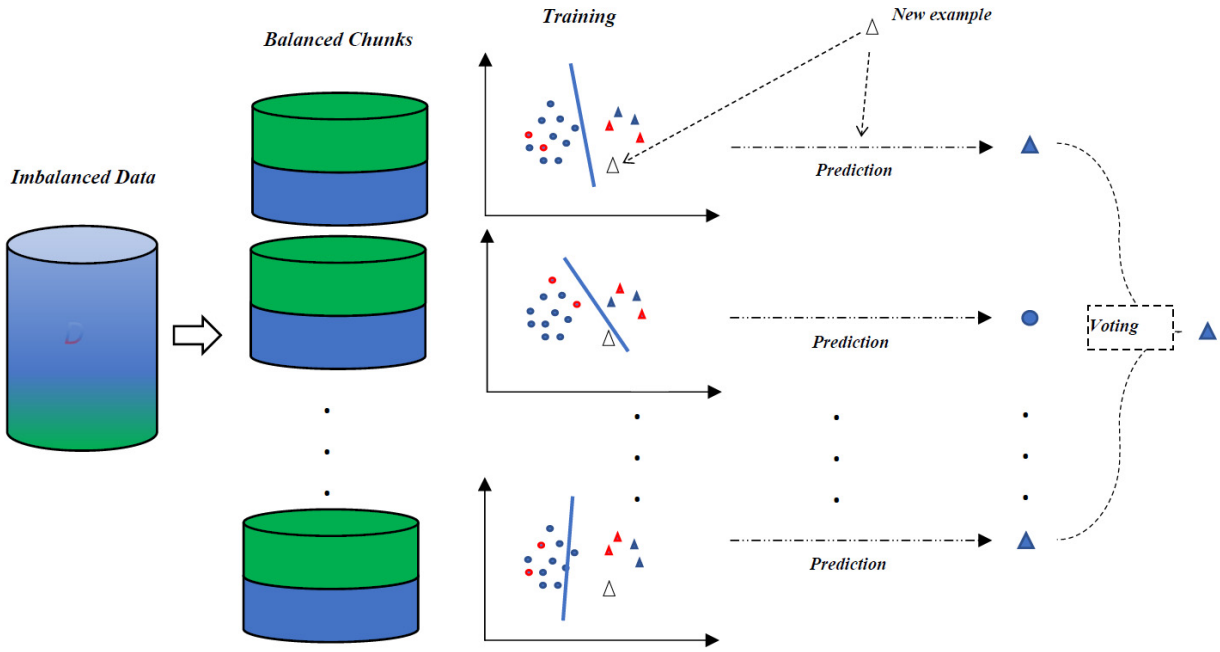

Figure 1. Illustration of the proposed RDPVR method, Green color represents minority examples, blue represents majority examples.

We employed the following classifiers to evaluate the proposed RDPVR method: KNN, SVM, NB, and CART, because they are commonly used in this type of work. It is worth noting that we utilize the same classifier across all data partitions, as depicted in Figure 2.

One important point to mention here is that the EE method works in a similar way to our proposed method. However, the main difference is that the proposed method uses strong classifiers instead of employing weak learners.

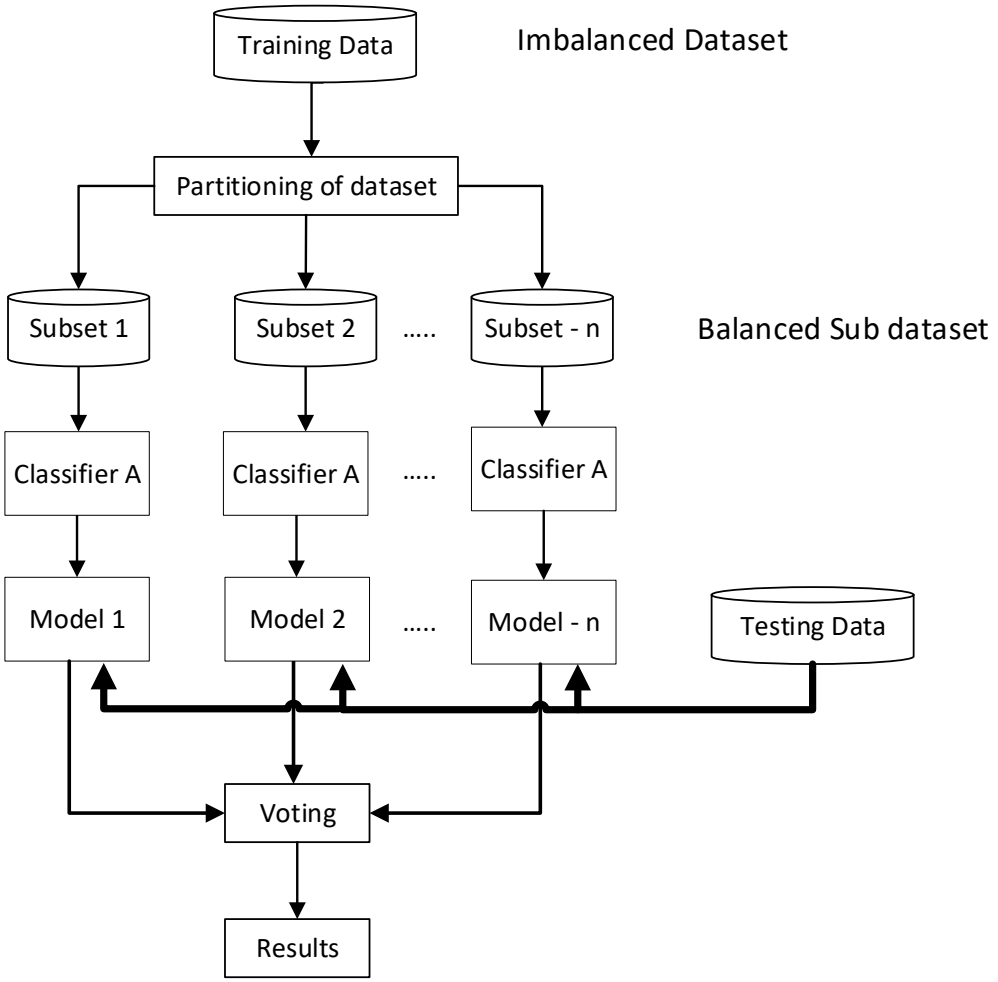

Figure 2. The proposed RDPVR method's flowchart.

4. Data

In order to compare the proposed RDPVR method to some of the other methods, we selected 33 small, medium, and large benchmark datasets, all of which had only two classes [5]. The datasets were retrieved from the Kaggle website (https://www.kaggle.com/ accessed on 1 December 2021). The datasets are described in Table 1 in terms of the number of examples, features, classes, and imbalance ratios.

To compare the proposed RDPVR method to the other methods in this study, we used the f-score measure. The F-score is a widely used classification measure for evaluating classifiers, particularly those trained on poorly balanced datasets, mostly because it harmonically combines recall and precision. Therefore, it helps to comprehend the performance of a classifier after the resampling process. The F-score can be calculated using the following formula.

$$F - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (1)$$

Table 1. Description of the datasets used to compare the methods in this work.

ID	Name	No. of Instances	No. of Features	No. of Classes	Imbalance Ratio
DS1	Ar1	121	30	2	13.4
DS2	Ar3	63	30	2	7.875
DS3	Ar4	107	30	2	5.45
DS4	Ar5	36	30	2	4.5
DS5	Ar6	101	30	2	6.7
DS6	Kc1	2109	22	2	60.5
DS7	Kc2	522	22	2	4.9
DS8	Pc1	1109	22	2	14.4
DS9	Pc3	1563	38	2	9.8
DS10	Pc4	1458	38	2	8.2
DS11	Australian	690	42	2	1.25
DS12	Bank	1372	4	2	1.25
DS13	Heart	270	25	2	1.25
DS14	Oil-Spill	937	49	2	21.85
DS15	Phoneme	5404	5	2	2.41
DS16	Apalone19	4174	8	2	129.44
DS17	Apalone9-18	731	8	2	16.4
DS18	Page-blocks0	5472	11	2	8.79
DS19	Pima	768	9	2	1.87
DS20	Segment0	2308	20	2	6.2
DS21	Shuttle-c0	1829	10	2	13.87
DS22	Vehicle0	846	19	2	3.25
DS23	Vehicle1	846	19	2	2.9
DS24	Vehicle2	846	19	2	2.88
DS25	Vehicle3	846	19	2	2.99
DS26	Vowe10	988	14	2	9.98
DS27	Wisconsin	683	10	2	1.86
DS28	Yeast-1-2-8-9	947	9	2	30.57
DS29	Yeast-1-4-5-8	693	9	2	22.1
DS30	Yeast1	1484	9	2	2.46
DS31	Yeast3	1484	9	2	8.1
DS32	Yeast4	1484	9	2	28.1
DS33	Yeast5	1484	9	2	32.73

5. Experiments and Results

We used the Python programming language on Google Collaboratory, a Google Research gift, to program and evaluate the proposed RDPVR and the other methods compared. It is a gift to researchers all over the world because this free technology allows groups to collaborate, write, and run any Python code, and it is especially well-suited to machine learning while also providing free access to computer resources such as GPUs [143]. Table 2 shows the specifications of the Google Collaboratory hardware utilized.

Table 2. Hardware specifications of Google Collaboratory.

Property	Value
CPU Model Name	Intel (R) Xeon (R)
CPU Freq.	2.30 GHz
No. CPU Cores	2
CPU Family	Haswell
Available RAM	12 GB (upgradable to 26.75 GB)
Disk Space	25 GB

We arbitrarily used the CART classifier on different numbers of subdatasets in the range of 10 to 100 and 100 to 1000 to find the best number of subdatasets to use for the comparisons, as shown in Table 3. Our method has a parameter that needs to be tuned,

which is the number of subdatasets generated from the original dataset. We do not report the results of using a small number of subdatasets, such as 1 or 2, because we believe that doing so is illogical because a large number of the majority examples will be excluded, potentially affecting the learning process.

Table 3. The CART classifier average F-score result of the proposed RDPVR over all datasets for each number of subdatasets.

