www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 420-440. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i3.846

Empirical study of editing sampling on deep learning hidden layers space to classify imbalanced hyperspectral remote sensing images

Daniel Cervantes Ambriz[1], Carlos Mauricio Castorena Lara[2], Roberto Alejo Eleuterio[1], Everardo Efrén Granda Gutiérrez[3], Federico del Razo López[1], Vicente García Jiménez[4].
[1] Division of Postgraduate Studies and Research, National Technological of Mexico, Campus Toluca, Metepec, 52149, Mexico, Mexico.
[2] Information Technology Department, University of Valencia, Spain.
[3] University Center at Atlacomulco, Autonomous University of the State of Mexico, Atlacomulco, 50450, Mexico, Mexico.
[4] Department of Electrical and Computer Engineering, Autonomous University of Ciudad Juarez, Ciudad Juarez, 32310, Chihuahua, Mexico
mm23281647@toluca.tecnm.mx, carlos.castorena@uv.es, ralejoe@toluca.tecnm.mx, eegrandag@uaemex.mx, fdelrazol@toluca.tecnm.mx, vicente.jimenez@uacj.mx

Abstract. This study explores an approach involving the adaptation of data sampling techniques within the hidden feature space of deep neural networks. By modifying traditional prototype selection and cleaning methods, our methodology eliminates noisy samples and condenses the data into representative points, thereby enhancing class separation and improving generalisation. A nearest-neighbours search in the hidden space enables more refined sample selection. Comprehensive experiments on four multi-class imbalanced hyperspectral datasets (Indian Pines, Salinas, PaviaU, and Pavia) demonstrated that combining over-sampling in the spectral space with editing in the hidden feature space outperforms conventional sampling methods. The best results were achieved with configurations such as ROS-TL-H2 and ROS-ENN-H3, which consistently yielded g-mean values above 0.90, showcasing the effectiveness of hidden-space editing. This strategy effectively balances class distributions while preserving informative samples, thereby improving classification performance and model robustness. Despite the increased computational complexity, the benefits justify its adoption in challenging scenarios involving class imbalance. The findings suggest that this approach may be particularly valuable for remote sensing and other highly imbalanced data classification tasks.
Keywords: Deep learning, hidden layer space, multi-class imbalance problem, sampling methods, hyperspectral remote sensing images.

Article Info
Received 31/01/2025
Accepted 04/04/2025

1 Introduction

The classification of hyperspectral remote sensing images has become essential for spatial data analysis tasks. Several machine learning methods, such as deep learning neural networks (DL-NN), have been developed as classification models using datasets where each pixel is represented as a spectral feature vector [1]. However, the successful construction of accurate DL-NN depends on the intrinsic data characteristics [2].

Semantically meaningful classes do not uniformly cover the Earth [3]; thus, the data obtained from remote sensing devices have land classes poorly represented by only a few pixels [4], [5]. This phenomenon is part of the class imbalance problem present in various datasets. For example, the BigEarthNet archive [6] has 19 classes, where 164,775 samples represent the Coniferous Forest class, and 194,148 samples are land class; in contrast, the Coastal wetlands and Beaches-dunes-sands classes only have 1,566 and 1,536 samples, respectively.

The class imbalance negatively impacts the performance of DL-NN models, ignoring the less represented classes, either partially or entirely [7], [8]. Consequently, it remains a challenge in hyperspectral remote sensing image classification [9], where most research has focused on extending traditional class imbalance approaches to DL-NN [10].

This issue has been addressed using three primary strategies [11]: 1) special-purpose learning approaches, 2) data sampling, and 3) hybrid methods. From the first category, the proposals have been developed for specific problems and classifiers adapting the learning algorithms by incorporating cost functions [4], [5], [12], [13]. Nevertheless, their extrapolation to other imbalances domains is restricted for various drawbacks, including non-straightforward adaptations of the learning algorithms, a retraining process if the loss function is changed, and a deep knowledge of the learning algorithm [11].

Data sampling algorithms have been widely used to address the class imbalance problem because they can be applied to any imbalance dataset problem and deep learning model [14], [15]. Thus, researchers are focused on balancing the classes by under-sampling (US), over-sampling (OS), or both (hybrid methods) [16].

The first approach to reduce the majority classes involved the random elimination of samples; however, it has been reported that this straightforward strategy can discard valuable samples. Therefore, an informed technique to reduce majority classes is based on prototype selection methods that eliminate noisy or redundant instances. Some examples include editing methods like the Editing Nearest Neighbor (ENN) [17], the Wilson editing [18], the Tomek Links (TL) [19], and the condensing methods, such as the Condensed Nearest Neighbor rule (CondNN) [20].

