Article

Data Augmentation Methods for Deep Learning Neural Networks

MD. Hasan Sharif * , Joni Hyttinen , Xiao-Zhi Gao * , Markku Hauta-Kasari , Dr Riikka Räisänen

Posted Date: 27 December 2024

doi: 10.20944/preprints202412.2343.v1

Keywords: Artificial Neural Network; Image Processing; Deep Learning Neural Network; Image Augmentation; Oversampling; Under-sampling

Article

Data Augmentation Methods for Deep Learning Neural Networks

Md Hasan Sharif [1,†,*], Joni Hyttinen [1,†], Xiao-Zhi Gao [1,†*], Markku Hauta-Kasari [1,†] and Riikka Räisänen [2,†]

[1] Department of Computer Science, University of Eastern Finland, Joensuu, 80130 North Karelia, Finland
[2] University of Helsinki, 00100 Helsinki, Finland
[*] Correspondence: swe.hs1994@gmail.com (M.H.S.); xiao-zhi.gao@uef.fi (X.-Z.G.)
[†] These authors contributed equally to this work.

Abstract: Standard algorithms face difficulties when learning from unbalanced datasets because they are built to handle balanced class distributions. Although there are various approaches to solving this issue, solutions that create false data represent a more all-encompassing strategy than algorithmic changes. In particular, they produce fictitious data that any algorithm can use without limiting the user's options. In this paper, we present five oversampling methods: Synthetic Minority Oversampling Technique (SMOTE), Random Over Sampling (ROS), K-Means Smote (KMS), Affinity Propagation and Random Over Sampling-Based Oversampling (APROSO), and Self-Organizing Map-based Oversampling (SOMO). We also present four undersampling methods: Random Under Sampling (RUS), Cluster Centroids (CCs), Neighborhood Cleaning Rule (NCR), Near Miss-1 (NM1). To evaluate those over and under-sampling methods, we have used two different Deep Neural Network (DNN) models, i.e., DNN model 1 and DNN model 2. The empirical result shows that all the over and under-sampling methods are providing more effective results on DNN model 2. The result analysis also shows that the oversampling methods are more effective in classifying the Magnoliopsida and Pinopsida images.

Keywords: artificial neural network; image processing; deep learning neural network; image augmentation; oversampling; under-sampling

1. Introduction

In our daily lives, color is significant in informing us, evoking moods, and even influencing our decisions. Additionally, color tastes impact the things we buy, our clothes, and how we decorate our spaces. Due to their widespread availability and affordability, synthetic dyes are often utilized extensively in various industries, including textile, paper, leather, cosmetics, food, etc. However, the lack of biodegradability of synthetic colorants results in environmental pollution and health risks for people and other living things [1,2]. Naturally occurring colorants from renewable biological sources, including plants, algae, fungi, and microorganisms, are safer thanks to their low allergenic and non-toxic qualities. They are also more environmentally friendly and sustainable [3]. The Biocolour project addresses challenges in the biocolourant industry, uniting international researchers from diverse fields

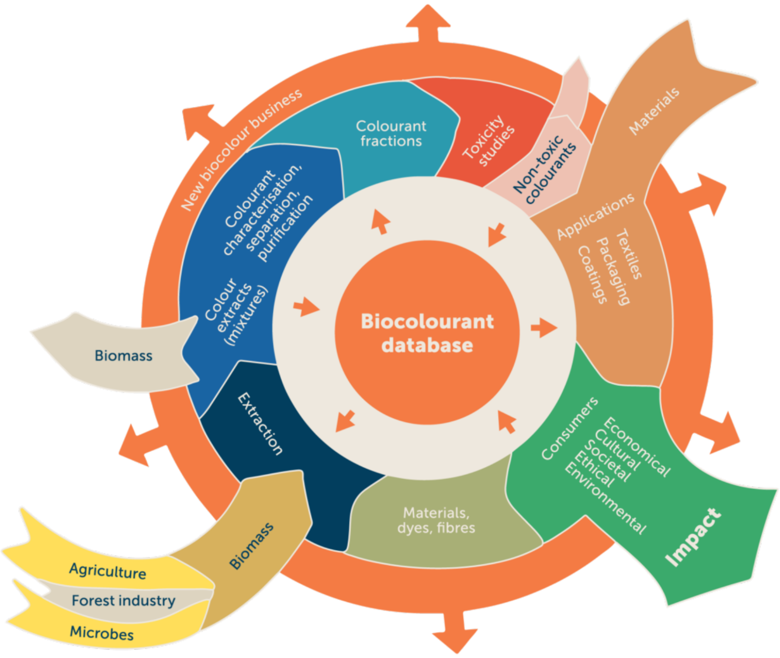


Figure 1. Biocolour project progression

The consortium focuses on extraction, production, safety assessments, industrial applications, and commercialization of biocolourants. Aimed at promoting biocolour adoption in Finland, the project seeks to enhance safety and well-being. Its primary goal is sustainable colorant creation for industry, building on prior research with class imbalance issues in the dataset.In machine learning, the term "class imbalance problem" refers to classification jobs where data classes are not equally represented. The majority class has a large majority of observations, while the others are the minority classes. The ratio of each minority class to the majority class is known as the imbalance ratio (IR). Learning from imbalanced data is a significant issue for the scientific community and business professionals. For many practitioners, imbalanced data is "pervasive and ubiquitous," posing a difficult challenge [4]. Their significance is demonstrated by the recent increase in study interest in this area [4–6]. For many different types of machines practical uses and learning activities, such as systems for retrieving information, detection of fraud, medical evaluation, data communications, and direct marketing, handling this kind of data and still being able to build accurate forecasts is a significant issue [7]. A previous study by [8] has highlighted credit scoring as a class imbalance problem with unequal sample distributions in which the number of non-risky applicants is typically substantially larger than that of credit defaulters. In recent years, machine learning and data mining techniques have been introduced to improve financial decision-making prediction accuracy, lowering credit analysis costs, enabling quicker loan decisions, guaranteeing payment collections, and providing adequate risk mitigation [9]. As a result, good models often be in need of the ability to learn from (very) imbalanced data. According to [10], the price of miscategorized for the minority class is frequently superior to that of the majority class [11]. An illustration is the discovery of uncommon or even undiscovered particles in fact-finding energetic physics [12]. Only a little portion the information generated by particle impact in the actuator may be seen in these encouraging samples. In this instance, the experiment fails to uncover novel, intriguing events due to the classifier's low performance in detecting uncommon particles. Based on theoretical justifications, the reverse case of misclassifying known constituents as uncommon doesn't incur a significant expense. Due to their assumptions of equal misclassification costs or balanced class distributions, conventional learning strategies perform imperfectly in imbalanced data sets. As previously stated, the minority classes are those in real-world problems where selecting the classifier is to accomplish the best accuracy [13]. For two key reasons, applying standard algorithms results in a bias in favor of the dominant class. First, minority classes in any classifier contribute less during the training phase to minimize the objective function of the classifier, and second, it is frequently