#Subsets	Avg. F-Score	#Subsets	Avg. F-Score	Improvement
10	0.778	100	0.798	2.0%
20	0.785	200	0.806	2.1%
30	0.780	300	0.803	2.3%
40	0.780	400	0.802	2.2%
50	0.781	500	0.799	1.8%
60	0.783	600	0.808	2.5%
70	0.786	700	0.809	2.3%
80	0.782	800	0.815	3.3%
90	0.788	900	0.810	2.1%
100	0.787	1000	0.804	1.7%

Table 3 shows that with a few exceptions, the RDPVR's performance improves as the number of the subdatasets increases. These exceptions are necessary for two reasons: (1) The majority examples are chosen at random, and (2) Different types of datasets prefer a different number of subdatasets. In general, the larger the number of subdatasets, the more examples from the majority class that can participate; however, the larger the number of subdatasets, the more training time is required. Here, we utilized 400 subdatasets for comparative purposes because using that many boosts the F-score by 2.2% when compared to using 40 subdatasets for instance, with the caveat that using 800 should be better in terms of accuracy but requires more training time. According to the results in Table 3, employing 300 is better than 400, but we want to make sure that a higher number of the majority examples participate in the learning process.

We compared our method to a number of the most common resampling methods used to solve class-imbalance problem, which is also implemented on the same Google Collaboratory platform. These methods include EE [115], SMOTE [73], ADASYN [80], SOMTEFUNA) [5], SVMSMOTE [76], and Borderline-SMOTE [75]. After applying these methods to the datasets mentioned in Table 1, we used different classifiers to classify each dataset. The comparison accuracy results using SVM are shown in Table 4. In Tables 4–7, The values with gray background are superior compared to EE method.

As shown in Table 4, the best results are obtained when SMOTEFUNA is used for oversampling on almost all datasets. SMOTEFUNA, on the other hand, has a significant disadvantage: it is time consuming. SMOTEFUNA searches for the farthest neighbor using quadratic time. Furthermore, it takes another quadratic time for each created point to determine the class of the closest point to the created one. As a result, it may not be suitable for large datasets.

Theoretically, the proposed RDPVR has a linear O(n) time complexity because it works on the training data set a constant (C) times to generate C random undersampled subdatasets. However, the actual time complexity is O(Cn); this means that the higher the number of the subdatasets, the more time it consumes, even though it is significantly faster than SMOTEFUNA even when we used 400 subdatasets, especially when dealing with large datasets. Furthermore, in certain cases where the number of examples in the minority class is much smaller than n (the size of the dataset), the time complexity becomes constant, i.e., O(C), which is asymptotically equivalent to O(1), because we will have a constant number of very small subdatasets.

Table 4. SVM classification results using F-score, with shaded cells with gray indicate the best performance of the proposed method compared to EE.

Dataset ID	Proposed		EE		SMOTE		ADASYN		SVMSMOTE		Borderline-SMOTE		SMOTEFUNA	
	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD
DS1	0.51	0.08	0.52	0.09	0.58	0.10	0.53	0.08	0.53	0.07	0.56	0.09	0.95	0.03
DS2	0.63	0.17	0.55	0.10	0.66	0.13	0.63	0.10	0.69	0.13	0.72	0.11	0.95	0.04
DS3	0.69	0.05	0.59	0.07	0.56	0.08	0.58	0.05	0.65	0.06	0.59	0.07	0.90	0.04
DS4	0.79	0.13	0.83	0.14	0.81	0.14	0.80	0.15	0.79	0.12	0.78	0.15	0.90	0.06
DS5	0.55	0.10	0.52	0.10	0.59	0.12	0.58	0.12	0.57	0.11	0.57	0.10	0.93	0.02
DS6	0.75	0.02	0.99	0.00	0.92	0.02	0.92	0.02	0.91	0.02	0.92	0.02	0.99	0.00
DS7	0.85	0.04	0.98	0.02	0.91	0.03	0.93	0.03	0.90	0.03	0.94	0.01	0.98	0.01
DS8	0.70	0.05	0.97	0.02	0.82	0.03	0.82	0.03	0.83	0.03	0.83	0.03	0.97	0.01
DS9	0.63	0.02	0.58	0.03	0.63	0.02	0.62	0.02	0.66	0.02	0.65	0.03	0.94	0.01
DS10	0.65	0.02	0.74	0.02	0.71	0.01	0.70	0.02	0.72	0.02	0.70	0.02	0.91	0.03
DS11	0.84	0.02	0.84	0.02	0.86	0.02	0.86	0.02	0.86	0.03	0.85	0.02	0.98	0.01
DS12	0.95	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.87	0.01
DS13	0.79	0.02	0.79	0.04	0.79	0.04	0.79	0.04	0.79	0.03	0.78	0.04	0.98	0.01
DS14	0.74	0.04	0.64	0.03	0.71	0.04	0.70	0.03	0.75	0.07	0.74	0.06	0.79	0.02
DS15	0.75	0.01	0.78	0.01	0.79	0.01	0.77	0.01	0.78	0.01	0.77	0.01	0.98	0.00
DS16	0.47	0.01	0.43	0.01	0.50	0.01	0.50	0.01	0.52	0.02	0.52	0.03	0.97	0.01
DS17	0.65	0.04	0.64	0.04	0.68	0.05	0.66	0.04	0.72	0.04	0.71	0.04	0.91	0.01
DS18	0.84	0.01	0.86	0.01	0.87	0.01	0.83	0.01	0.85	0.01	0.85	0.02	1.00	0.00
DS19	0.70	0.03	0.71	0.03	0.71	0.02	0.71	0.02	0.72	0.03	0.71	0.02	0.80	0.02
DS20	0.88	0.03	0.98	0.01	0.99	0.00	0.98	0.01	0.98	0.01	0.98	0.01	0.84	0.02
DS21	0.99	0.01	1.00	0.01	0.99	0.01	0.99	0.00	0.99	0.00	0.99	0.00	0.97	0.01
DS22	0.74	0.05	0.94	0.02	0.95	0.01	0.95	0.02	0.94	0.02	0.94	0.02	0.90	0.01
DS23	0.65	0.03	0.73	0.03	0.76	0.03	0.76	0.03	0.75	0.03	0.76	0.03	1.00	0.00
DS24	0.80	0.02	0.95	0.01	0.97	0.01	0.96	0.01	0.96	0.01	0.96	0.01	0.99	0.00
DS25	0.62	0.03	0.71	0.02	0.74	0.02	0.75	0.02	0.74	0.02	0.74	0.02	0.84	0.02
DS26	0.81	0.03	0.90	0.03	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.86	0.02
DS27	0.96	0.01	0.97	0.01	0.97	0.01	0.97	0.01	0.97	0.01	0.97	0.01	1.00	0.00
DS28	0.67	0.05	0.90	0.05	0.77	0.05	0.77	0.05	0.77	0.07	0.77	0.06	0.84	0.01
DS29	0.72	0.05	0.93	0.05	0.83	0.06	0.82	0.06	0.84	0.05	0.84	0.05	0.94	0.01
DS30	0.65	0.03	0.72	0.02	0.69	0.02	0.67	0.02	0.69	0.02	0.67	0.02	0.96	0.01
DS31	0.88	0.03	0.83	0.01	0.86	0.03	0.83	0.02	0.87	0.03	0.85	0.02	0.98	0.01
DS32	0.65	0.04	0.57	0.03	0.61	0.04	0.61	0.04	0.65	0.06	0.63	0.05	1.00	0.00
DS33	0.74	0.03	0.73	0.02	0.82	0.03	0.82	0.03	0.80	0.04	0.82	0.04	0.98	0.01
Average	**0.74**	**0.04**	**0.78**	**0.03**	**0.79**	**0.04**	**0.78**	**0.03**	**0.79**	**0.04**	**0.79**	**0.04**	**0.94**	**0.04**

Table 5. NB classification results using the F-score, with shaded cells with gray indicate the best performance of the proposed method compared to EE.

Dataset ID	Proposed		EE		SMOTE		ADASYN		SVMSMOTE		Borderline-SMOTE		SMOTEFUNA	
	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD
DS1	0.45	0.05	0.51	0.08	0.41	0.07	0.41	0.07	0.43	0.08	0.42	0.07	0.89	0.04
DS2	0.77	0.13	0.57	0.11	0.77	0.12	0.72	0.14	0.79	0.13	0.78	0.12	0.94	0.07
DS3	0.68	0.09	0.64	0.04	0.67	0.09	0.68	0.06	0.68	0.08	0.66	0.07	0.90	0.04
DS4	0.67	0.15	0.69	0.17	0.66	0.14	0.67	0.16	0.67	0.16	0.67	0.16	0.90	0.06
DS5	0.55	0.09	0.49	0.07	0.61	0.13	0.63	0.12	0.59	0.12	0.61	0.11	0.88	0.03
DS6	0.89	0.04	1.00	0.00	0.75	0.02	0.73	0.02	0.72	0.02	0.74	0.02	0.92	0.01
DS7	0.83	0.03	0.98	0.01	0.73	0.03	0.73	0.05	0.71	0.04	0.73	0.04	0.90	0.02
DS8	0.88	0.06	0.96	0.02	0.70	0.04	0.70	0.04	0.70	0.04	0.70	0.04	0.97	0.01
DS9	0.62	0.03	0.60	0.02	0.37	0.13	0.32	0.11	0.57	0.06	0.46	0.12	0.93	0.01
DS10	0.68	0.02	0.74	0.02	0.68	0.04	0.68	0.03	0.65	0.05	0.68	0.04	0.90	0.01
DS11	0.85	0.01	0.83	0.02	0.74	0.07	0.72	0.07	0.72	0.07	0.73	0.07	0.97	0.01
DS12	0.89	0.03	1.00	0.00	0.85	0.02	0.85	0.02	0.84	0.02	0.84	0.02	0.79	0.01
DS13	0.80	0.04	0.79	0.02	0.81	0.06	0.82	0.07	0.80	0.06	0.80	0.06	0.94	0.01
DS14	0.65	0.03	0.64	0.04	0.22	0.08	0.21	0.08	0.37	0.10	0.31	0.12	0.42	0.01
DS15	0.76	0.01	0.78	0.01	0.69	0.01	0.67	0.01	0.68	0.01	0.67	0.01	0.95	0.00
DS16	0.47	0.03	0.43	0.01	0.36	0.01	0.37	0.01	0.41	0.04	0.40	0.05	0.98	0.01
DS17	0.58	0.03	0.64	0.05	0.49	0.03	0.48	0.03	0.53	0.03	0.50	0.03	0.73	0.01
DS18	0.84	0.01	0.87	0.01	0.73	0.02	0.77	0.01	0.71	0.02	0.76	0.02	0.94	0.01