In contrast to random under-sampling (RUS), random over-sampling (ROS) replicates samples to upsize the minority classes. Regardless, this straightforward strategy results in overfitting the learning model [10] and increasing the training time [21]. From the intelligent solutions focused on generating better minority samples, the most influential pioneer is the Synthetic Minority Over-sampling technique (SMOTE) [22]. Its concept of generating artificial minority samples through interpolation has inspired the scientific community to create over 68 alternative versions [23], [24], [25], [26], [27].

Rendon et al. [16] show that hybrid methods can be improved if the selection strategy exploits the high-level features obtained from the output of a neural network (NN); that is, the nearest neighbor search is performed in this new output space rather than as it is usually done in the input feature space. Recognizing that the potential of deep learning models lies in the hidden layer [28], where the capacity to abstract and solve complex problems exists [29], and noting that each hidden node represents a dimension of the problem, this paper builds on Rendon's idea by utilizing the hidden output nodes to construct a hidden feature vector. This vector is subsequently employed to perform the nearest neighbor search, acting as the selection mechanism for two under-sampling algorithms: ENN and TL.

This study analyzes four multi-class imbalanced hyperspectral remote sensing image datasets. We employ two over-sampling methods (ROS and SMOTE) alongside three types of feature vectors (input, hidden, and output), which are utilized independently and in combination with two under-sampling methods (ENN and TL). The hidden and output vectors were derived from a Deep Learning Multilayer Perceptron (DL-MLP) model.

The main goal of this work is to evaluate the effectiveness of applying classical under and over-sampling techniques in the neural network hidden space and to improve their performance in the classification of multi-class and highly imbalanced hyperspectral datasets. The novelty of this perspective is that typical sampling approaches were designed to work in the feature space of the neural network. Thus, we take advantage of the potential of neural networks to transform feature space into hidden space, where the discrimination of class/region is more accessible than in the original space (the feature space). On the other hand, classifying multi-class and highly imbalanced problems is a significant challenge that remains in force in remote sensing classification tasks.

1.1 Related works

The class imbalance problem represents a significant challenge in hyperspectral image classification. Various researchers have proposed methodologies to address this limitation in recent decades, as shown in Table 1. Traditional sampling methods include the Synthetic Minority Over-sampling Technique (SMOTE), developed by Chawla et al. [22], who achieved F1-Score values between 0.70-0.85 by generating synthetic instances. Batista et al. [30] improved this approach by combining SMOTE with cleaning techniques like Tomek Links and ENN, reaching AUC (Area Under the receiver operating characteristic Curve) values of 0.75-0.90.

Buda et al. [2] analyzed the integration of these methods into deep architectures. Their research demonstrated that combinations of under-sampling and oversampling improve the performance of CNNs in datasets such as MNIST and ImageNet, reaching 86-92% accuracy. LeCun et al. [29] provided the essential theoretical foundations for deep learning architectures (DNN, CNN, RNN), establishing benchmark parameters with exceptionally high precision (95-99%) and minimal error rates (1-5%) across multiple domains, laying the conceptual groundwork for future developments in the field.

Specifically for hyperspectral images, Özdemir et al. [27] and Zhong et al. [26], implemented SMOTE-based strategies, obtaining G-mean (geometric mean) values of 0.85-0.89 and Kappa coefficients of 0.92-0.95. In parallel, Vuttipittayamongkol et al. [31] explored editing methods for noise reduction. Rendón et al. [16] proposed editing in the output space of multilayer networks, with G-mean values of 0.88-0.92. However, this approach does not leverage the intermediate representations generated in hidden layers.

Our work introduces "Deep Edition," applying editing techniques in the transformed space of hidden layers of DL-MLP networks. Experiments conducted on four hyperspectral datasets demonstrate that the ROS-TL-H2 and ROS-ENN-H3 configurations significantly outperform conventional methods, with G-mean values between 0.90-0.97, evidencing the effectiveness of this approach for problems with high spectral dimensionality and class imbalance.

Table 1. Related class imbalance problem studies.

Author	Reference	Model(s)	Data Source	Metrics & Values	Distinctive Characteristics
Buda et al., 2018	[2]	CNN	MNIST, CIFAR-10 and ImageNet	Accuracy: 86–92% ROC AUC: 0.88–0.94	Under-sampling and Oversampling methods.
Chawla et al., 2002	[22]	Bayesian, C4.5	UCI Repository	F1-Score: 0.70–0.85 AUC: 0.82–0.95	SMOTE
Batista et al., 2004	[30]	C4.5	UCI Repository	AUC: 0.75–0.90 F1-Score: 0.72–0.88	SMOTE + Tomek Links and SMOTE + ENN.
Rendón et al., 2020	[16]	Multilayer Perceptron (MLP)	Images, UCI Repository	G-mean: 0.88–0.92 Accuracy: 92–96%	Output-layer editing combined with hybrid methods of sampling.
Özdemir et al., 2021	[27]	CNN, DNN	Hyperspectral images	G-mean: 0.85–0.89 Kappa: 0.82–0.87	SMOTE
Zhong et al., 2021	[26]	CNN, Random Forest	Hyperspectral images	OA: 93–96% AA: 91–95% Kappa: 0.92–0.95	SMOTE
Vuttipittayamongkol et al., 2021	[31]	SVM, k-NN	UCI Repository	G-mean: 0.80–0.86 F1-Score: 0.78–0.85	Editing methods (ENN, TL) to reduce noise.
LeCun et al., 2015	[29]	DNN, CNN, RNN	Multiple datasets	Accuracy: 95–99% Error: 1–5%	Theoretical underpinnings methods.
Our work	--	DL-MLP (4 hidden layers)	Hyperspectral images (Indian Pines, Salinas, PaviaU, Pavia)	G-mean: 0.90–0.97 Rank: 4.25–7.25	Deep Edition in hidden layers (ROS-TL-H2, ROS-ENN-H3).