doi:10.20944/preprints202412.2343.v1

challenging to distinguish in the middle of noise instances and minority class instances. Additionally, it does not address the possibility that the minority classes' misclassification costs perhaps greater than such as the dominant class. The topic of class inequality has been handled by the machine learning community in three different approaches. One involves changing the distribution of classes at the data level via under-sampling, over-sampling, or hybrid methods. The design or modification of algorithms that support learning for the minority class is the second. The final strategy involves using cost-sensitive strategies to reduce higher cost errors based on data or algorithms level [14]. In contrast to algorithmic changes, strategies that deal with the issue by altering the data level—in particular, over-sampling methods that do so by creating artificial data—constitute a more all-encompassing approach. Particularly, they produce fake data that any algorithm can employ throughout the learning stage. By keeping all the data in the minority class and reducing the size of the majority class, under-sampling is a method for balancing unequal data sets. In this research, we have compared the performance of five oversampling methods and four undersampling methods for imbalanced image classification problems. The presented over-sampling methods are SMOTE, ROS, KMS, APROS, and SOMO. On the other hand, the under-sampling methods are RUS, CCs, NCR, and NM1.In this paper, we describe the previous work conducted on artificial intelligence (AI) and provide a detailed survey on oversampling and undersampling techniques. In this paper, we reveal the working architecture of our implemented system. Section 4 displays the results of the experimental evaluations of our five oversampling and four undersampling techniques. Finally, in the discussion section, we uncover the most effective techniques for oversampling and undersampling.

2. Related Work

In this section, we provide a brief overview of various methods focusing on the alteration of the data level, often referred to as the sampling system. An assessment of alternative systems can be found in [15]. Sampling techniques can generally be categorized into two groups: under-sampling and oversampling. Undersampling reduces the number of examples from the majority class by removing samples from the training set. On the other hand, oversampling involves creating artificial cases and attaching them to the training set for the minority class. These techniques can also be further divided into heuristic and random methods. The former involves selecting cases randomly, while the latter considers the distribution of cases in the specified unbalanced learning problem. The effectiveness of oversampling and under sampling has been shown to depend on the specific problem being addressed [16]. Comparing these two approaches, it's evident that under sampling may omit valuable information from the learning process, potentially impacting classifier performance when dealing with small data files [17]. In contrast, oversampling generates artificial cases that contribute additional information. Informed oversampling also helps mitigate the risk of overfitting compared to random oversampling [18].

To address class imbalance, [19] introduced a two-stage strategy that combines the DBSCAN clustering algorithm with a graph-based procedure to filter noisy majority class instances. Random under sampling, a non-heuristic method, evens out the class distribution by randomly removing some majority class instances. However, this approach has the drawback that valuable information for training might be left out due to the absence of selection criteria for these examples. Several informative under-sampling techniques have been proposed, including the Neighborhood Cleaning Rule [20], Tomek Links [21], the Condensed Nearest Neighbor rule [22], and One-Sided Selection [23].A study found CNN and Tomek Links help to remove noisy instances by eliminating misclassified observations and improving class separation. CNN also targets majority class instances far from the decision boundary, while One-Sided Selection combines CNN with Tomek Links[24].

Some recent research on over-sampling has been introduced. The research introduced a similarity-based technique that enhances multi-label text classification by utilizing label correlations [25]. Another article combines the boosting algorithms with methods such as SMOTE that show better predictions for minority classes in imbalanced datasets [26]. The approach called BM-WGAN employs bootstrap sampling and Wasserstein GANs to generate high-quality synthetic data, addressing challenges in GAN training [27]. Improved GANs better align generated samples with the minority class distribution, which enhances both fidelity and diversity [25,27,28]. Additionally, a reevaluation of Random Oversampling underscores its simplicity and competitiveness in certain scenarios [29]. SMOTE is a more informative approach that addresses this issue. It generates synthetic examples for the minority class by using a minority class instance and creating new data points along the line connecting its k closest neighbors. Nevertheless, SMOTE has the drawback that it may cause the majority and minority class clusters to overlap, particularly in certain areas of the input space where they are not well separated [18]. Strategies to address this situation have been proposed, such as selecting appropriate original minority classes in SMOTE [30]. An experimental analysis of SMOTE can be found in [31]. Other research on SMOTE includes the combination of SMOTE with Edited Nearest Neighbor [32], which eliminates incorrectly classified instances based on the designation by its three nearest neighbors. Safe-Level SMOTE [33] modifies the SMOTE algorithm by adding a weight known as the safe level throughout the data generation process. ADASYN [34], Borderline-SMOTE [35], Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning [36], and its variant Kernel ADASYN [37] aim to select difficult-to-learn examples instead of generating samples using noisy data. Minority clustering SMOTE (MC-SMOTE) [38] involves clustering minority class samples to enhance the effectiveness of imbalance classification. Various methods addressing within-class imbalance problems have been proposed [33,39–43]. These methods rely on clustering to divide the input space before applying sampling techniques to adjust the size of clusters. Cluster-SMOTE [44] combines k-means and SMOTE, while DBSMOTE uses DB-SCAN to find clusters of arbitrary shapes. A-SUWO [41] is a complex method that generates synthetic instances based on a specified weighting scheme and clusters instances of the minority class. Other ensemble-based algorithms include SMOTEBoost [4] and DataBoost-IM. Finally, group-based methods have also been proposed [45].

2.1. Deep Neural Network

This section provides an in-depth exploration of deep learning neural networks, a pivotal subset of artificial neural networks inspired by the intricacies of the human brain [46]. The architecture comprises interconnected layers of artificial neurons, denoted by h hidden layers, an input layer (x), and an output layer (y) [47]. The forward pass is expressed as:

$$y = f(W_{ho}.f(W_{h2}.f(...f(W_{h1.x} + b_{h1}) + ...) + b_{h2}) + b_{ho}) \tag{1}$$

where W represents weights, f is the activation function (e.g., ReLU), and b denotes biases. The "deep" nature arises from the depth (h) of these hidden layers, allowing the network to automatically acquire hierarchical data representations [46]. The training process involves minimizing a loss function L by adjusting the weights through backpropagation and gradient descent:

$$\frac{\partial L}{\partial W} = -\eta.\frac{\partial L}{\partial y}.\frac{\partial y}{\partial W} \tag{2}$$

Subsequently, a comprehensive overview of applications is provided, showcasing the versatility of deep learning neural networks across diverse domains. Specific instances include Convolutional Neural Networks (CNNs) for image recognition and the utilization of Recurrent Neural Networks (RNNs) and transformer models in natural language processing tasks [46].