Table 5. *Cont.*

Dataset ID	Proposed		EE		SMOTE		ADASYN		SVMSMOTE		Borderline-SMOTE		SMOTEFUNA	
	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD
DS19	0.69	0.03	0.74	0.03	0.73	0.03	0.73	0.02	0.73	0.03	0.74	0.03	0.79	0.02
DS20	0.91	0.02	0.98	0.01	0.75	0.03	0.71	0.02	0.71	0.03	0.73	0.04	0.79	0.10
DS21	0.99	0.01	1.00	0.00	0.99	0.01	0.99	0.01	0.99	0.01	0.99	0.01	0.80	0.02
DS22	0.73	0.04	0.92	0.01	0.64	0.03	0.60	0.03	0.62	0.03	0.61	0.03	0.79	0.01
DS23	0.67	0.03	0.71	0.02	0.63	0.03	0.60	0.03	0.62	0.03	0.59	0.03	1.00	0.00
DS24	0.90	0.04	0.95	0.01	0.73	0.04	0.71	0.03	0.73	0.06	0.78	0.05	0.94	0.01
DS25	0.64	0.04	0.70	0.03	0.62	0.03	0.62	0.03	0.61	0.03	0.61	0.04	0.69	0.02
DS26	0.85	0.03	0.91	0.02	0.76	0.03	0.77	0.06	0.77	0.03	0.77	0.04	0.72	0.04
DS27	0.96	0.01	0.96	0.01	0.96	0.01	0.95	0.02	0.95	0.01	0.95	0.02	0.84	0.02
DS28	0.91	0.06	0.90	0.03	0.88	0.06	0.88	0.06	0.89	0.05	0.88	0.05	0.70	0.02
DS29	0.94	0.05	0.94	0.04	0.85	0.06	0.85	0.07	0.88	0.08	0.86	0.07	0.69	0.01
DS30	0.65	0.03	0.71	0.02	0.27	0.01	0.26	0.01	0.26	0.01	0.27	0.01	0.71	0.02
DS31	0.88	0.02	0.82	0.03	0.27	0.09	0.21	0.07	0.23	0.08	0.21	0.07	0.99	0.00
DS32	0.64	0.04	0.58	0.03	0.15	0.05	0.14	0.06	0.23	0.09	0.18	0.08	0.95	0.00
DS33	0.80	0.04	0.71	0.02	0.50	0.03	0.49	0.03	0.42	0.06	0.49	0.03	0.79	0.02
Average	**0.76**	**0.04**	**0.78**	**0.03**	**0.63**	**0.05**	**0.63**	**0.05**	**0.64**	**0.05**	**0.64**	**0.05**	**0.85**	**0.05**

Table 6. CART classification results using the F-score, with shaded cells with gray indicate the best performance of the proposed method compared to EE.

Dataset ID	Proposed		EE		SMOTE		ADASYN		SVMSMOTE		Borderline-SMOTE		SMOTEFUNA	
	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD
DS1	0.58	0.08	0.50	0.08	0.52	0.09	0.56	0.15	0.56	0.10	0.54	0.10	0.93	0.04
DS2	0.74	0.08	0.56	0.11	0.70	0.14	0.66	0.10	0.70	0.15	0.67	0.14	0.89	0.05
DS3	0.68	0.11	0.64	0.06	0.62	0.08	0.61	0.06	0.63	0.07	0.63	0.09	0.87	0.04
DS4	0.68	0.17	0.76	0.20	0.69	0.21	0.63	0.16	0.63	0.25	0.58	0.17	0.88	0.04
DS5	0.54	0.12	0.53	0.10	0.54	0.07	0.53	0.08	0.53	0.06	0.55	0.08	0.87	0.02
DS6	0.99	0.01	1.00	0.00	1.00	0.00	1.00	0.00	0.99	0.01	1.00	0.00	1.00	0.00
DS7	0.97	0.01	0.98	0.01	0.97	0.01	0.96	0.02	0.95	0.02	0.97	0.02	0.99	0.01
DS8	0.98	0.01	0.97	0.01	0.99	0.01	0.99	0.01	0.98	0.01	0.99	0.01	1.00	0.00
DS9	0.63	0.03	0.58	0.02	0.61	0.02	0.62	0.03	0.63	0.02	0.62	0.02	0.91	0.01
DS10	0.64	0.02	0.74	0.02	0.73	0.03	0.72	0.05	0.73	0.02	0.71	0.03	0.93	0.02
DS11	0.86	0.02	0.84	0.02	0.81	0.03	0.81	0.02	0.82	0.02	0.81	0.02	0.96	0.01
DS12	0.92	0.01	1.00	0.00	0.98	0.01	0.98	0.01	0.98	0.01	0.98	0.01	0.89	0.01
DS13	0.77	0.06	0.81	0.05	0.74	0.05	0.76	0.04	0.74	0.04	0.73	0.03	0.96	0.01
DS14	0.65	0.04	0.60	0.04	0.64	0.05	0.63	0.06	0.64	0.05	0.63	0.04	0.74	0.01
DS15	0.72	0.01	0.78	0.01	0.83	0.01	0.83	0.01	0.84	0.01	0.83	0.02	0.98	0.00
DS16	0.49	0.01	0.43	0.01	0.51	0.02	0.51	0.02	0.50	0.01	0.50	0.01	1.00	0.00
DS17	0.61	0.06	0.63	0.04	0.63	0.04	0.61	0.08	0.62	0.06	0.62	0.04	0.92	0.00
DS18	0.80	0.03	0.88	0.01	0.91	0.01	0.90	0.01	0.91	0.01	0.90	0.01	0.99	0.00
DS19	0.70	0.04	0.71	0.02	0.67	0.02	0.64	0.03	0.66	0.04	0.65	0.02	0.74	0.01
DS20	0.93	0.03	0.98	0.01	0.98	0.00	0.99	0.01	0.98	0.01	0.99	0.01	0.77	0.05
DS21	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.94	0.01
DS22	0.76	0.04	0.93	0.02	0.90	0.03	0.89	0.03	0.90	0.03	0.89	0.03	0.91	0.01
DS23	0.61	0.02	0.70	0.03	0.69	0.02	0.67	0.02	0.66	0.04	0.65	0.01	1.00	0.00
DS24	0.86	0.03	0.95	0.02	0.93	0.02	0.94	0.02	0.94	0.02	0.93	0.02	0.98	0.01
DS25	0.62	0.01	0.71	0.04	0.68	0.03	0.69	0.04	0.66	0.04	0.67	0.03	0.81	0.02
DS26	0.85	0.02	0.91	0.03	0.94	0.03	0.94	0.02	0.92	0.02	0.92	0.02	0.85	0.02
DS27	0.94	0.02	0.95	0.01	0.94	0.02	0.94	0.02	0.93	0.02	0.94	0.02	0.98	0.01
DS28	0.97	0.04	0.92	0.05	1.00	0.01	1.00	0.01	1.00	0.01	1.00	0.01	0.80	0.02
DS29	0.98	0.03	0.92	0.06	0.99	0.01	0.99	0.01	0.99	0.02	0.99	0.01	0.94	0.01
DS30	0.65	0.02	0.72	0.02	0.66	0.03	0.66	0.03	0.66	0.02	0.66	0.02	0.95	0.01
DS31	0.80	0.02	0.82	0.04	0.84	0.02	0.83	0.01	0.84	0.02	0.83	0.02	1.00	0.00
DS32	0.65	0.03	0.56	0.03	0.62	0.06	0.63	0.04	0.60	0.04	0.62	0.03	0.99	0.00
DS33	0.73	0.05	0.73	0.05	0.81	0.07	0.82	0.05	0.79	0.05	0.82	0.03	0.96	0.02
Average	**0.77**	**0.04**	**0.78**	**0.04**	**0.79**	**0.04**	**0.79**	**0.04**	**0.79**	**0.04**	**0.78**	**0.03**	**0.92**	**0.04**

Table 7. KNN classification results using the F-score, with shaded cells with gray indicate the best performance of the proposed method compared to EE.