The paper is structured as follows. Section 2 describes the DL-MLP model, data sampling methods, and the proposed methodology. Section 3 details the experimental setup, while Sections 4 and 5 present and discuss the experimental results. Finally, Section 6 offers conclusions and future research directions.

2 Methods

This section summarizes the DL-MLP model used as a classifier, from which the hidden feature vectors were obtained from each hidden layer. It also includes a general overview of the data sampling methods and the deep edition process for dealing with the multi-class imbalance problem.

2.1 Deep Learning Multilayer Perceptron

A DL-MLP is a feed-forward neural network composed of multiple layers of interconnected neurons. The input layer receives the input data vector $x \in \mathbb{R}^{N}$, where N represents the dimensionality of the input feature space, specifically the number of features or input variables processed by the neural network. In the context of hyperspectral imagery analysis presented in this study, N corresponds to the number of spectral bands in the image. The hidden layers are located between the input and output layers, where feature space is transformed into hidden space, and the discrimination of class/region is more accessible than in the original space [28]. The output layer is represented by $z \in \mathbb{R}^{J}$, where J represents the dimensionality of the output space, specifically the number of classes or categories the neural network must classify. The neurons from the previous layer are fully connected using a synaptic weight w to the following layer, and all neurons in each layer employ an activation function $\varphi(\cdot)$, usually sigmoid. Logistic and hyperbolic tangent functions are commonly used for their corresponding sigmoid function form [32].

The input to a neuron i on the l-th layer (Eq. 1) is defined by the product of the results from the activation function $\varphi(\cdot)$, which comes from the previous layer $l-1$, and the weights vector w of the layer l. h represents a specific neuron in its respective layer and b_l corresponds to bias weight. Then, the neuron output is a spatial transformation of r_i^l by the activation function, in this case, logistic (Eq. 2).

Consider that when $l = 1$, $r_i = \sum_h w_{hi} x_h$ (i.e., r_i is the input to i-th neuron of the first hidden layer). w_{hi}^l is the weight of the i-th input that is connected from the h neuron of the $(l-1)-th$ layer to the $l-th$ layer. φ_h^{l-1} is the $h-th$ output node on the $(l-1)-th$ hidden layer. Also, L is the total number of hidden layers, and H_l is the total of hidden neurons in the layer l.

$$r_i^l = \sum_h w_{hi}^l \, \varphi_h^{(l-1)}\left(r_h^{(l-1)}\right), \tag{1}$$

$$\varphi^{(l)}\left(r_i^l\right) = \frac{1}{\left(1 + e^{-\alpha r_i^l}\right)}, \tag{2}$$

where α is the slope parameter of the sigmoid function. The output j of the last layer in a DL-MLP is $z_j = F_j(x, W)$. It depends on all parameters w on all hidden layers, and it uses a linear combination of $\varphi(\cdot)$ to estimate of the transformation real $\{\boldsymbol{f}: \mathbb{R}^{N} \longrightarrow \mathbb{R}^{J}\}$, which partitions the input space $\{\boldsymbol{f}: \mathbb{R}^{N} \longrightarrow \mathbb{R}^{J}\}$ into J classification regions $\mathbb{R}^{J}$ [33].

The training is performed by methods based on the stochastic gradient descent [29]. For example, Adagrad adjusts the learning rate parameters; it performs changes of greatest magnitude for the less frequent parameters, while smaller changes are done for those that more often appear. In contrast, Adadelta searches for a reduced aggressiveness by decreasing the learning rate instead of accumulating previous gradients, thus offering a fixed-size accumulation. Another example is Adam, which modifies the learning rate and uses the exponentially decreasing average of past gradients for each parameter [34].