The challenges associated with training deep neural networks are discussed, encompassing considerations such as computational demands and data requirements [48]. These insights provide a nuanced understanding of potential limitations and avenues for improvement.

Furthermore, recent advancements in the field are explored, highlighting novel approaches in neural network architectures [49] and the growing significance of transfer learning [50]. These developments not only address existing challenges but also open new avenues for research and application. As the field of deep learning continues to evolve, ongoing efforts are directed towards enhancing model interpretability, scalability, and adaptability to diverse datasets. The dynamic nature of deep learning research underscores the need for continued exploration and innovation in the pursuit of more efficient and effective neural network models.

2.2. Data over Sampling Techniques

2.2.1. Smote

The SMOTE algorithm helps balance a training set by generating synthetic examples of the minority class. It accomplishes this by finding the nearest neighbors of a randomly selected minority class instance and creating new data points between them. This process results in a balanced class distribution.

2.2.2. Random over Sampler (ROS)

The Random Over Sampler (ROS) addresses imbalanced datasets by randomly duplicating instances from the minority class until both classes have the same number of instances. While simple, this technique may lead to overfitting.

2.2.3. K-Means SMOTE (KMS)

This approach utilizes both SMOTE oversampling and k-means clustering to balance skewed datasets. It involves clustering, filtering, and oversampling, preventing overfitting by generating synthetic samples and strategically targeting areas for oversampling.

2.2.4. Affinity Propagation and Random over Sampler-Based Oversampling (APROSO)

Affinity Propagation is a clustering technique that doesn't require a predetermined number of clusters. It uses data point similarity matrices to find exemplars, which are then used in Random Over-sampling Synthetic (ROS) to create synthetic samples that balance minority datasets.

2.2.5. Self-Organizing Map Oversampling (SOMO)

SOMO uses Self-Organizing Maps (SOM) to cluster and balance skewed datasets. It involves three steps: running the dataset through SOM to cluster information, generating fictitious data points for the minority class, and using these artificial examples to modify the distribution based on cluster density.

2.3. Data Under Sampling Techniques

2.3.1. Cluster Centroids (CCs)

CCs generate synthetic samples by replacing majority class samples with a cluster's centroid. The centroid is computed by averaging the feature values of all samples in the cluster, resulting in a balanced dataset.

2.3.2. Near Miss 1 (NM1)

The Near Miss algorithm is a controlled under-sampling technique that selects majority class samples based on their average distance to the minority class samples. NearMiss-1 chooses positive samples with the shortest average distance to the k-closest negative class samples.

2.3.3. Random Under Sampler (RUS)

According to our sampling strategy, the Random Under Sampler deletes the rows of the majority class(es) at random. By randomly choosing a subset of data for the specified classes, the Random Under Sampler is a quick and simple technique to balance the data.

2.3.4. Neighborhood Cleaning Rule (NCR)

In order to balance a dataset by using data reduction, the NCR under-sampling technique. Its key benefit is that it considers data quality, putting a stronger emphasis on data cleansing than reduction.

The edited nearest neighbor (ENN) algorithm, a developed instance reduction algorithm, is used to eliminate noisy majority instances. It is employed to remove those occurrences whose classes deviate by at least two standard deviations from those of their three closest neighbors.
Based on the literature, oversampling and undersampling methods have various advantages and disadvantages. These are summarized below:

2.4. Pros and Cons of oversampling methods

Table 1. Pros and Cons of Oversampling Methods.

Method	Pros	Cons
SMOTE	SMOTE (Synthetic Minority Over-sampling Technique) is a valuable tool for addressing class imbalance in machine learning. It improves model performance [16] and preserves data patterns, reducing overfitting to the majority class [35].	SMOTE introduces complexity, potentially lengthening training times, and risks overfitting to the minority class [16]. The choice of the k-value can impact results, and SMOTE may generate noisy synthetic samples in certain scenarios [34]. Careful parameter tuning and consideration of dataset characteristics are essential [33].
ROS	Random Over-sampling is a straightforward technique used to balance imbalanced datasets. Its primary advantage is that it effectively increases the representation of the minority class, which can lead to improved model performance [51,52].	Despite its simplicity, Random Over-sampling has certain drawbacks. It can lead to overfitting, especially when applied excessively [16]. Additionally, it doesn't introduce new information into the dataset, potentially resulting in a loss of variety in the minority class [53]. Careful consideration of these limitations and experimentation is necessary when employing this technique.
K-means SMOTE	K-Means SMOTE is an extension of the SMOTE algorithm that uses clustering to generate synthetic samples. Its advantages include improved handling of class imbalance by creating diverse synthetic samples and its potential to mitigate overfitting [33].	However, K-Means SMOTE has its limitations. It introduces complexity, which can increase training times, and it may require careful parameter tuning [33]. Additionally, like SMOTE, it can generate noisy synthetic samples, especially in cases of high class overlap [34].
APROSO	Affinity propagation is a method of identifying clusters and selecting high-quality exemplars through communication between data points [54]. When it comes to large scale datasets, Affinity Propagation is an effective method to use [55].	Affinity propagation algorithm is not efficient for solving high-dimensional problems [56].
SOMO	SOMO effectively identifies oversampling areas and helps to avoid noisy examples [57]. SOM provides data distribution visualization [58].	The self-organizing map is not capable of detecting small shifts in individual pixels [59].

doi:10.20944/preprints202412.2343.v1

2.5. Pros and cons of Undersampling Methods

Table 2. Pros and Cons of Under Sampling Techniques Methods.

Method	Pros	Cons
Cluster Centroids (CCs)	CCs reduce computational cost, preserve class distribution, and improve model performance [16,23,53].	Risk of information loss, sensitivity to clustering algorithm, and challenges with high-dimensional data [23,53].
Random Under Sampler (RUS)	RUS speeds up training time, prevents overfitting, and enhances recall of the minority class [15,16,51].	RUS may lead to information loss, bias towards the majority class, and reduced diversity for large datasets [15,16,51].
Neighborhood Cleaning Rules (NCL)	NCL preserves minority class information, improves generalization, and works with noisy data [20,32].	Requires tuning hyperparameters, can be a time-consuming process, and may eliminate relevant majority class instances [20].
Near Miss 1 (NM1)	NM1 maintains the minority class, decreases computational costs, and works with high-dimensional data [18,20].	Possible information loss, sensitivity to minority class samples, and potential for overfitting [18,20].