Dataset ID	Proposed		EE		SMOTE		ADASYN		SVMSMOTE		Borderline-SMOTE		SMOTEFUNA	
	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD	AVG	±STD
DS1	0.47	0.07	0.50	0.09	0.52	0.09	0.52	0.09	0.57	0.11	0.54	0.10	0.95	0.03
DS2	0.75	0.15	0.62	0.10	0.65	0.08	0.62	0.10	0.72	0.13	0.73	0.11	0.93	0.05
DS3	0.69	0.07	0.68	0.06	0.59	0.08	0.60	0.05	0.66	0.07	0.62	0.07	0.90	0.04
DS4	0.78	0.15	0.78	0.15	0.76	0.14	0.76	0.14	0.81	0.17	0.77	0.15	0.88	0.05
DS5	0.61	0.15	0.56	0.05	0.65	0.09	0.65	0.10	0.65	0.10	0.64	0.09	0.91	0.02
DS6	0.77	0.02	0.99	0.01	0.90	0.01	0.90	0.01	0.89	0.01	0.89	0.01	0.97	0.01
DS7	0.86	0.04	0.99	0.01	0.92	0.03	0.92	0.02	0.92	0.03	0.92	0.02	0.98	0.01
DS8	0.72	0.03	0.97	0.02	0.79	0.04	0.78	0.03	0.79	0.04	0.79	0.04	0.98	0.01
DS9	0.62	0.02	0.59	0.02	0.62	0.02	0.61	0.02	0.64	0.03	0.63	0.03	0.93	0.01
DS10	0.60	0.03	0.74	0.03	0.69	0.03	0.68	0.03	0.69	0.03	0.69	0.03	0.90	0.02
DS11	0.86	0.02	0.85	0.02	0.82	0.02	0.80	0.02	0.81	0.02	0.81	0.01	0.98	0.01
DS12	0.94	0.02	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.91	0.01
DS13	0.80	0.03	0.79	0.05	0.77	0.04	0.75	0.03	0.76	0.02	0.76	0.02	0.98	0.00
DS14	0.68	0.04	0.61	0.02	0.68	0.05	0.69	0.04	0.71	0.06	0.71	0.05	0.76	0.01
DS15	0.75	0.01	0.78	0.01	0.85	0.01	0.84	0.01	0.85	0.01	0.84	0.01	0.98	0.00
DS16	0.46	0.01	0.44	0.02	0.52	0.02	0.52	0.02	0.51	0.02	0.51	0.02	0.94	0.01
DS17	0.61	0.04	0.62	0.03	0.64	0.04	0.64	0.03	0.66	0.04	0.66	0.04	0.92	0.01
DS18	0.78	0.01	0.88	0.01	0.92	0.01	0.90	0.01	0.92	0.01	0.92	0.01	1.00	0.00
DS19	0.72	0.03	0.74	0.03	0.66	0.03	0.66	0.03	0.67	0.04	0.67	0.03	0.78	0.02
DS20	0.60	0.03	0.99	0.00	0.97	0.01	0.97	0.01	0.96	0.01	0.98	0.00	0.82	0.03
DS21	1.00	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.93	0.01
DS22	0.68	0.02	0.93	0.01	0.88	0.02	0.88	0.02	0.88	0.02	0.88	0.02	0.91	0.01
DS23	0.59	0.02	0.71	0.02	0.67	0.02	0.67	0.02	0.67	0.02	0.67	0.02	1.00	0.00
DS24	0.69	0.05	0.96	0.01	0.92	0.01	0.92	0.01	0.91	0.01	0.92	0.01	0.98	0.00
DS25	0.61	0.02	0.71	0.03	0.66	0.01	0.67	0.02	0.67	0.02	0.66	0.02	0.77	0.02
DS26	0.80	0.02	0.90	0.03	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.83	0.02
DS27	0.97	0.01	0.96	0.01	0.97	0.01	0.96	0.01	0.96	0.01	0.96	0.01	1.00	0.00
DS28	0.66	0.05	0.91	0.05	0.72	0.05	0.71	0.04	0.73	0.05	0.74	0.06	0.77	0.03
DS29	0.65	0.03	0.93	0.04	0.75	0.05	0.74	0.04	0.80	0.05	0.80	0.03	0.93	0.01
DS30	0.60	0.03	0.71	0.02	0.66	0.03	0.64	0.03	0.66	0.03	0.65	0.03	0.95	0.00
DS31	0.84	0.03	0.81	0.02	0.83	0.01	0.83	0.02	0.83	0.02	0.83	0.02	0.95	0.01
DS32	0.64	0.03	0.56	0.02	0.61	0.03	0.61	0.03	0.61	0.04	0.61	0.05	0.99	0.00
DS33	0.72	0.05	0.70	0.04	0.77	0.05	0.77	0.05	0.77	0.05	0.77	0.05	0.96	0.01
Average	**0.71**	**0.04**	**0.79**	**0.03**	**0.77**	**0.03**	**0.76**	**0.03**	**0.78**	**0.04**	**0.77**	**0.04**	**0.92**	**0.04**

When excluding the results of SMOTEFUNA, the proposed RDPVR outperforms the other methods on four datasets, namely DS3, DS13, DS31, and DS32, when SVM is used. In comparison to other methods, EE is also an efficient resampling method. Comparing to EE, the proposed RDPVR is better in 11 datasets, and it obtains comparable results in the other datasets; therefore, we make a direct comparison between the RDPVR and EE in Figure 3. A closer inspection of this graph reveals that the proposed RDPVR and EE yield comparable results.

Compared to the SVM classifier, the Naive Bayes (NB) classifier produced significantly better F-score results. Table 5 shows the NB classifier's F-score results after applying each resampling method to all datasets.

Table 5 shows that the proposed RDPVR results are better than other methods on 11 datasets, excluding the SMOTEFUNA results. On the other hand, on 19 datasets, the EE method achieved better. As shown in Table 5, when using NB for classification, the RDPVR's performance improves significantly in terms of the F-score (from 74% to 76% on average), achieving competitive results as shown in Figure 4, where the proposed RDPVR's curve is almost identical to that of the EE resampling method, which was possibly due to the similar voting system used by both methods.

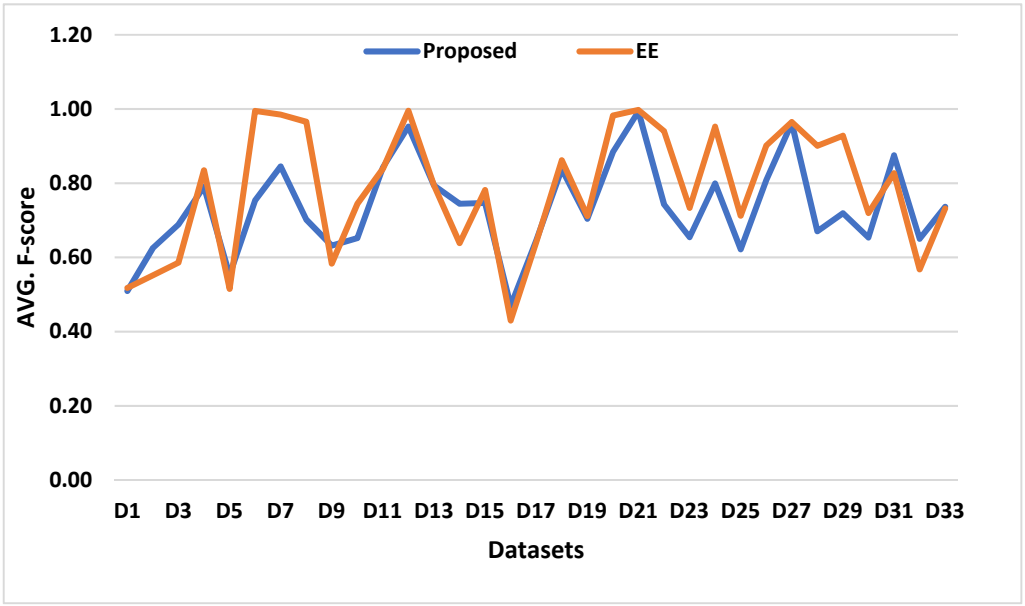

Figure 3. SVM-based F-score comparison of RDPVR and EE methods on all datasets.

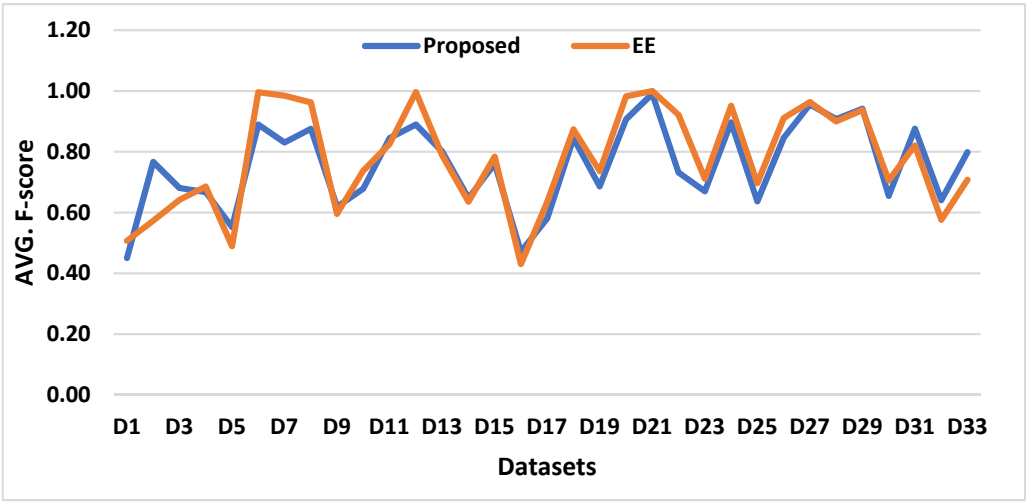

Figure 4. NB-based F-score comparison of RDPVR and EE methods on all datasets.

As can be seen in Tables 4 and 5, the classification results are influenced by the classifier used to some extent, so we decided to run more tests with more classifiers, namely, CART and KNN.

On 12 datasets, we see an improvement in the F-score obtained by CART when using the proposed RDPVR generated subdatasets; the results are higher than both SVM and NB (77% on average). The F-score results obtained by all classifiers on all resampled data generated by the resampling methods compared, including the proposed RDPVR, are shown in Table 6.

As shown in Tables 4–6, SMOTEFUNA has significantly better performance, even with different classifiers. RDPVR and EE methods, on the other hand, show comparable results as with the previous classifiers, however, with better performance than NB and SVM when using CART. It is worth noting that the proposed RDPVR resampling method recorded a perfect F-score (see DS21 in Table 6 and Figure 5).

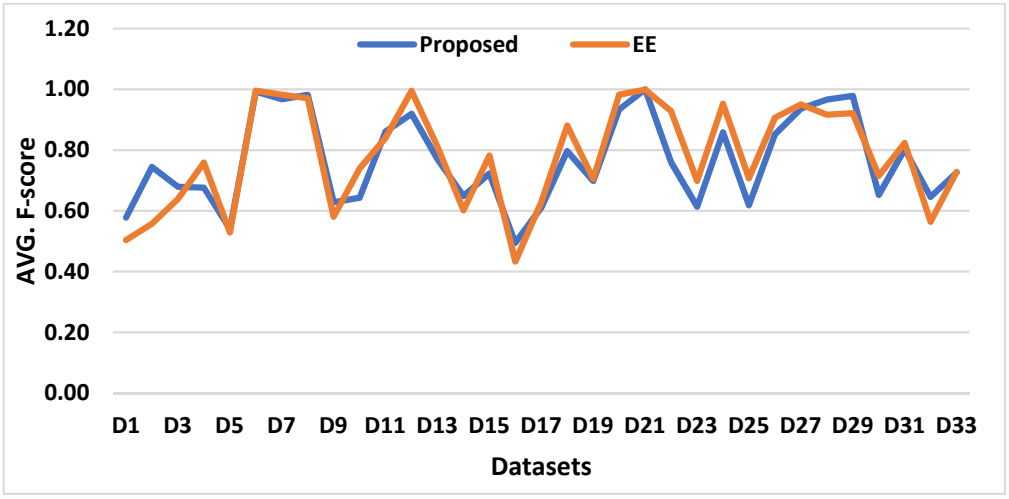

Figure 5. CART-based F-score comparison of RDPVR, EE, and SMOTEFUNA methods on all datasets.