The main goal of training algorithms is tuning the neural network weights to allow that $F(\cdot)$ perform a non-linear input-output mapping for any classification or general regression problem; that is: if x is the input vectors set, and t is the wanted responses set, then the function $F(x)$ should estimate the best parameters associated with those specific sets and approximate the unknown function $f(x)$. The training method is designed to minimize the error ε (Eq. 3), which is arbitrarily small [32].

$$|F(x) - f(x)| < \varepsilon. \tag{3}$$

2.2 Prototype Selection Methods with Over-Sampling

Class imbalance exists when the difference between the number of instances in one or more classes is much more significant than the number of instances in another class (or classes) [15], [16]. It is commonly measured as a ratio (Eq. 4).

$$IR = \frac{|Class_{maj}|}{|Class_{min}|}, \tag{4}$$

where $Class_{maj}$ and $Class_{min}$ are the most and less represented classes in the dataset, respectively. Because larger values of IR imply higher class imbalance [31], it allows a quick overview of the nature of data and imbalance severity [10].

In data sampling, researchers develop new proposals based on two primary questions 1) what samples should be eliminated and 2) how to generate new informative samples. This has led to informed strategies that heuristically answer these questions. Initially, achieving equal size between the classes was considered the solution to the class imbalance problem. However, increasing or reducing the classes to improve the classifier performance is insufficient since other complexities, such as high dimensionality, overlap, or class imbalance ratio, should be considered. For this reason, hybrid sampling methods combine over-sampling (*OS*), such as *ROS* and *SMOTE*, with prototype selection techniques to remove only the majority samples. This solves some disadvantages that data sampling methods present when applied separately.

Kubat & Matwin's [35] suggest that the prototype selection strategies should remove three kinds of majority samples: 1) borderline, 2) outlier, and 3) redundant. Tomek Links and editing techniques have been proposed to eliminate the examples of categories 1 and 2. In the case of redundant samples, the classical condensed nearest neighbor is employed to build a consistent set [30]. Table 2 summarizes the methods used in this paper.

Table 2. Description of data sampling approaches and editing methods studied in this work.

Over-Sampling Algorithms (OS)		
Method	SMOTE, Synthetic Minority Over-Sampling Technique	[22]
Description	SMOTE interpolates nearby instances of the minority class to create new synthetic instances until the balance of the class distribution is achieved. It takes a random instance from the minority class and finds its k closer neighbors (of the same class); next, it chooses some of those closest neighbors and generates a new instance.	
Method	ROS, Random Over-Sampling	[36]
Description	ROS duplicates minority instances randomly until the class distribution balance is reached	
Under-Sampling Editing Methods (US)		
Method	ENN, Editing Nearest Neighbor	[18]
Description	ENN calculates the k nearest neighbors of any instance a if the label of most of its nearest neighbors is not equal to a label, then a is erased because it can be noisy or one instance on the overlap region.	
Method	TL, Tomek Links	[19]
Description	TL are sets of pairs of instances a and b from different classes, where not exist an instance c, such that $\mathrm{d}(\mathrm{a},\mathrm{c}) < \mathrm{d}(\mathrm{a},\mathrm{b})$ or $\mathrm{d}(\mathrm{b},\mathrm{c}) < \mathrm{d}(\mathrm{a},\mathrm{b})$, where d is the distance between the paired instances. If two instances form a TL, both are deleted [30].	
Hybrid Sampling Approaches *(HS)*		
Method	SMOTE-TL, SMOTE-ENN	[30]
Description	The SMOTE method is used to balance the class distribution; next, editing methods (TL and ENN) are applied to erase the noise or overlap samples.	
Method	ROS-TL, ROS-ENN	[30]
Description	ROS balances the class distribution, then editing methods (TL and ENN) clean the resultant dataset of noise or overlap samples.	

2.3 Deep Edition Strategy

DL-MLP is sensitive to class imbalance [2]. During neural network training, the majority classes usually benefit, while the minority classes often lose accuracy [7]; this is constant for binary and multi-class problems [14]. For this reason, researchers have worked to address the class imbalance problem in this context [10], [37], [38].

Unlike classical hybrid strategies where both over-sampling and under-sampling methods are applied consecutively in the input feature space, Rendon et al. [16] show that the cleaning strategies like ENN applied on the ANN output obtain better classification results than using the input feature space. This method can be summarized in four steps: 1) training the ANN with a Training Dataset (*TRD*) balanced by any over-sampling method, 2) applying ENN cleaning technique to the output of the ANN, and removing those instances from balanced TRD (*BD*) that are related to outputs suspected to be noise or atypical values, 3) training the ANN again with the resultant BD_{edited}($BD_{edited} \subseteq BD$), and 4) to evaluate the ANN performance with the Test Dataset (*TD*), where $TRD \cap TD == \emptyset$, i.e., they are disjoint datasets.

Motivated by the results obtained by Rendon et al., we hypothesize that the cleaning strategies can be improved if, instead of using the original feature space or the neural network output, the nearest neighbor search is performed on a transformed dataset containing high-level feature vectors. These vectors are obtained from the hidden layer representations of the neural network, where each hidden node encodes a dimension of the problem. This transformation provides a more structured and abstract feature space, potentially improving classification performance. As presented in previous works [28], [29] the values generated in the hidden layers correspond to a new vector of size H_l (where H is the total of hidden neurons in layer l), i.e., a new data representation of the input data (please see Eq. 1 and Eq. 2).