3. Methodology

The goal of this study is to evaluate the effectiveness of five distinct oversampling techniques and four different undersampling methods. Various metrics, including F-Score, AUC, and G-Score, were utilized to assess the performance of these techniques. Additionally, two different DNN models were employed to evaluate the techniques on the dataset.

3.1. Dataset

The study used 870 photographs of plant species from Jouko Lehmuskallio's collection, classified into two plant classes: Pinopsida and Magnoliopsida. The class distribution was imbalanced, and augmentation techniques were used to balance the data. In total, 1037 data samples were used for the study.

3.2. Samples And Labels

When training a neural network for supervised learning, the first requirement is a dataset containing samples and their corresponding labels. The term "samples" refers to the data points within the dataset, while the labels are the associated tags for each sample. For instance, in a sentiment analysis project using headlines from a news source, the labels could be "positive" or "negative" for each headline. Similarly, in our model trained on images of Magnoliopsida and Pinopsida, the labels for the images are "Magnoliopsida" or "Pinopsida".

3.3. Expected Data Format

In a study when we are preparing data, it's important to understand the format required for end goal. In our case, we want the data to be compatible with a neural network model. The upcoming model will be a Sequential model from the Keras API integrated within TensorFlow. We need to understand the type of data expected by a Sequential model. During training, the Sequential model receives data when we call the fit() function. Therefore, we need to ensure that our input data x and corresponding label data y are in the expected format. In addition to formatting the data for the model, another reason to process the data is to make it easier, faster, or more efficient for the network to learn from. This can be achieved through data normalization or standardization techniques.

3.4. Organize the Data

After obtaining the data we now need to organize the directory structure on disk to hold the data set. We'll manually do some parts of the organization and programmatically do the rest. We will create the directory for “Magnoliopsida” or “Pinopsida”. That’s it for the manual entry part. At this point, we have unlabeled Magnoliopsida 857 and Pinopsida 180 images. Now we transferred all Magnoliopsida 857 and Pinopsida 180 Pinopsida to a new folder and labeled them as Magnoliopsida and Pinopsida images. We can see that we have 2 problems that we need to address before we process our data:

- A small sample of Magnoliopsida images and an extremely small sample of Pinopsida images
- The extreme imbalance between the two classes (Magnoliopsida and Pinopsida)

3.5. Data Conversion and Balancing

In this phase we have write functions to preprocess images, making them suitable for our machine learning tasks. Initially, we access our image file and the image undertakes conversion to grayscale applying the grayscale() function. Then we resized to a standard dimension of 28x28 pixels via the resize() method. After, the image is flattened into a one-dimensional array using np.ravel() from the NumPy library. Finally, the pixel values are normalized to a range between 0 and 1 by dividing each pixel value by 255.0. Next, we create a data frame of the data using pandas, and rename the class column “Outcome”. Since, the “Outcome” column is the last column in our dataset we want to move it to be the first column. Since, our data is not shuffle, that is, our dataframe is built from 180 Pinopsida rows and after it 857 Magnoliopsida rows, we want to shuffle our data. The data set consists of 1037 data points, with 785 features each. “Outcome” is the feature we are going to predict, Magnoliopsida means Magnoliopsida images, Pinopsida means Pinopsida images. Of these 1037 data points, 857 are labeled as Magnoliopsida and 180 as Pinopsida. Then we create our X matrix and Y vector. At this point we have been applied oversampling and under sampling for balancing the imbalance data. This research we have been applied five over sampling and four under sampling techniques. After applying those techniques, we have sent those to our DNN1 and DNN2.

3.6. Metrics

F-Score, G-Score, and AUC were used to evaluate the predictive power of machine learning classifiers on the balanced dataset. A confusion matrix was used to assess and visualize classifier performance.

Table 3. Confusion Matrix.

Confusion Matrix	Observed Positive	Observed Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$G - Score = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}$$

3.7. Experimental Procedure

The experimental procedure involved splitting the raw dataset into training and test sets. Oversampling and undersampling techniques were applied to the training set. Classification was performed, and the obtained outputs were evaluated using various parameters, including F-Score, G-Score, and AUC.

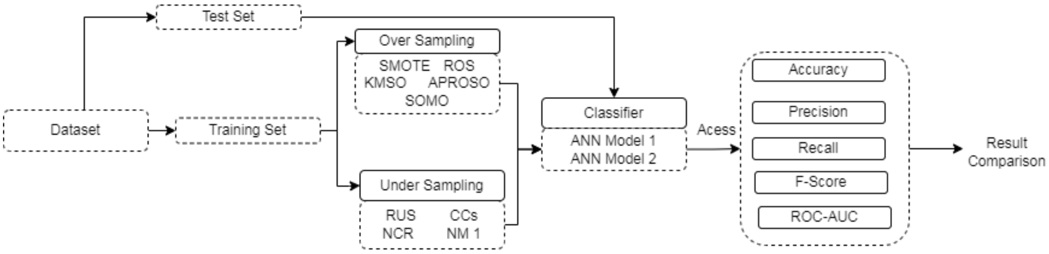


Figure 2. Experimental procedure of our implemented system

Materials and Methods should be described with sufficient details to allow others to replicate and build on published results. Please note that publication of your manuscript implicates that you must make all materials, data, computer code, and protocols associated with the publication available to readers. Please disclose at the submission stage any restrictions on the availability of materials or information. New methods and protocols should be described in detail while well-established methods can be briefly described and appropriately cited.

Research manuscripts reporting large datasets that are deposited in a publicly avail-able database should specify where the data have been deposited and provide the relevant accession numbers. If the accession numbers have not yet been obtained at the time of submission, please state that they will be provided during review. They must be provided prior to publication.

Interventionary studies involving animals or humans, and other studies require ethical approval must list the authority that provided approval and the corresponding ethical approval code.

4. Result Analysis

4.1. F1 Score for Oversampling Methods

The following table displays the outcomes of different oversampling techniques in relation to two Artificial Neural Network (DNN) models, namely "DNN Model 1" and "DNN Model 2." It is crucial to note that APROSO persistently demonstrates remarkable performance, obtaining F1 scores of 0.92 in Model 1 and 0.93 in Model 2, which makes it the most effective method among the available options. ROS and KMS also provide competitive outcomes, with F1 scores ranging from 0.9 to 0.92 across both models. On the other hand, SMOTE and SOMO generally lead to slightly lower F1 scores, varying from 0.87 to 0.91. These variations in the results emphasize the significance of selecting the most efficient oversampling technique, wherein APROSO proves to be particularly effective in addressing class imbalance in binary classification tasks.

Table 4. F1 Score for Oversampling Methods.