As can be seen in Table 7, which displays KNN (k = 1) classification results using the F-score, the performance of the proposed RDPVR is detracted if compared to its results obtained by the previous classifiers, recording only a 71% F-score on average. Figure 6 shows that when RDPVR and EE were compared using KNN, the results were comparable, and this is similar to the previous findings in Figures 3–5.

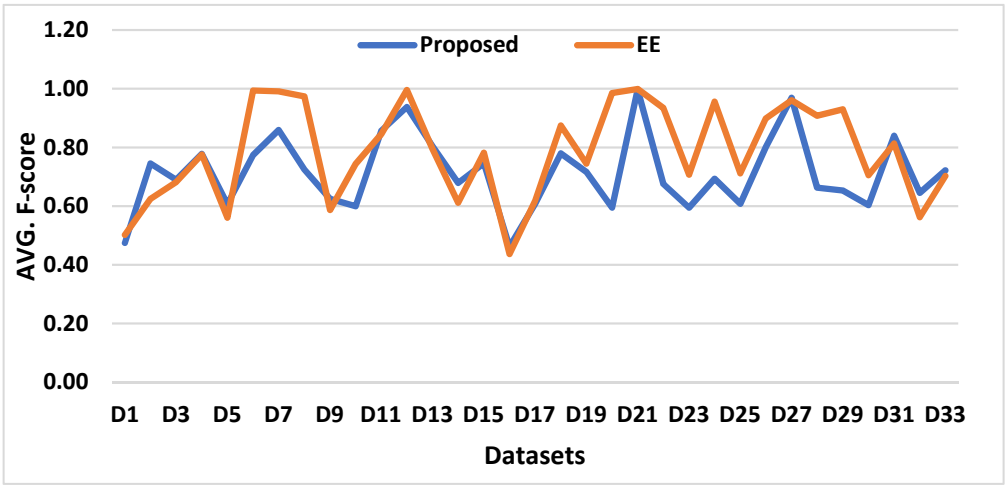

Figure 6. KNN-based F-score comparison of RDPVR, EE, and SMOTEFUNA methods on all datasets.

The performance of the SMOTEFUNA resampling method was not affected by any of the classifiers used, as shown in Tables 4–7 and Figures 3–6, and it achieved the best results compared to all the resampling methods investigated. On the contrary, we notice that the classifier had a significant impact on all other resampling methods, including our proposed method. We also notice that the second place can go to either EE or the proposed RDPVR in most cases, as both methods have roughly similar performance. We attribute this to the core of both methods, as they both use a voting rule on a number of subdatasets, and both methods employ the majority of examples from the majority class as well as those from the minority class.

Most of the oversampling methods, namely SMOTE, ADASYN, SVMSMOTE, and Borderline-SMOTE, are faster than both EE and the proposed RDPVR, according to the time comparison. This is to be expected, given that these approaches only utilize one training dataset, whereas EE uses ten estimators and RDPVR (in this experiment) uses ten subdatasets. Figure 7 compares the training times of RDPVE and EE only for the same reason. The SMOTEFUNA is an exception, which is to be expected given that it requires

quadratic time to find the farthest point (example) in the feature space while using only one training—partially generated—dataset.

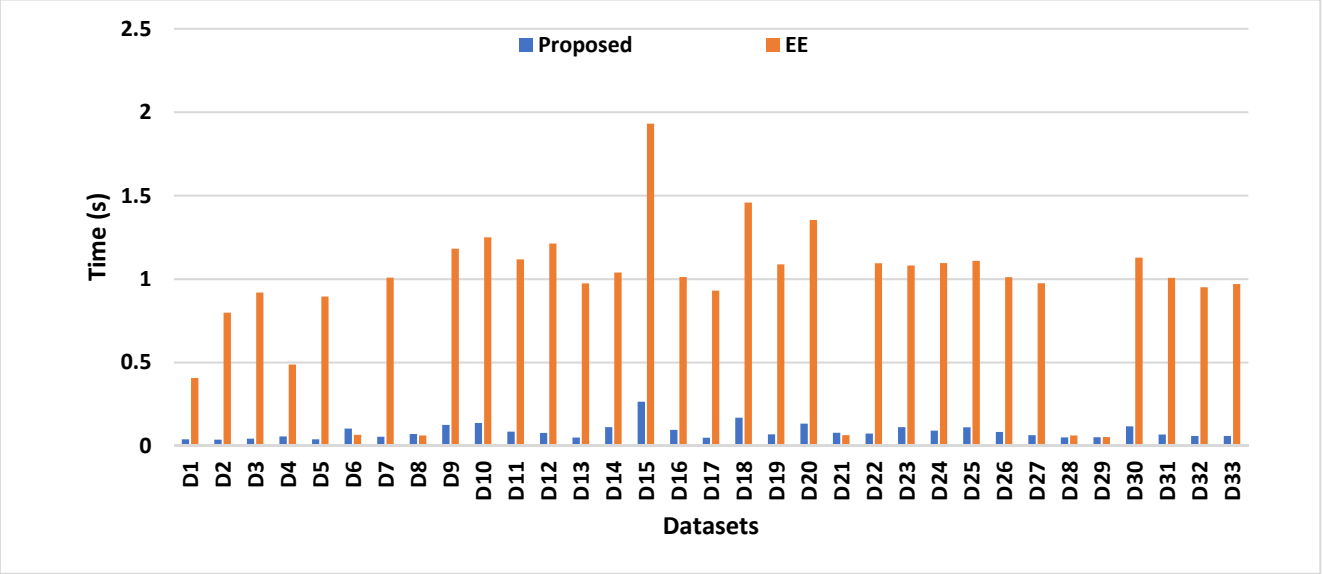

Figure 7. Training time comparison of the RDPVE and EE undersampling methods, using CART classifier.

The proposed method is faster than the EE, as seen in Figure 7. This is most likely due to the EE's time-consuming endeavor to find the best deterministic examples for its estimators, whereas ours select them at random. However, in order to make the comparison fair, we used only ten subdatasets for the proposed RDPVE, because RDPVE achieves reasonable results with only ten subdatasets, as shown in Table 3, and the EE already uses ten estimators.

6. Conclusions

In this paper, we propose a random data partitioning with majority voting rule for undersampling machine learning class-imbalanced datasets. The proposed method divides an imbalanced dataset into a number of small subdatasets, each of which is enforced to be class balanced. Then, a specific classifier is trained on each subdatasets, and the final classification result is obtained by performing the majority rule voting on all the results obtained by the trained models.

The proposed method takes linear time and, in some cases, constant time, especially when the number of examples in the minority class is small in comparison to the total number of examples in the training dataset. We chose an undersampling-like approach because we believe that the oversampling approach is problematic because it may increase the probability of overfitting.

On 33 benchmark machine learning class-imbalanced datasets, we evaluated the performance of the proposed method to some of the most well-known oversampling and undersampling methods, employing a variety of classifiers. Except for the SVMSMOTE, which is an oversampling method that may overfit the learning process, the classification results obtained by the classifiers employed on the generated data by the proposed method were better than most of the resampling methods tested.

Apart from that, the proposed method performed the best in at least 11 datasets scoring 71% to 77% average F-score depending on the classifier used. In general, the results of the proposed method were comparable to that of the EE, which is also an undersampling method. Therefore, we recommend the use of any of these undersampling methods for solving the problem of machine learning from class-imbalanced datasets.

The proposed method's weakness is obviously its classification accuracy, since it is outperformed by one of the oversampling methods, which we do not care about as much as discussed earlier, but it is also outperformed by one of the undersampling methods on occasion. The explanation could be due to a random selection of the majority cases in a certain subdataset. As a result, we need to come up with a better technique for such selection, aiming to preserve the most deterministic majority example possible while applying some criteria. Another weakness of the proposed method is its lack of ability to optimally determine the number of subdatasets, since we observed that this varies depending on the type of datasets used. Therefore, we need to develop a better strategy to propose the best number of subdatasets based on the trained data browning techniques from [144–146].

Our future plan to improve the proposed method's performance will focus mostly on its weaknesses, namely, accuracy and determining the appropriate number of subdatasets.

Author Contributions: Conceptualization and methodology, A.B.H., M.A. (Malek Alrashid) and M.A. (Mansoor Alghamdi); implementation, A.S.T. and S.S.A.; data curation, G.A.A.; writing—original draft preparation, all authors; writing—review and editing, all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the datasets used in this paper are publicly available at the Kaggle website: https://www.kaggle.com (accessed on 1 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 463–484. [CrossRef]
2. He, H.; Ma, Y. *Imbalanced Learning: Foundations, Algorithms, and Applications*; Wiley-IEEE Press: Hoboken, NJ, USA, 2013.
3. Wu, J.; Zhao, Z.; Sun, C.; Yan, R.; Chen, X. Learning from Class-imbalanced Data with a Model-Agnostic Framework for Machine Intelligent Diagnosis. *Reliab. Eng. Syst. Saf.* **2021**, *216*, 107934. [CrossRef]
4. Peng, M. Trainable undersampling for class-imbalance learning. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 1 July 2019.
5. Tarawneh, A.S.; Hassanat, A.B.A.; Almohammadi, K.; Chetverikov, D.; Bellinger, C. SMOTEFUNA: Synthetic Minority Over-Sampling Technique Based on Furthest Neighbour Algorithm. *IEEE Access* **2020**, *8*, 59069–59082. [CrossRef]
6. Hassanat, A.; Jassim, S. Visual Words for Lip-Reading. In *Mobile Multimedia/Image Processing, Security, and Applications 2010*; SPIE Press: Orlando, FL, USA, 2010.
7. Hassanat, A.B.A. Visual Speech Recognition. In *Speech and Language Technologies*; InTech: London, UK, 2011; pp. 279–304.
8. Hassanat, A.; Btoush, E.; Abbadi, M.; Al-Mahadeen, B.; Al-Awadi, M.; Mseidein, K.; Almseden, A.; Tarawneh, A.; Alhasanat, M.; Prasath, V.; et al. Victory sign biometrie for terrorists identification: Preliminary results. In Proceedings of the 2017 8th International Conference on Information and Communication Systems, ICICS 2017, Irbid, Jordan, 4–6 April 2017.
9. Hassanat, A. On Identifying Terrorists Using Their Victory Signs. *Data Sci. J.* **2018**, *17*, 27. [CrossRef]
10. Tarawneh, A.S.; Chetverikov, D.; Verma, C.; Hassanat, A.B. Stability and reduction of statistical features for image classification and retrieval: Preliminary results. In Proceedings of the 2018 9th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 3–5 April 2018.
11. Al-Shamaileh, M.Z.; Hassanat, A.B.; Tarawneh, A.S.; Rahman, M.S.; Celik, C.; Jawthari, M. New Online/Offline text-dependent Arabic Handwriting dataset for Writer Authentication and Identification. In Proceedings of the 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 11–13 June 2019.
12. Hassanat, A.; Al-Awadi, M.; Btoush, E.; Btoush, A.A.; Alhasanat, E.; Altarawneh, G. New Mobile Phone and Webcam Hand Images Databases for Personal Authentication and Identification. *Procedia Manuf.* **2015**, *3*, 4060–4067. [CrossRef]
13. Al-Btoush, A.I.; Abbadi, M.A.; Hassanat, A.B.; Tarawneh, A.S.; Hasanat, A.; Prasath, V.B.S. New Features for Eye-Tracking Systems: Preliminary Results. In Proceedings of the 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 11–13 June 2019.
14. Alhasanat, S.M.M.; Prasath, V.S.; Al Mahadeen, B.M.; Hassanat, A. Classification and gender recognition from veiled-faces. *Int. J. Biom.* **2017**, *9*, 347. [CrossRef]
15. Hammad, M.; Alkinani, M.H.; Gupta, B.B.; El-Latif, A.A.A. Myocardial infarction detection based on deep neural network on imbalanced data. *Multimed. Syst.* **2021**, 1–13. [CrossRef]