Some authors suggest that this transformation results in a high-level feature vector [9]. As input data are mapped into a different feature space (hidden space), class separation may become more effective, even for non-linearly separable problems [32]. However, it is essential to note that this transformation and subsequent sample editing may affect the class balance. While the editing process primarily removes noisy or redundant instances from the majority class, in some cases, it may also reduce instances from the minority class. Therefore, monitoring the Imbalance Ratio (IR) after this transformation is recommended to determine if further adjustments are necessary before the second training phase.

Therefore, for the l-th layer, we can extract a transformed dataset to which the cleaning strategy could be applied to identify the samples that can be eliminated. However, the deletion process is noteworthy because it is performed in the balanced dataset, that is, in the input feature space.

Input: TRD: Training dataset; TD: Test dataset; // TRD ∩ TD == ∅
Output: Values of *g-mean*; // *Measurement criteria*

Read DL-MLP, SMOTE, ROS, ENN, TL and OSS configuration files;
OVER-SAMPLING PROCEDURE:
$OS = \{\text{SMOTE, ROS}\}$; // *Over-sampling methods*
for $s = 0$ to $s < \|OS\|$ **do**
    $\text{BD}_s = \text{Over-Sampling(TRD,}OS[s]\text{)}$; // BD_s:*Balanced* TRD *with method s*
end for
FIRST TRAINING:
for $s = 0$ to $s < \|OS\|$ **do**
    for $l = 0$ to $l < L$ **do**
        // *L is the total of DL-MLP layers, including the input and output layers*
        // *Train* DL-MLP, I_{init} *epochs with balanced dataset* (BD_s)
        $\text{OUT}_s^l = \text{DL-MLP}(\text{BD}_s, I_{init})$;
        //OUT_s^l *saves hidden neurons output values of each layer "l", including the input and output layer of DL-MLP*
    end for
end for
DEEP EDITING PROCEDURE:
$E = \{\text{ENN, TL, OSS}\}$; // *Editing methods*
for $s = 0$ to $s < \|OS\|$ **do**
    for $edit = 0$ to $edit < \|E\|$ **do**
        for $l = 0$ to $l < L$ **do**
            $\text{OUT}_{s,edit}^l = \text{Edition}(\text{OUT}_s^l, \text{E}[edit])$; // *Apply editing method* E[*edit*] *to layer "l"*
            $\text{BD}_{s,edit}^l = \text{REMOVE}(\text{BD}_s, \text{OUT}_{s,edit}^l)$;
            //*Remove samples from* BD_s, *that correspond to neuron outputs* ($\text{OUT}_{s,edit}^l$) *of the layer "l" that edition method "edit" consider suspicious to be outliers or noise*
        end for
    end for
end for
SECOND TRAINING:
for $s = 0$ to $s < \|OS\|$ **do**
    for $edit = 0$ to $edit < \|E\|$ **do**
        for $l = 0$ to $l < L$ **do**
            //*Train DL-MLP during* I_{end} *epochs with edited and balanced dataset; and test it with TD*
            $g\text{-}mean_{s,edit}^l = \text{DL-MLP}\,(\text{BD}_{s,edit}^l, \text{TD}, I_{end})$;
        end for
    end for
end for
//$g\text{-}mean_{s,edit}^l$ *obtained by applying OS method "s", and editing technique "edit" in the layer "l"*
return ***g-mean***;

Fig. 1. Implementing the editing methods on the deep hidden neural network space.

This study uses a DL-MLP, two cleaning methods (ENN and TL), and the methodology from Ref. [16]. The primary purpose of this paper is to study the classifier performance when ENN and TL methods are applied to the outputs of a hidden layer. We named this procedure *Deep Edition* because the nearest neighbor search processing occurs in the DL-MLP hidden space instead of the feature space. Fig. 1 shows the Deep Edition methodology, which can be divided into four stages:

1. **Over-sampling procedure**: TRD is balanced by SMOTE or ROS until reaching a relative class balance ($IR \approx 1$, Eq. 4).
2. **First training**: the DL-MLP is trained l_{init} epochs with resultant balanced training dataset (BD_s).
3. **Deep edition procedure**: for a particular hidden layer, the cleaning algorithm is applied. Samples that the cleaning method marked in the hidden layers as suspicious to be outliers or noise are removed from BD_s.
4. **Second training:** the DL-MLP is trained again during I_{end} epochs with the cleaned dataset $BD_{s,edit}^l$ and the classification performance is evaluated by using TD.

Figure 2 graphically summarizes the deep edition method, highlighting the main steps to simplify its understanding and working path. The source code of the proposed strategy is also available[1]. The deep edition block is an essential part of the proposed methodology for improving the classifier's performance in the presence of imbalanced data. This stage employs the two edition methods selected for this experimentation, ENN and TL, to eliminate noisy or irrelevant samples from the dataset.