Oversampling Method	Class	F1 Score
SMOTE	0	0.9
	1	0.9
ROS	0	0.9
	1	0.9
KMS	0	0.9
	1	0.89
APROSO	0	0.92
	1	0.91
SOMO	0	0.88
	1	0.87

4.2. F1 Score for Undersampling Methods

The impact of various under-sampling methods on two DNN models for binary classification is presented in Table III. Notably, Random Under Sampler improves class 1 F1 scores significantly (Model

1: 0.76, Model 2: 0.84) while maintaining class 0 scores (Model 1: 0.77, Model 2: 0.81). Cluster Centroids consistently boosts class 1 F1 scores (Model 1: 0.77, Model 2: 0.85) with slight reductions in class 0 (Model 1: 0.76, Model 2: 0.87). The Neighborhood Cleaning Rule enhances class 0 F1 scores (Model 1: 0.79, Model 2: 0.86) with some trade-offs in class 1 (Model 1: 0.67, Model 2: 0.79). Overall, Cluster Centroids appears to be the preferred choice, significantly improving minority class performance while minimally affecting the majority class.

Table 5. F1 Score for Undersampling Methods.

Undersampling Method	Class	F1 Score
Random Under Sampler	0	0.77
	1	0.76
Cluster Centroids	0	0.76
	1	0.77
Neighborhood Cleaning Rule	0	0.79
	1	0.67
Near Miss	0	0.73
	1	0.75

4.3. AUC Comparison for Oversampling

Table 6. AUC Comparison for Oversampling Techniques.

Oversampling Technique	DNN1	DNN2
SMOTE	0.9563	0.9722
ROS	0.9488	0.971
KMS	0.9551	0.967
APROSO	0.9741	0.9773
SOMO	0.9331	0.9747

DNN1 and DNN2 models are two different models applied on the AUC values of different oversampling techniques. Regardless of everything, the studies presented in Table 4 suggest that the Affinity Propagation and Random Over Sampler-Based Oversampling (APROSO) technique produce the highest AUC values, as the AUC values are 0.9741 and 0.9773 for DNN1 and DNN2, respectively. SMOTE also performs reasonably well, generating values of 0.9563 and 0.9722 for DNN1 and DNN2, respectively, but Random Over Sampling (ROS) and Self-Organizing Map Oversampling (SOMO) fall short of their performance. This result underscores the importance of selecting an appropriate oversampling method, and using APROSO may improve the model fitting of the two DNN models.

4.4. AUC Comparison for Undersampling

Table 7. AUC Comparison for Undersampling Methods.

Undersampling Method	DNN1	DNN2
Random Under Sampler	0.7794	0.8871
Cluster Centroids	0.8446	0.9463
Neighborhood Cleaning Rule	0.8257	0.9134
Near Miss	0.8127	0.8729

In Table 5, the results of AUC values from employing different under-sampling methods with two different deep neural network models are contrasted, and it is concluded that Cluster Centroids was the standout method. Indeed, the AUC values predicted 0.8446 for DNN1 and 0.9463 for DNN2. Next, the Neighborhood Cleaning Rule also performed well, garnering AUC values of 0.8257 and 0.9134 for DNN1 and DNN2, respectively, while Random Under Sampler was next with AUC values of 0.7794 and 0.8871 for DNN1 and DNN2. Furthermore, Near Miss also demonstrated good performance, in that it obtained 0.8127 and 0.8729 for DNN1 and DNN2. These results underline the necessity of an appropriate selection of the under-sampling technique, as well as present the Cluster Centroids method as the option that was able to particularly improve the overall AUC for both of the DNN models.

4.5. Oversampling Method G-Score Comparison

Table 8. Oversampling Method G-Score Comparison.

Oversampling Method	DNN1	DNN2
SMOTE	0.895794	0.911724
ROS	0.897774	0.919521
KMS	0.897477	0.91594
APROSO	0.915892	0.925475
SOMO	0.88259	0.925475

In Table 6, we can see a comparison of G-Score values achieved by different oversampling methods on two different DNN models, namely DNN1 and DNN2. Out of all the methods, APROSO stands out as the most effective technique, giving G-Scores of approximately 0.915892 for DNN1 and around 0.925475 for DNN2. ROS (Random Over Sampling) and KMS (K-Means SMOTE) also work well, with G-Scores close to APROSO, which are roughly 0.897774 and 0.897477 for DNN1 and approximately 0.919521 and 0.91594 for DNN2, respectively. SMOTE gives G-Scores of around 0.895794 for DNN1 and roughly 0.911724 for DNN2, while SOMO (Self-Organizing Map Oversampling) produces G-Scores of around 0.88259 for DNN1 and approximately 0.925475 for DNN2. These results highlight the importance of selecting the right oversampling method, with APROSO and ROS showing great potential in improving G-Scores for both DNN models.

4.6. Undersampling Methods G-Score Comparison

Table 9. Undersampling Methods G-Score Comparison.

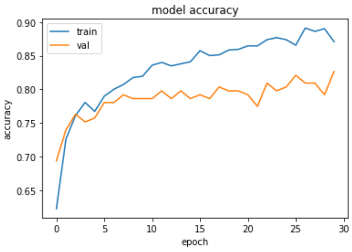
Undersampling Method	DNN1	DNN2
Random Under Sampler	0.763864	0.835467
Cluster Centroids	0.764929	0.863856
Neighborhood Cleaning Rule	0.747892	0.838093
Near Miss	0.742217	0.799503

It has been found that Cluster Centroids and Random Under Sampler are particularly effective in enhancing the G-Scores of DNN1 and DNN2, with scores of approximately 0.764929 and 0.863856 for Cluster Centroids, and approximately 0.763864 and 0.835467 for Random Under Sampler. While Neighborhood Cleaning Rule and Near Miss achieve slightly lower G-Scores, with scores of approximately 0.747892 and 0.742217 for DNN1, and approximately 0.838093 and 0.799503 for DNN2, respectively. It is crucial to select the appropriate under-sampling method, and the results suggest that Cluster Centroids and Random Under Sampler are the most effective methods.

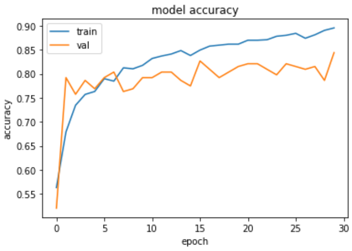
4.7. Oversampling Model Accuracy Comparison

Table 10. DNN model 1 represents accuracy comparison of five different over-sampling augmentation techniques (a) SMOTE (b) ROS (c) KMSO (d) APROSO (e) SOMO.