16. Fatima, M.; Pasha, M. Survey of Machine Learning Algorithms for Disease Diagnostic. *J. Intell. Learn. Syst. Appl.* **2017**, *9*, 1–16. [CrossRef]
17. Alqatawneh, A.; Alhalaseh, R.; Hassanat, A.; Abbadi, M. Statistical-Hypothesis-Aided Tests for Epilepsy Classification. *Computers* **2019**, *8*, 84. [CrossRef]
18. Aseeri, M.; Hassanat, A.B.; Mnasri, S.; Tarawneh, A.S.; Alhazmi, K.; Altarawneh, G.; Alrashidi, M.; Alharbi, H.; Almohammadi, K.; Chetverikov, D.; et al. Modelling-based Simulator for Forecasting the Spread of COVID-19: A Case Study of Saudi Arabia. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **2020**, *20*, 114.
19. Hassanat, A.B.; Mnasri, S.; Aseeri, M.; Alhazmi, K.; Cheikhrouhou, O.; Altarawneh, G.; Alrashidi, M.; Tarawneh, A.S.; Almohammadi, K.; Almoamari, H. A simulation model for forecasting covid-19 pandemic spread: Analytical results based on the current saudi covid-19 data. *Sustainability* **2021**, *13*, 4888. [CrossRef]
20. Fiore, U.; De Santis, A.; Perla, F.; Zanetti, P.; Palmieri, F. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Inf. Sci.* **2019**, *479*, 448–455. [CrossRef]
21. Ghatasheh, N.; Faris, H.; Abukhurma, R.; Castillo, P.A.; Al-Madi, N.; Mora, A.M.; Al-Zoubi, A.M.; Hassanat, A. Cost-sensitive ensemble methods for bankruptcy prediction in a highly imbalanced data distribution: A real case from the Spanish market. *Prog. Artif. Intell.* **2020**, *9*, 361–375. [CrossRef]
22. Xu, H.; Zhang, C.; Hong, G.S.; Zhou, J.; Hong, J.; Woon, K.S. Gated Recurrent Units Based Neural Network for Tool Condition Monitoring. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.
23. Mnasri, S.; Bossche, A.V.D.; Nasri, N.; Val, T. The 3D Redeployment of Nodes in Wireless Sensor Networks with Real Testbed Prototyping. In Proceedings of the International Conference on Ad-Hoc Networks and Wireless, Messina, Italy, 17–18 September 2017; pp. 18–24.
24. Mnasri, S.; Nasri, N.; Val, T. The 3D indoor deployment in DL-IoT with experimental validation using a particle swarm algorithm based on the dialects of songs. In Proceedings of the 2018 14th International Wireless Communications and Mobile Computing Conference, IWCMC 2018, Limassol, Cyprus, 25–29 June 2018.
25. Mnasri, S.; van den Bossche, A.; Narsi, N.; Val, T. The 3D Deployment Multi-objective Problem in Mobile WSN: Optimizing Coverage and Localization. *Int. Res. J. Innov. Eng.-IRJIE* **2015**, *1*, 1–14.
26. Mnasri, S.; Nasri, N.; AlRashidi, M.; Bossche, A.V.D.; Val, T. IoT networks 3D deployment using hybrid many-objective optimization algorithms. *J. Heuristics* **2020**, *26*, 663–709. [CrossRef]
27. Abdallah, W.; Mnasri, S.; Val, T. Genetic-Voronoi algorithm for coverage of IoT data collection networks. In Proceedings of the 30th International Conference on Computer Theory and Applications, ICCTA 2020-Proceedings, Alexandria, Egypt, 12–14 December 2020.
28. Abdallah, W.; Mnasri, S.; Nasri, N.; Val, T. Emergent IoT Wireless Technologies beyond the year 2020: A Comprehensive Comparative Analysis. In Proceedings of the 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 9–10 September 2020.
29. Mnasri, S.; Nasri, N.; Bossche, A.V.D.; Val, T. A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem. *ISA Trans.* **2019**, *91*, 262–280. [CrossRef]
30. Mnasri, S.; Abbes, F.; Zidi, K.; Ghedira, K. A multi-objective hybrid BCRC-NSGAII algorithm to solve the VRPTW. In Proceedings of the 13th International Conference on Hybrid Intelligent Systems, HIS 2013, Gammarth, Tunisia, 4–6 December 2013.
31. Tlili, S.; Mnasri, S.; Val, T. A multi-objective Gray Wolf algorithm for routing in IoT Collection Networks with real experiments. In Proceedings of the 2021 IEEE 4th National Computing Colleges Conference, NCCC 2021, Taif, Saudi Arabia, 27–28 March 2021.
32. Mnasri, S.; Nasri, N.; van den Bossche, A.; Val, T. A Hybrid Ant-Genetic Algorithm to Solve a Real Deployment Problem: A Case Study with Experimental Validation. In Proceedings of the International Conference on Ad-Hoc Networks and Wireless, Messina, Italy, 20–22 September 2017; pp. 367–381.
33. Mnasri, S.; Nasri, N.; van den Bossche, A.; Val, T. A comparative analysis with validation of NSGA-III and MOEA/D in resolving the 3D indoor redeployment problem in DL-IoT. In Proceedings of the 2017 International Conference on Internet of Things, Embedded Systems and Communications, IINTEC 2017-Proceedings, Gafsa, Tunisia, 20–22 October 2017.
34. Alghamdi, M.; Teahan, W. Experimental evaluation of Arabic OCR systems. *PSU Res. Rev.* **2017**, *1*, 229–241. [CrossRef]
35. Hassanat, A.B.; Altarawneh, G. 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.* **2014**, *8*, 1630–1638. [CrossRef]
36. Al-Kasassbeh, M.; Khairallah, T. Winning tactics with DNS tunnelling. *Netw. Secur.* **2019**, *2019*, 12–19. [CrossRef]
37. Al-Naymat, G.; Al-Kasassbeh, M.; Al-Hawari, E. Using machine learning methods for detecting network anomalies within SNMP-MIB dataset. *Int. J. Wirel. Mob. Comput.* **2018**, *15*, 67–76. [CrossRef]
38. Zuraiq, A.A.; Alkasassbeh, M. Review: Phishing Detection Approaches. In Proceedings of the 2019 2nd International Conference on New Trends in Computing Sciences, ICTCS 2019-Proceedings, Amman, Jordan, 9–11 October 2019.
39. Almseidin, M.; Abu Zuraiq, A.; Al-Kasassbeh, M.; Alnidami, N. Phishing Detection Based on Machine Learning and Feature Selection Methods. *Int. J. Interact. Mob. Technol. (iJIM)* **2019**, *13*, 171–183. [CrossRef]
40. Abuzuraiq, A.; Alkasassbeh, M.; Almseidin, M. Intelligent Methods for Accurately Detecting Phishing Websites. In Proceedings of the 2020 11th International Conference on Information and Communication Systems, ICICS 2020, Irbid, Jordan, 7–9 April 2020.