The ENN algorithm identifies those instances in the majority class whose class membership is inconsistent with their k-nearest neighbors; these instances are considered noise and removed from the dataset. In other words, ENN looks for misclassified samples and removes them, assuming they are noise. On the other hand, the TL algorithm identifies the samples that form a Tomek-Link, which is a pair of samples of different classes that are the nearest neighbors of each other. Such samples are likely to be noisy or misclassified instances, so the algorithm removes the nearest neighbor of the opposite class sample.

[1] https://github.com/tonitk1978/Edicion_Profunda.git

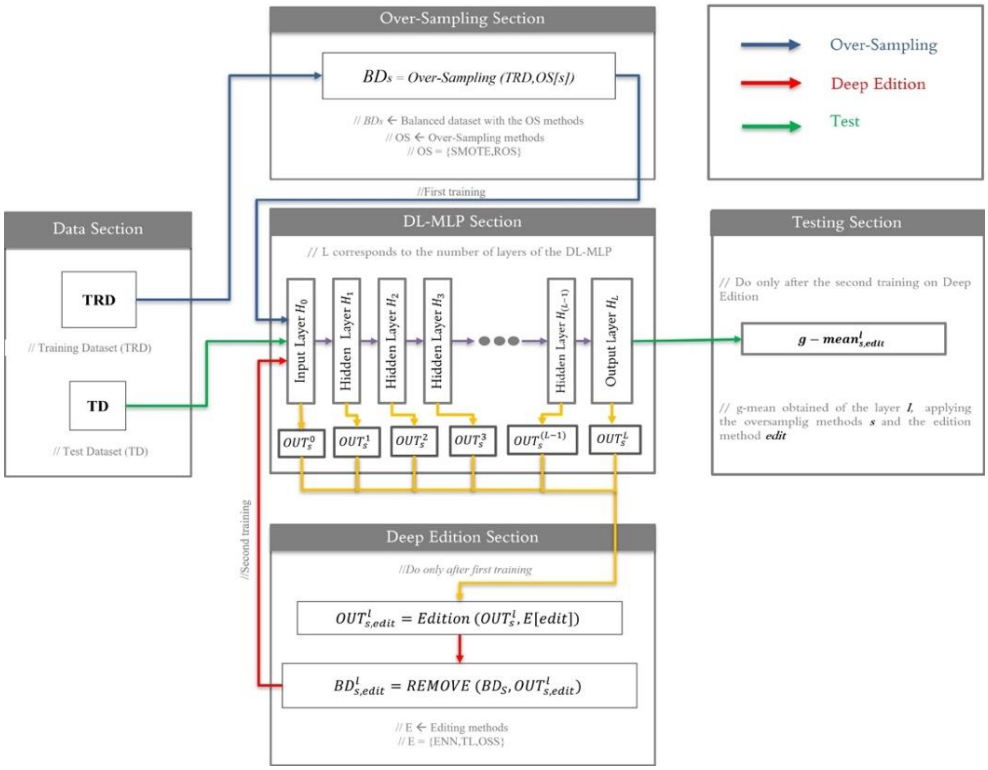


Fig. 2. Flowchart of the proposed Deep Edition strategy. Blue lines correspond to the over-sampling stage; yellow lines show that in each hidden layer, a new transformed dataset can be obtained where the nearest neighbor search can be performed; red lines indicate that the elimination of samples is performed in the original dataset in the input feature space.

All these editing methods are systematically implemented across each hidden layer of the DL-MLP architecture. This hierarchical approach ensures that noise sample removal occurs at diverse abstraction levels, substantially enhancing the classifier's generalization capabilities. By eliminating noise and outliers from the dataset, the classifier can more effectively focus on learning the intrinsic underlying patterns within the data, consequently leading to superior performance in minority class classification, which is critical when dealing with highly imbalanced datasets.

3 Experimental Set-Up

This section details the datasets and their characteristics, the algorithms' parameters, and the performance metrics used in the experimental study.

3.1 Data description

This work uses Indian, Salinas, PaviaU, and Pavia datasets obtained from GIC (Grupo de Inteligencia Computacional, by its Spanish acronym)[2] from hyperspectral images. The Indian and Salinas datasets were captured by NASA's AVIRIS (Airborne Visible / Infrared Imaging Spectrometer) sensor[3]. They provided 224 bands at 10 *nm* width, ranging from 400 to 2500 *nm* while excluding water absorption bands, with a spatial resolution of 3.7 *m/pixel* for both Indian and Salinas. The ROSIS (Reflective Optics System Imaging Spectrometer) sensor acquired images from Pavia University (PaviaU) and Pavia Center (Pavia) with 103 and 102 bands, respectively; their width is 4 *nm,* and both have a resolution of 1.3 *m/pixel*.