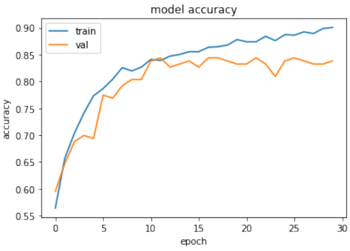
(A) SMOTE



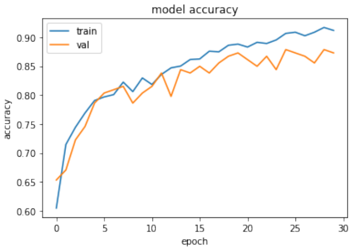
(B) ROS



(C) KMS



(D) APROSO



(E) SOMO

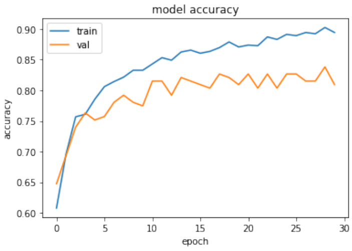
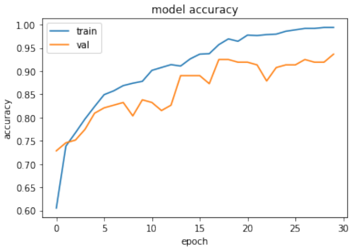


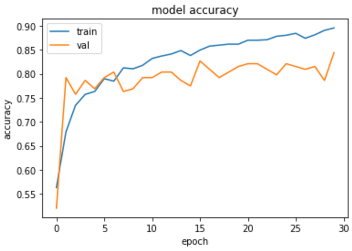
Table 8 shows accuracy comparisons of DNN Model 1. B exhibits overfitting at the beginning, but A, C, D, and F show underfitting. It seems C and D perform best fitting among all oversampling techniques.

Table 11. DNN model 2 represents accuracy comparison of five different over-sampling augmentation techniques (a) SMOTE (b) ROS (c) KMSO (d) APROSO (e) SOMO.

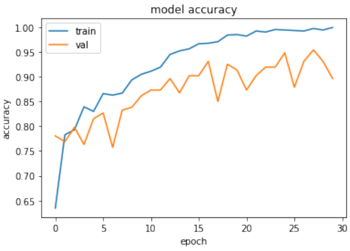
(A) SMOTE



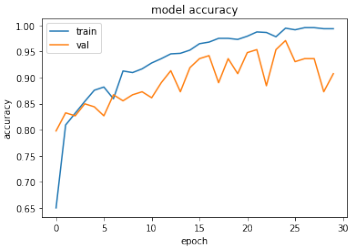
(B) ROS



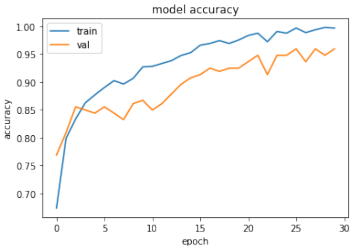
(C) KMS



(D) APROSO



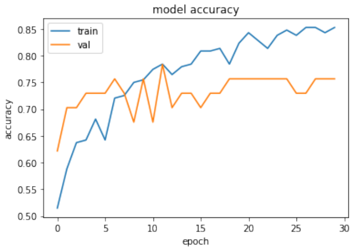
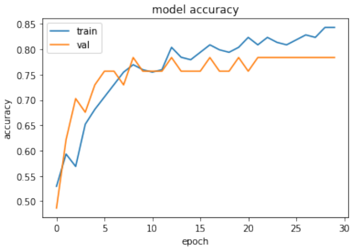
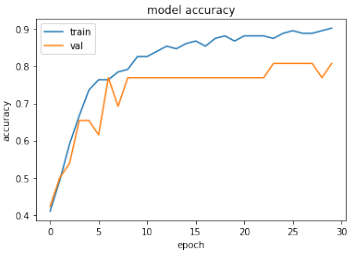
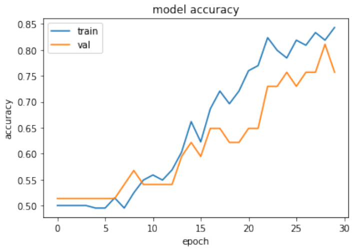
(E) SOMO



In Table 9, we have compared the accuracy of our applied DNN Model 2 with different sampling techniques. We noticed that random oversampling caused overfitting initially, while other sampling methods led to underfitting. However, after considering the performance of all the sampling techniques, we found that SOMO was the most effective for the DNN Model 2.

4.8. Undersampling Model Accuracy Comparison

Table 12. DNN Model 1 represents accuracy comparison of four different under-sampling augmentation techniques (a) Random Under Sampler (b) Cluster Centroids (c) Neighborhood Cleaning Rule (d) Near Miss.

(A) Random Under Sampler	**(B) Cluster Centroids**	**(C) Neighborhood Cleaning Rule**
		
(D) Near Miss		
		

In Table 10, there are four models labeled A, B, C, and D. Among these models, A, C, and D show overfitting, where the validation accuracy is higher than the training accuracy. These three models also demonstrate noisy fittings. However, in the case of model B, the noisy fitting is relatively less than the other three models, and the training accuracy and validation accuracy are equal, indicating perfect fitting. Therefore, we can conclude that among the four models, model B is the best for ANN model 1 when using under-sampling augmentation techniques.

Table 13. DNN Model 2 represents accuracy comparison of four different under-sampling augmentation techniques: (a) Random Under Sampler (b) Cluster Centroids (c) Neighborhood Cleaning Rule (d) Near Miss.

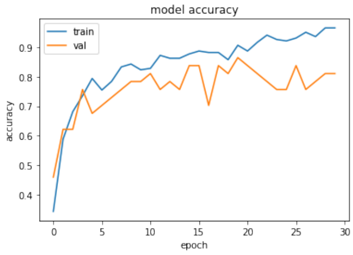
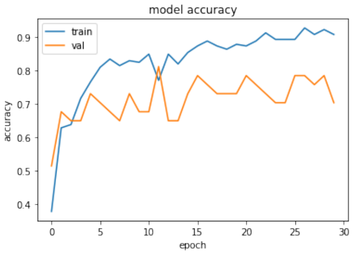
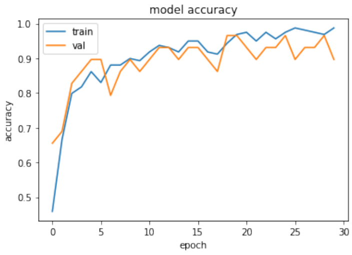
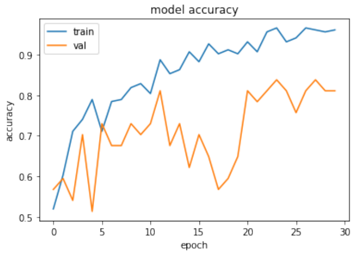
(A) Random Under Sampler	(B) Cluster Centroids	(C) Neighborhood Cleaning Rule
		
(D) Near Miss		
		

Table 11 shows that out of the four models mentioned, more noisy fittings were observed. However, the c model was the only one that displayed less noisy fitting. Despite that, training accuracy and validation accuracy were equal in this case, which is signifying of perfect fitting. Therefore, it can be concluded that among the four models mentioned, the c model is the best option when using under-sampling augmentation techniques in DNN model 2.