41. Almseidin, M.; Piller, I.; Alkasassbeh, M.; Kovacs, S. Fuzzy Automaton as a Detection Mechanism for the Multi-Step Attack. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2019**, *9*, 575–586. [CrossRef]
42. Al-Kasassbeh, M.; Mohammed, S.; Alauthman, M.; Almomani, A. Feature selection using a machine learning to classify a malware. In *Handbook of Computer Networks and Cyber Security*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 889–904.
43. Almseidin, M.; Al-Kasassbeh, M.; Kovacs, S. Detecting Slow Port Scan Using Fuzzy Rule Interpolation. In Proceedings of the 2019 2nd International Conference on New Trends in Computing Sciences, ICTCS 2019-Proceedings, Amman, Jordan, 9–11 October 2019.
44. Alothman, Z.; Alkasassbeh, M.; Baddar, S.A.-H. An efficient approach to detect IoT botnet attacks using machine learning. *J. High Speed Netw.* **2020**, *26*, 241–254. [CrossRef]
45. Al Hawawreh, M.; Rawashdeh, A.; Alkasassbeh, M. An anomaly-based approach for DDoS attack detection in cloud environment. *Int. J. Comput. Appl. Technol.* **2018**, *57*, 312. [CrossRef]
46. Alkasassbeh, M. A Novel Hybrid Method for Network Anomaly Detection Based on Traffic Prediction and Change Point Detection. *J. Comput. Sci.* **2018**, *14*, 153–162. [CrossRef]
47. Hamadaqa, E.; Abadleh, A.; Mars, A.; Adi, W. Highly Secured Implantable Medical Devices. In Proceedings of the 2018 International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 18–19 November 2018.
48. Mulhem, S.; Abadleh, A.; Adi, W. Accelerometer-Based Joint User-Device Clone-Resistant Identity. In Proceedings of the 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 30–31 October 2018; pp. 230–237.
49. Mars, A.; Abadleh, A.; Adi, W. Operator and Manufacturer Independent D2D Private Link for Future 5G Networks. In Proceedings of the INFOCOM 2019-IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2019, Paris, France, 29 April–2 May 2019.
50. Alabadleh, A.; Alja'Afreh, S.; Aljaafreh, A.; Alawasa, K. A RSS-based localization method using HMM-based error correction. *J. Locat. Based Serv.* **2018**, *12*, 273–285. [CrossRef]
51. Aljaafreh, A.; Alawasa, K.; Alja'afreh, S.; Abadleh, A. Fuzzy inference system for speed bumps detection using smart phone accelerometer sensor. *J. Telecommun. Electron. Comput. Eng.* **2017**, *9*, 133–136.
52. Abadleh, A.; Al-Hawari, E.; Alkafaween, E.; Al-Sawalqah, H. Step Detection Algorithm for Accurate Distance Estimation Using Dynamic Step Length. In Proceedings of the 2017 18th IEEE International Conference on Mobile Data Management (MDM), Daejeon, Korea, 29 May–1 June 2017.
53. Abadleh, A.; Han, S.; Hyun, S.J.; Lee, B.; Kim, M. Construction of indoor floor plan and localization. *Wirel. Netw.* **2016**, *22*, 175–191. [CrossRef]
54. Hassanat, A.; Prasath, V.; Mseidein, K.; Al-Awadi, M.; Hammouri, A. A hybridwavelet-shearlet approach to robust digital imagewatermarking. *Informatica* **2017**, *41*, 3–24.
55. Hassanat, A.; Jassim, S. Color-based lip localization method. In Proceedings of the SPIE-The International Society for Optical Engineering, Orlando, FL, USA, 28 April 2010.
56. Hassanat, A.B.; Alkasassbeh, M.; Al-Awadi, M.; Alhasanat, E.A. Color-based object segmentation method using artificial neural network. *Simul. Model. Pract. Theory* **2016**, *64*, 3–17. [CrossRef]
57. Narloch, P.; Hassanat, A.; Altarawneh, A.S.A.; Anysz, H.; Kotowski, J.; Almohammadi, K. Predicting Compressive Strength of Cement-Stabilized Rammed Earth Based on SEM Images Using Computer Vision and Deep Learning. *Appl. Sci.* **2019**, *9*, 5131. [CrossRef]
58. Hassanat, A.; Prasath, V.B.S.; Alkasassbeh, M.; Altarawneh, A.S.A.; Al-Shamailh, A.J. Magnetic energy-based feature extraction for low-quality fingerprint images. *Signal Image Video Process.* **2018**, *12*, 1471–1478. [CrossRef]
59. Hassanat, A.B.; Alkasassbeh, M.; Al-Awadi, M.; Alhasanat, E.A. Colour-based lips segmentation method using artificial neural networks. In Proceedings of the 2015 6th International Conference on Information and Communication Systems, ICICS 2015, Irbid, Jordan, 7–9 April 2015.
60. Mansour, R.F.; Abdel-Khalek, S.; Hilali-Jaghdam, I.; Nebhen, J.; Cho, W.; Joshi, G.P. An intelligent outlier detection with machine learning empowered big data analytics for mobile edge computing. *Clust. Comput.* **2021**, 1–13. [CrossRef]
61. Aljehane, N.O.; Mansour, R.F. Optimal allocation of renewable energy source and charging station for PHEVs. *Sustain. Energy Technol. Assess.* **2021**, *49*, 101669. [CrossRef]
62. Mansour, R.F.; Escorcia-Gutierrez, J.; Gamarra, M.; Díaz, V.G.; Gupta, D.; Kumar, S. Artificial intelligence with big data analytics-based brain intracranial hemorrhage e-diagnosis using CT images. *Neural Comput. Appl.* **2021**, 1–13. [CrossRef]
63. Hassanat, A.B. Two-point-based binary search trees for accelerating big data classification using KNN. *PLoS ONE* **2018**, *13*, e0207772. [CrossRef]
64. Hassanat, A.B. Norm-Based Binary Search Trees for Speeding Up KNN Big Data Classification. *Computers* **2018**, *7*, 54. [CrossRef]
65. Hassanat, A.B. Furthest-Pair-Based Decision Trees: Experimental Results on Big Data Classification. *Information* **2018**, *9*, 284. [CrossRef]
66. Hassanat, A.B. Furthest-Pair-Based Binary Search Tree for Speeding Big Data Classification Using K-Nearest Neighbors. *Big Data* **2018**, *6*, 225–235. [CrossRef]
67. Wang, S.; Jiang, W.; Tsui, K.L. Adjusted support vector machines based on a new loss function. *Ann. Oper. Res.* **2008**, *174*, 83–101. [CrossRef]

68. Fernández, A.; García, S.; Galar, M.; Prati, R.; Krawczyk, B.; Herrera, F. *Learning from Imbalanced Data Sets*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2018; p. 83.
69. Branco, P.; Torgo, L.; Ribeiro, R.P. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.* **2016**, *49*, 1–50. [CrossRef]
70. Haixiang, G.; Yijing, L.; Shang, J.; Mingyun, G.; Yuanyue, H.; Bing, G. Learning from class-imbalanced data: Review of methods and applications. *Expert Syst. Appl.* **2017**, *73*, 220–239. [CrossRef]
71. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [CrossRef]
72. Tanha, J.; Abdi, Y.; Samadi, N.; Razzaghi, N.; Asadpour, M. Boosting methods for multi-class imbalanced data classification: An experimental review. *J. Big Data* **2020**, *7*, 1–47. [CrossRef]
73. Chawla, N.V.; Bowye, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
74. Drummond, C.; Holte, R.C. *C4.5*, Class Imbalance, and Cost Sensitivity: Why Under-Sampling Beats Over-Sampling. In Proceedings of the International Conference on Machine Learning (ICML 2003) Workshop on Learning from Imbalanced Datasets II, Washington, DC, USA, 1 July 2003.
75. Han, H.; Wang, W.Y.; Mao, B.H. *Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 878–887.
76. Tang, Y.; Zhang, Y.Q.; Chawla, N.V. SVMs modeling for highly imbalanced classification. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2009**, *39*, 281–288. [CrossRef]
77. Das, R. An Oversampling Technique by Integrating Reverse Nearest Neighbor in SMOTE: Reverse-SMOTE. In Proceedings of the International Conference on Smart Electronics and Communication, ICOSEC 2020, Trichy, India, 10–12 September 2020; pp. 1239–1244.
78. Liu, C. Constrained Oversampling: An Oversampling Approach to Reduce Noise Generation in Imbalanced Datasets with Class Overlapping. *IEEE Access* **2020**. [CrossRef]
79. Barua, S.; Islam, M.M.; Yao, X.; Murase, K. MWMOTE-Majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 405–425. [CrossRef]
80. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the International Joint Conference on Neural Networks, Hong Kong, China, 1–8 June 2008.
81. Bellinger, C. Framework for extreme imbalance classification: SWIM—sampling with the majority class. *Knowl. Inf. Syst.* **2019**, *62*, 841–866. [CrossRef]
82. Tian, C. A New Majority Weighted Minority Oversampling Technique for Classification of Imbalanced Datasets. In Proceedings of the 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2020, Fuzhou, China, 12–14 June 2020; pp. 154–157.
83. Domingos, P. MetaCost: A General Method for Making Classifiers Cost-Sensitive. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Lisboa, Portugal, 1 August 1999; Volume 55, pp. 155–164.
84. Kurniawati, Y.E. Adaptive Synthetic–Nominal (ADASYN–N) and Adaptive Synthetic–KNN (ADASYN- KNN) for Multiclass Imbalance Learning on Laboratory Test Data. In Proceedings of the 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesian, 7–8 August 2018; pp. 1–6.
85. Zhang, W.; Ramezani, R.; Naeim, A. WOTBoost: Weighted Oversampling Technique in Boosting for imbalanced learning. In Proceedings of the 2019 IEEE International Conference on Big Data, Big Data 2019, Los Angeles, CA, USA, 9–12 December 2019.
86. Kovács, G. Smote-variants: A python implementation of 85 minority oversampling techniques. *Neurocomputing* **2019**, *366*, 352–354. [CrossRef]
87. Raghuwanshi, B.S.; Shukla, S. SMOTE based class-specific extreme learning machine for imbalanced learning. *Knowl.-Based Syst.* **2020**, *187*, 104814. [CrossRef]
88. Douzas, G.; Bacao, F. Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning. *Expert Syst. Appl.* **2017**, *82*, 40–52. [CrossRef]
89. Pradipta, G.A.; Wardoyo, R.; Musdholifah, A.; Sanjaya, I.N.H. Radius-SMOTE: A New Oversampling Technique of Minority Samples Based on Radius Distance for Learning from Imbalanced Data. *IEEE Access* **2021**, *9*, 74763–74777. [CrossRef]
90. Krawczyk, B.; Koziarski, M.; Wozniak, M. Radial-based oversampling for multiclass imbalanced data classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 2818–2831. [CrossRef]
91. Hong, J.; Kang, H.; Hong, T. Oversampling-based prediction of environmental complaints related to construction projects with imbalanced empirical-data learning. *Renew. Sustain. Energy Rev.* **2020**, *134*, 110402. [CrossRef]
92. Ibrahim, M.H. ODBOT: Outlier detection-based oversampling technique for imbalanced datasets learning. *Neural Comput. Appl.* **2021**, *33*, 15781–15806. [CrossRef]
93. Wang, L.; Wang, H.; Fu, G. Multiple Kernel Learning with Minority Oversampling for Classifying Imbalanced Data. *IEEE Access* **2020**, *9*, 565–580. [CrossRef]
94. Bej, S.; Davtyan, N.; Wolfien, M.; Nassar, M.; Wolkenhauer, O. LoRAS: An oversampling approach for imbalanced datasets. *Mach. Learn.* **2021**, *110*, 279–301. [CrossRef]