Table 3 presents a compilation of the main characteristics of each dataset, highlighting the number of classes and the class Imbalance Ratio *IR*, which is computed in Eq. 4.

[2] http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
[3] https://aviris.jpl.nasa.gov/

Many hyperspectral remote sensing works analyze the Indian, Salinas, PaviaU, and Pavia datasets. However, most research does not consider class 0 (background) for the objectives of the studies because the usual solution for addressing class imbalance is to sample each class uniformly, often leading to the background being overlooked as a class (see Refs. [39], [40], [41]). Nevertheless, we are interested in going deep into the study of highly imbalanced problems in classifying hyperspectral remote sensing images. Therefore, we considered class 0 in the four studied datasets in this work.

Thus, the resultant datasets are highly imbalanced; for example, the Indian dataset has 10,776 and only 20 samples in the majority and minority class, respectively, i.e., *IR*=538.8; a similar characteristic is observed in Pavia. In the remaining datasets (see Table 3), the class imbalance rate (*IR*) is lower than in India and Pavia but still highly imbalanced.

Table 3. Database characteristics. Observe their (*IR*) mainly.

Dataset	Sensor	Classes	Samples	Bands	Width	Resolution	IR
Indian	AVIRIS	17	21,025	220	10	3.7	538.8
Salinas	AVIRIS	17	111,104	224	10	3.7	62.2
PaviaU	ROSIS	10	207,400	103	4	1.3	173.8
Pavia	ROSIS	10	783,640	102	4	1.3	236.7

3.2 Free parameters specification

Over-sampling and editing methods used on the input or feature space were obtained from Imbalanced-learn to Python module[4]. Each method was used with the parameters assigned by default, i.e., for ENN $k_{\text{neighbors}} = 3$; SMOTE $k_{\text{neighbors}} = 5$. Deep Edition methods (see sec. 2.3) used the same parameters and source code hosted on website[5].

DL-MLP was developed using TensorFlow 2.0 and Keras 2.3.1 frameworks; the configuration is shown in Table 4. Although there are established methods for choosing the number of layers and neurons per layer in deep networks, these are typically designed for specific purposes and applications, suggesting a particular investigation [42], such optimization is not the focus of this study. However, a general method commonly used in most cases serves as the best starting point to evaluate our proposed approach, recognizing that the primary goal of our research is to illustrate the effects of applying the editing methods in the hidden space rather than the feature space of the MLP.

The configuration of the DL-MLP was determined using a trial-and-error iterative strategy [43]: 1) we began with two hidden layers (the minimum requirement for defining it as a DL-MLP) and assessed the network's performance on the validation set while increasing the number of layers; when performance on the validation set ceased to improve (or began to degrade), we identified four layers, and 2) we started with a small number of neurons per layer, gradually increasing this number; again, we evaluated the network's performance after each adjustment, and when performance on the validation set stopped improving (or started to degrade), we found the combination reported in Table 4. It is important to emphasize that we selected this network configuration not to optimize classification performance but to provide a stable experimental framework for evaluating the impact of sample editing in the hidden space.

The learning rate was established at 0.001, and the training algorithm was Adam using 500 epochs and a batch size of 500 for the databases Salinas and Indian, and 250 epochs with a batch size of 1000 for the rest of the databases. In Fig. 1, I_{init} and I_{end} were set to $I_{init} = I_{end} = 250$ epochs and a batch size of 500 to Salinas and Indian datasets, and $I_{init} = I_{end} = 125$ epochs and batch size of 1000 to PaviaU and Pavia.

To enhance the reliability of our results, each DL-MLP configuration was trained and tested five times with different weight initializations. This ensures that a particular random initialization does not bias the reported findings. Given that the datasets have distinct characteristics, different configurations were tested. However, the results presented correspond to the best neural network configuration that provides a stable foundation for investigating the core research question: how editing methods influence classification performance when applied in the hidden space instead of the input feature space.

[4] https://imbalanced-learn.org/stable/
[5] https://github.com/tonitk1978/Edicion_Profunda.git

Table 4. DL-MLP configuration: number of layers, neurons by layer, and activation function for each case. The number of input and output neurons corresponds to the datasets presented in Table 3; they appear in the same order, i.e., first Indian, Salinas, and so on.

Layer	Neurons	Activation Function
Input layer (H0)	[220/224/103/102]	--
Hidden layer 1 (H1)	50	ReLU
Hidden layer 2 (H2)	40	ReLU
Hidden layer 3 (H3)	30	ReLU
Hidden layer 4 (H4)	20	ReLU
Output layer (H5)	[17/17/10/10]	Softmax

3.3 Classifier performance

To measure the classifier performance, the hold-out method was used [32]; it randomly split the originals datasets (sec. 3.1) on training (TRD) 70% and testing (*TD*) 30%, where $TRD \cap TD == \emptyset$ (disjoint sets).