4.9. Discussions

Overall in the analysis of oversampling methods, APROSO showed a robust performance across the board. More concretely, If we put it differently, it reached an F1 score, a ROC, and a G-Score of 0.92, 0.94, and 0.9159 respectively for model-1 and 0.93, 0.9741, 0.9255 respectively for model-2. From these performances we concluded APROSO outperformed the applied oversampling methods. Most of their metrics like F1 (the scores range from 0.9 to 0.92) and AUC are high, so they are good candidates within oversampling methods, i.e., for ROS and KMS. However, SMOTE and SOMO were less effective with respect to F1 scores, AUC values, and G-Scores, when compared to APROSO, KMS, and ROS.

While among the undersampling methods, Cluster Centroids has been consistently shown to be the best. The intermediate F1 scores and G-Scores of approx 0.764929 (DNN1) and 0.863856 (DNN2) show that the proposed Paraphrase Enhanced DNN is able to improve the overall performance. While the Neighborhood Cleaning Rule and Random Under Sampler did a little better (in the case of F1 and AUC scores), they were largely outperformed by the oversampling techniques.

Based on the results of the data provided, oversampling methods like APROSO, ROS, and KMS have been shown to give better performance by obtaining higher F1 scores, AUC values, and G-Scores as compared to undersampling methods. In this way, oversampling provides the method to create fake data and enhance the accuracy of the model to classify well minority classes. However one should know that the answer may vary depending on the dataset, and we must ensure that we are not overfitting.

5. Conclusion

In this paper, we presented five oversampling methods, namely Synthetic Minority Oversampling Technique (SMOTE), Random Over Sampling (ROS), K-Means and Smote-Based Oversampling

(KMSO), Affinity Propagation and Random Over Sampling-Based Oversampling (APROSO), and Self-Organizing Map-based Oversampling (SOMO). We also presented four undersampling methods, namely Random Under Sampling (RUS), Cluster Centroids (CCs), Neighborhood Cleaning Rule (NCR), and Near Miss-1 (NM1).

The performance of these selected oversampling and undersampling techniques was evaluated on our bio-color image dataset. The results showed that for DNN model 1 and 2, these sampling techniques' performances varied. For model 2, these sampling techniques performed better all the time. The better performance is ensured using different evaluation parameters, i.e., precision, recall, F1-score.

From the investigation, we reveal that oversampling techniques perform better most of the time than the undersampling techniques. These oversampling techniques can be helpful for researchers and practitioners since they result in the generation of high-quality artificial data and only require tuning a small number of parameters. **Author Contributions:** The authors contributed equally to this work.

Funding: This paper is supported by the Strategic Research Council at the Research Council of Finland under Grant 352460.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study used 870 photographs of plant species from Jouko Lehmuskallio's collection, classified into two plant classes.

Conflicts of Interest: Not Applicable.

References

1. Hassaan, M.A.; El Nemr, A.; Hassaan, A. Health and environmental impacts of dyes: mini review. *American Journal of Environmental Science and Engineering* **2017**, *1*, 64–67.
2. Gürses, A.; Açıkyıldız, M.; Güneş, K.; Gürses, M.S. Colorants in Health and Environmental Aspects. In *SpringerBriefs in Molecular Science*; Springer International Publishing, 2016; pp. 69–83. https://doi.org/10.1007/978-3-319-33892-7_5.
3. Yusuf, M.; Shabbir, M.; Mohammad, F. Natural Colorants: Historical, Processing and Sustainable Prospects. *Natural Products and Bioprospecting* **2017**, *7*, 123–145. https://doi.org/10.1007/s13659-017-0119-9.
4. Chawla, N.; Japkowicz, N.; Kolcz, A. Workshop learning from imbalanced data sets II. In Proceedings of the Proc. Int'l Conf. Machine Learning, 2003.
5. Sanz, J.; Sesma-Sara, M.; Bustince, H. A fuzzy association rule-based classifier for imbalanced classification problems. *Information Sciences* **2021**, *577*, 265–279. https://doi.org/10.1016/j.ins.2021.07.019.
6. Chawla, N.V.; Japkowicz, N.; Kotcz, A. Editorial. *ACM SIGKDD Explorations Newsletter* **2004**, *6*, 1–6. https://doi.org/10.1145/1007730.1007733.
7. Zhao, X.M.; Li, X.; Chen, L.; Aihara, K. Protein classification with imbalanced data. *Proteins: Structure, Function, and Bioinformatics* **2007**, *70*, 1125–1132. https://doi.org/10.1002/prot.21870.
8. Yu, L.; Zhou, R.; Tang, L.; Chen, R. A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data. *Applied Soft Computing* **2018**, *69*, 192–202. https://doi.org/10.1016/j.asoc.2018.04.049.
9. Dastile, X.; Celik, T.; Potsane, M. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing* **2020**, *91*, 106263. https://doi.org/10.1016/j.asoc.2020.106263.
10. Domingos, P. MetaCost. In Proceedings of the Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, aug 1999. https://doi.org/10.1145/312129.312220.
11. Ting, K.M. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering* **2002**, *14*, 659–665.
12. Clearwater, S.; Stern, E. A rule-learning program in high energy physics event classification. *Computer Physics Communications* **1991**, *67*, 159–182. https://doi.org/10.1016/0010-4655(91)90014-c.
13. Prati, R.C.; Batista, G.E.A.P.A.; Silva, D.F. Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems* **2014**, *45*, 247–270. https://doi.org/10.1007/s10115-014-0794-3.