95. Zhu, T.; Lin, Y.; Liu, Y. Improving interpolation-based oversampling for imbalanced data learning. *Knowl.-Based Syst.* **2020**, *187*, 104826. [CrossRef]
96. Douzas, G.; Bacao, F.; Last, F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Inf. Sci.* **2018**, *465*, 1–20. [CrossRef]
97. Faris, H.; Abukhurma, R.; Almanaseer, W.; Saadeh, M.; Mora, A.M.; Castillo, P.A.; Aljarah, I. Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: A case from the Spanish market. *Prog. Artif. Intell.* **2020**, *9*, 31–53. [CrossRef]
98. Jiang, Z.; Yang, J.; Liu, Y. Imbalanced Learning with Oversampling based on Classification Contribution Degree. *Adv. Theory Simul.* **2021**, *4*, 2100031. [CrossRef]
99. Douzas, G.; Bacao, F.; Fonseca, J.; Khudinyan, M. Imbalanced learning in land cover classification: Improving minority classes' prediction accuracy using the geometric SMOTE algorithm. *Remote Sens.* **2019**, *11*, 3040. [CrossRef]
100. Zhang, Y.; Li, X.; Gao, L.; Wang, L.; Wen, L. Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. *J. Manuf. Syst.* **2018**, *48*, 34–50. [CrossRef]
101. Wang, Z.; Wang, H. Global Data Distribution Weighted Synthetic Oversampling Technique for Imbalanced Learning. *IEEE Access* **2021**, *9*, 44770–44783. [CrossRef]
102. Liu, G.; Yang, Y.; Li, B. Fuzzy rule-based oversampling technique for imbalanced and incomplete data learning. *Knowl.-Based Syst.* **2018**, *158*, 154–174. [CrossRef]
103. Wu, X.; Yang, Y.; Ren, L. Entropy difference and kernel-based oversampling technique for imbalanced data learning. *Intell. Data Anal.* **2020**, *24*, 1239–1255. [CrossRef]
104. Engelmann, J.; Lessmann, S. Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Syst. Appl.* **2021**, *174*, 114582. [CrossRef]
105. Li, Q.; Li, G.; Niu, W.; Cao, Y.; Chang, L.; Tan, J.; Guo, L. Boosting imbalanced data learning with Wiener process oversampling. *Front. Comput. Sci.* **2016**, *11*, 836–851. [CrossRef]
106. Wang, C.R.; Shao, X.H. An Improving Majority Weighted Minority Oversampling Technique for Imbalanced Classification Problem. *IEEE Access* **2021**, *9*, 5069–5082. [CrossRef]
107. Malhotra, R.; Kamal, S. An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. *Neurocomputing* **2019**, *343*, 120–140. [CrossRef]
108. Kovács, G. An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Appl. Soft Comput.* **2019**, *83*. [CrossRef]
109. Dhurjad, R.K.; Banait, P.S.S. A survey on Oversampling Techniques for Imbalanced Learning. *Int. J. Appl. Innov. Eng. Manag.* **2014**, *3*, 279–284.
110. Li, J.; Zhu, Q.; Wu, Q.; Fan, Z. A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors. *Inf. Sci.* **2021**, *565*, 438–455. [CrossRef]
111. Jiang, Z.; Pan, T.; Zhang, C.; Yang, J. A new oversampling method based on the classification contribution degree. *Symmetry* **2021**, *13*, 194. [CrossRef]
112. Yao, B. An Improved Under-sampling Imbalanced Classification Algorithm. In Proceedings of the 2021 13th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2021, Beihai, China, 16–17 January 2021; pp. 775–779.
113. Guzmán-Ponce, A.; Valdovinos, R.M.; Sánchez, J.S.; Marcial-Romero, J.R. A new under-sampling method to face class overlap and imbalance. *Appl. Sci.* **2020**, *10*, 5164. [CrossRef]
114. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
115. Liu, X.-Y.; Wu, J.; Zhou, Z.-H. Exploratory under-sampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2008**, *39*, 539–550.
116. Chennuru, V.K.; Timmapareddy, S.R. Simulated annealing based undersampling (SAUS): A hybrid multi-objective optimization method to tackle class imbalance. *Appl. Intell.* **2021**, 1–19. [CrossRef]
117. Devi, D.; Biswas, S.k.; Purkayastha, B. Redundancy-driven modified Tomek-link based undersampling: A solution to class imbalance. *Pattern Recognit. Lett.* **2017**, *93*, 3–12. [CrossRef]
118. Koziarski, M. Radial-Based Undersampling for imbalanced data classification. *Pattern Recognit.* **2020**, *102*, 107262. [CrossRef]
119. Vuttipittayamongkol, P.; Elyan, E. Overlap-Based Undersampling Method for Classification of Imbalanced Medical Datasets. In *IFIP Advances in Information and Communication Technology*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 584.
120. Fan, Q.; Wang, Z.; Gao, D. One-sided Dynamic Undersampling No-Propagation Neural Networks for imbalance problem. *Eng. Appl. Artif. Intell.* **2016**, *53*, 62–73. [CrossRef]
121. Arefeen, M.A.; Nimi, S.T.; Rahman, M.S. Neural Network-Based Undersampling Techniques. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, 1–10. [CrossRef]
122. Vuttipittayamongkol, P.; Elyan, E. Neighbourhood-based undersampling approach for handling imbalanced and overlapped data. *Inf. Sci.* **2020**, *509*, 47–70. [CrossRef]

123. Devi, D.; Biswas, S.K.; Purkayastha, B. Learning in presence of class imbalance and class overlapping by using one-class SVM and undersampling technique. *Connect. Sci.* **2018**, *31*, 105–142. [CrossRef]
124. Guo, H.; Diao, X.; Liu, H. Improving undersampling-based ensemble with rotation forest for imbalanced problem. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 1371–1386. [CrossRef]
125. Vuttipittayamongkol, P.; Elyan, E. Improved Overlap-based Undersampling for Imbalanced Dataset Classification with Application to Epilepsy and Parkinson's Disease. *Int. J. Neural Syst.* **2020**, *30*. [CrossRef]
126. Ofek, N.; Rokach, L.; Stern, R.; Shabtai, A. Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem. *Neurocomputing* **2017**, *243*, 88–102. [CrossRef]
127. García, S.; Herrera, F. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evol. Comput.* **2009**, *17*, 275–306. [CrossRef] [PubMed]
128. Guo, H.; Zhou, J.; Wu, C.A. Ensemble learning via constraint projection and undersampling technique for class-imbalance problem. *Soft Comput.* **2020**, *24*, 4711–4727. [CrossRef]
129. Trisanto, D.; Rismawati, N.; Mulya, M.F.; Kurniadi, F.I. Effectiveness undersampling method and feature reduction in credit card fraud detection. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 173–181. [CrossRef]
130. Liu, B.; Tsoumakas, G. Dealing with class imbalance in classifier chains via random undersampling. *Knowl.-Based Syst.* **2020**, *192*, 105292. [CrossRef]
131. Onan, A. Consensus Clustering-Based Undersampling Approach to Imbalanced Learning. *Sci. Program.* **2019**, *2019*, 1–14. [CrossRef]
132. Kaur, P.; Gosain, A. Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. *Adv. Intell. Syst. Comput.* **2018**, *653*, 23–30.
133. Komamizu, T. Combining Multi-ratio Undersampling and Metric Learning for Imbalanced Classification. *J. Data Intell.* **2021**, *2*, 462–475. [CrossRef]
134. Lin, W.C.; Tsai, C.F.; Hu, Y.H.; Jhang, J.S. Clustering-based undersampling in class-imbalanced data. *Inf. Sci.* **2017**, *409–410*, 17–26. [CrossRef]
135. Nugraha, W.; Maulana, M.S.; Sasongko, A. Clustering Based Undersampling for Handling Class Imbalance in C4.5 Classification Algorithm. *J. Phys. Conf. Ser.* **2020**, *1641*, 012014. [CrossRef]
136. Akkasi, A.; Varoğlu, E.; Dimililer, N. Balanced undersampling: A novel sentence-based undersampling method to improve recognition of named entities in chemical and biomedical text. *Appl. Intell.* **2018**, *48*, 1965–1978. [CrossRef]
137. Sarkar, S.; Khatedi, N.; Pramanik, A.; Maiti, J. *An Ensemble Learning-Based Undersampling Technique for Handling Class-Imbalance Problem*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 605.
138. Rekha, G.; Reddy, V.K.; Tyagi, A.K. An Earth mover's distance-based undersampling approach for handling class-imbalanced data. *Int. J. Intell. Inf. Database Syst.* **2020**, *13*, 376–392.
139. Lingden, P.; Alsadoon, A.; Prasad, P.W.; Alsadoon, O.H.; Ali, R.S.; Nguyen, V.T.Q. A novel modified undersampling (MUS) technique for software defect prediction. *Comput. Intell.* **2019**, *35*, 1003–1020. [CrossRef]
140. Devi, D.; Biswas, S.K.; Purkayastha, B. A Review on Solution to Class Imbalance Problem: Undersampling Approaches. In Proceedings of the 2020 International Conference on Computational Performance Evaluation, ComPE 2020, Shillong, India, 2–4 July 2020.
141. Kang, Q.; Shi, L.; Zhou, M.C.; Wang, X.S.; Wu, Q.D.; Wei, Z. A Distance-Based Weighted Undersampling Scheme for Support Vector Machines and its Application to Imbalanced Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 4152–4165. [CrossRef]
142. Devi, D.; Namasudra, S.; Kadry, S. A boosting-aided adaptive cluster- based undersampling approach for treatment of class imbalance problem. *Int. J. Data Warehous. Min.* **2020**, *16*, 60–86. [CrossRef]
143. Research Group. What Is Colaboratory? Google Inc., 1 October 2021. [Online]. Available online: https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=GJBs_flRovLc (accessed on 5 November 2021).
144. Tarawneh, A.S.; Hassanat, A.B.; Celik, C.; Chetverikov, D.; Rahman, M.S.; Verma, C. Deep Face Image Retrieval: A Comparative Study with Dictionary Learning. In Proceedings of the 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 11–13 June 2019.
145. Hassanat, A.; Tarawneh, A. Fusion of color and statistic features for enhancing content-based image retrieval systems. *J. Theor. Appl. Inf. Technol.* **2016**, *88*, 1–12.
146. Tarawneh, A.S.; Celik, C.; Hassanat, A.B.; Chetverikov, D. Detailed investigation of deep features with sparse representation and dimensionality reduction in CBIR: A comparative study. *Intell. Data Anal.* **2020**, *24*, 47–68. [CrossRef]