14. Fernández, A.; López, V.; Galar, M.; del Jesus, M.J.; Herrera, F. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems* **2013**, *42*, 97–110. https://doi.org/10.1016/j.knosys.2013.01.018.
15. 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 Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **2011**, *42*, 463–484.
16. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **2002**, *16*, 321–357. https://doi.org/10.1613/jair.953.
17. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). Ieee, 2008, pp. 1322–1328.
18. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* **2009**, *21*, 1263–1284.
19. Guzmán-Ponce, A.; Sánchez, J.; Valdovinos, R.; Marcial-Romero, J. DBIG-US: A two-stage under-sampling algorithm to face the class imbalance problem. *Expert Systems with Applications* **2021**, *168*, 114301. https://doi.org/10.1016/j.eswa.2020.114301.
20. Laurikkala, J. Improving Identification of Difficult Small Classes by Balancing Class Distribution. In *Artificial Intelligence in Medicine*; Springer Berlin Heidelberg, 2001; pp. 63–66. https://doi.org/10.1007/3-540-48229-6_9.
21. Tomek, I. Two modifications of CNN. **1976**.
22. Hart, P. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory* **1968**, *14*, 515–516.
23. Kubat, M.; Matwin, S.; et al. Addressing the curse of imbalanced training sets: one-sided selection. In Proceedings of the Icml. Citeseer, 1997, Vol. 97, p. 179.
24. Wang, H.; Liu, X. Undersampling bankruptcy prediction: Taiwan bankruptcy data. *PLOS ONE* **2021**, *16*, e0254030. https://doi.org/10.1371/journal.pone.0254030.
25. Hakki Karaman, I.; Koksal, G.; Eriskin, L.; Salihoglu, S. A Similarity-Based Oversampling Method for Multi-label Imbalanced Text Data. *arXiv e-prints* **2024**, pp. arXiv–2411.
26. Wongvorachan, T.; He, S.; Bulut, O. A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. *Information* **2023**, *14*, 54.
27. Hou, B.; Chen, G. A new imbalanced data oversampling method based on Bootstrap method and Wasserstein Generative Adversarial Network. *Mathematical Biosciences and Engineering* **2024**, *21*, 4309–4327.
28. Pan, T.; Pedrycz, W.; Yang, J.; Wang, J. An improved generative adversarial network to oversample imbalanced datasets. *Engineering Applications of Artificial Intelligence* **2024**, *132*, 107934.
29. Kamalov, F.; Leung, H.H.; Cherukuri, A.K. Keep it simple: random oversampling for imbalanced data. In Proceedings of the 2023 Advances in Science and Engineering Technology International Conferences (ASET). IEEE, 2023, pp. 1–4.
30. Shen, F.; Zhao, X.; Kou, G.; Alsaadi, F.E. A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing* **2021**, *98*, 106852. https://doi.org/10.1016/j.asoc.2020.106852.
31. Elreedy, D.; Atiya, A.F. A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. *Information Sciences* **2019**, *505*, 32–64. https://doi.org/10.1016/j.ins.2019.07.070.
32. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter* **2004**, *6*, 20–29. https://doi.org/10.1145/1007730.1007735.
33. Bunkhumpornpat, C.; Sinapiromsaran, K.; Lursinsap, C. Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. In *Advances in Knowledge Discovery and Data Mining*; Springer Berlin Heidelberg, 2009; pp. 475–482. https://doi.org/10.1007/978-3-642-01307-2_43.
34. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). Ieee, 2008, pp. 1322–1328.
35. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Lecture Notes in Computer Science*; Springer Berlin Heidelberg, 2005; pp. 878–887. https://doi.org/10.1007/11538059_91.

36. Barua, S.; Islam, M.M.; Yao, X.; Murase, K. MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on knowledge and data engineering* **2012**, *26*, 405–425.
37. Tang, B.; He, H. KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning. In Proceedings of the 2015 IEEE congress on evolutionary computation (CEC). IEEE, 2015, pp. 664–671.
38. Yi, H.; Jiang, Q.; Yan, X.; Wang, B. Imbalanced classification based on minority clustering synthetic minority oversampling technique with wind turbine fault detection application. *IEEE Transactions on Industrial Informatics* **2020**, *17*, 5867–5875.
39. Elyan, E.; Moreno-Garcia, C.F.; Jayne, C. CDSMOTE: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification. *Neural Computing and Applications* **2020**, *33*, 2839–2851. https://doi.org/10.1007/s00521-020-05130-z.
40. Vo, M.T.; Nguyen, T.; Vo, H.A.; Le, T. Noise-adaptive synthetic oversampling technique. *Applied Intelligence* **2021**, *51*, 7827–7836. https://doi.org/10.1007/s10489-021-02341-2.
41. Nekooeimehr, I.; Lai-Yuen, S.K. Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. *Expert Systems with Applications* **2016**, *46*, 405–416. https://doi.org/10.1016/j.eswa.2015.10.031.
42. Cieslak, D.A.; Chawla, N.V. Start globally, optimize locally, predict globally: Improving performance on imbalanced data. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. IEEE, 2008, pp. 143–152.
43. Jo, T.; Japkowicz, N. Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter* **2004**, *6*, 40–49. https://doi.org/10.1145/1007730.1007737.
44. Cieslak, D.A.; Chawla, N.V.; Striegel, A. Combating imbalance in network intrusion datasets. In Proceedings of the GrC, 2006, pp. 732–737.
45. Sun, Z.; Song, Q.; Zhu, X.; Sun, H.; Xu, B.; Zhou, Y. A novel ensemble method for classifying imbalanced data. *Pattern Recognition* **2015**, *48*, 1623–1637. https://doi.org/10.1016/j.patcog.2014.11.014.
46. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016. http://www.deeplearningbook.org.
47. Bishop, C. Pattern recognition and machine learning. *Springer google schola* **2006**, *2*, 5–43.
48. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436–444.
49. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
50. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* **2009**, *22*, 1345–1359.
51. Japkowicz, N.; Stephen, S. The class imbalance problem: A systematic study. *Intelligent data analysis* **2002**, *6*, 429–449.
52. Weiss, G.M.; McCarthy, K.; Zabar, B. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *Dmin* **2007**, *7*, 24.
53. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence* **2009**, *23*, 687–719.
54. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *science* **2007**, *315*, 972–976.
55. Liu, X.; Yin, M.; Li, M.; Yao, D.; Chen, W. Hierarchical affinity propagation clustering for large-scale data set. *Computer Science* **2014**, *41*, 185–188.
56. Wang, Z.J.; Zhan, Z.H.; Lin, Y.; Yu, W.J.; Yuan, H.Q.; Gu, T.L.; Kwong, S.; Zhang, J. Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems. *IEEE Transactions on Evolutionary Computation* **2017**, *22*, 894–908.
57. Douzas, G.; Rauch, R.; Bacao, F. G-SOMO: An oversampling approach based on self-organized maps and geometric SMOTE. *Expert Systems with Applications* **2021**, *183*, 115230.
58. Douzas, G.; Bacao, F. Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning. *Expert systems with Applications* **2017**, *82*, 40–52.
59. Hua, W.; Mo, L. Clustering ensemble model based on self-organizing map network. *Computational Intelligence and Neuroscience* **2020**, *2020*.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.