When evaluating the performance of a classifier for hyperspectral remote sensing images, accuracy or Kappa metrics are commonly used. However, these metrics can be inadequate when the dataset is highly imbalanced. For instance, suppose we have a dataset with ten classes, where each of the first nine classes has 100 samples, and the last class has only ten samples. If the classifier accurately identifies all of the majority classes and misclassifies the minority class, the accuracy would be 98.9%, and the Kappa value would be 0.987. On the surface, it may appear that the classifier's performance is acceptable. However, the minority class is ignored, as noted in previous studies [44].

Thus, alternative metrics for assessing classifier performance in imbalanced datasets have been suggested in the literature: AUC, the area under the Receiver Operating Characteristic (ROC) curve, which offers a thorough evaluation of the classifier's capability to differentiate among the different classes, precision/recall, which focuses on the classifier's ability to correctly identify positive samples (in this case, minority class samples) and reduce false positives, *F1-score*, which is a harmonic mean of precision and recall, among others [45]. These alternative metrics provide a more nuanced evaluation of the classifier's performance in highly imbalanced datasets and help to avoid ignoring the minority classes; however, they are not very informative when multiple datasets are present.

In this work, we use the geometric mean (*g-mean*), which is a standard metric to assess the overall effectiveness of the classification when the training dataset is imbalanced and multi-class because it is sensitive to the classifier's performance on each class [46], [47], [48]. In the example mentioned above, the *g-mean* value will be 0; i.e., it reports that any class was misclassified. It is very useful when the dataset has multiple classes and high imbalance, just like the datasets studied in this paper (see Table 3). Eq. 5 defines the *g-mean* as a geometric average of the partial accuracy for each class [49], where *J* is the total of classes, and acc_j is the classifier accuracy on the *j* class, i.e., I_{init}= (samples of the class *j* correctly classified)/(total of instances of the class *j*). Only when all accuracy rates of all classes are high enough, the *g-mean* can achieve a high value. It can be defined as follows:

$$g - mean = \sqrt[J]{\prod_{j=1}^{J} acc_j}. \quad \textbf{(5)}$$

Finally, to streamline the analysis of our results, we employ the Friedman rank test [50]. This test assigns a rank to each algorithm across all datasets, with the best-performing algorithm receiving rank 1, the second-best receiving rank 2, and so on. If there is a tie, we calculate the average ranks.

4 Experimental results

In this section, the main experimental results are presented. First, we analyze the performance of the editing methods (sec. 2.3) on each of the hidden layers of the DL-MLP (sec. 2.1). Then, the effectiveness of the classification is compared, considering all editing methods presented in this paper. Finally, we study the performance of the eleven best editing sampling methods in each of the datasets used in this work, and a visual analysis of them is performed.

4.1 Analysis and comparison of editing methods performance in the hidden space

This work aims to analyze the performance of training a DL-MLP with *TRD* edited in the hidden space, compared against the effectiveness of editing the *TRD* in the input space or output layers. Thus, we present the results layer by layer in Fig. 3; it summarizes the corresponding *g-mean* rank values obtained from DL-MLP classification (performed five times to reduce the effect due to randomness of the classifier) over all the datasets (see section 3.1) for each editing method in each DL-MLP layer.

Subfigures (of Fig. 3) represent the experimental results of every data sampling method. Subfigures (of Fig. 3) represent the experimental results of every data sampling method. Then, the first one corresponds to TL (a), the second one to ENN (b), the third one to SMOTE-TL (c), the fourth one to SMOTE-ENN (d), the fifth one to ROS-TL (e), and finally the ROS-ENN (f) approach. In addition, in these subfigures, axis *x* shows the several layers of the DL-MLP, and axis *y* corresponds to the *g-mean* ranks values computed from the average g-*mean* over all datasets, i.e., results shown in the subfigures were obtained from the average of the model performed over the four hyperspectral datasets (see Table 3).

The results shown in Fig. 3 show that the best option for applying TL and ENN is the input layer (or the feature space). However, these methods do not include an over-sampling method to deal with imbalanced class scenarios; thus, they only focus on deleting noise or outlier samples, which we consider insufficient when the dataset is highly imbalanced (see Table 4).

Theoretical and empirical studies have shown that neural network performance is negatively affected if the class imbalance rate is not reduced. This is because, in the training stage, the neural network could ignore the less represented classes, either partially or altogether (see Refs. [7], [8]). Thus, it is necessary to use over-sampling methods, especially when the dataset is highly imbalanced, like the ones studied in this work.

On the other hand, when ROS or SMOTE are combined with editing sampling methods, the best results are obtained in the hidden, output, or input space. Overall, Fig. 3 evidences that 50% of the best ranks obtained by editing methods (which include ROS or SMOTE) are localized in the hidden layers, while 25% are found in the output layer and 25% in the input layer. In other words, in most cases, the best option is to apply editing methods in the hidden layers.

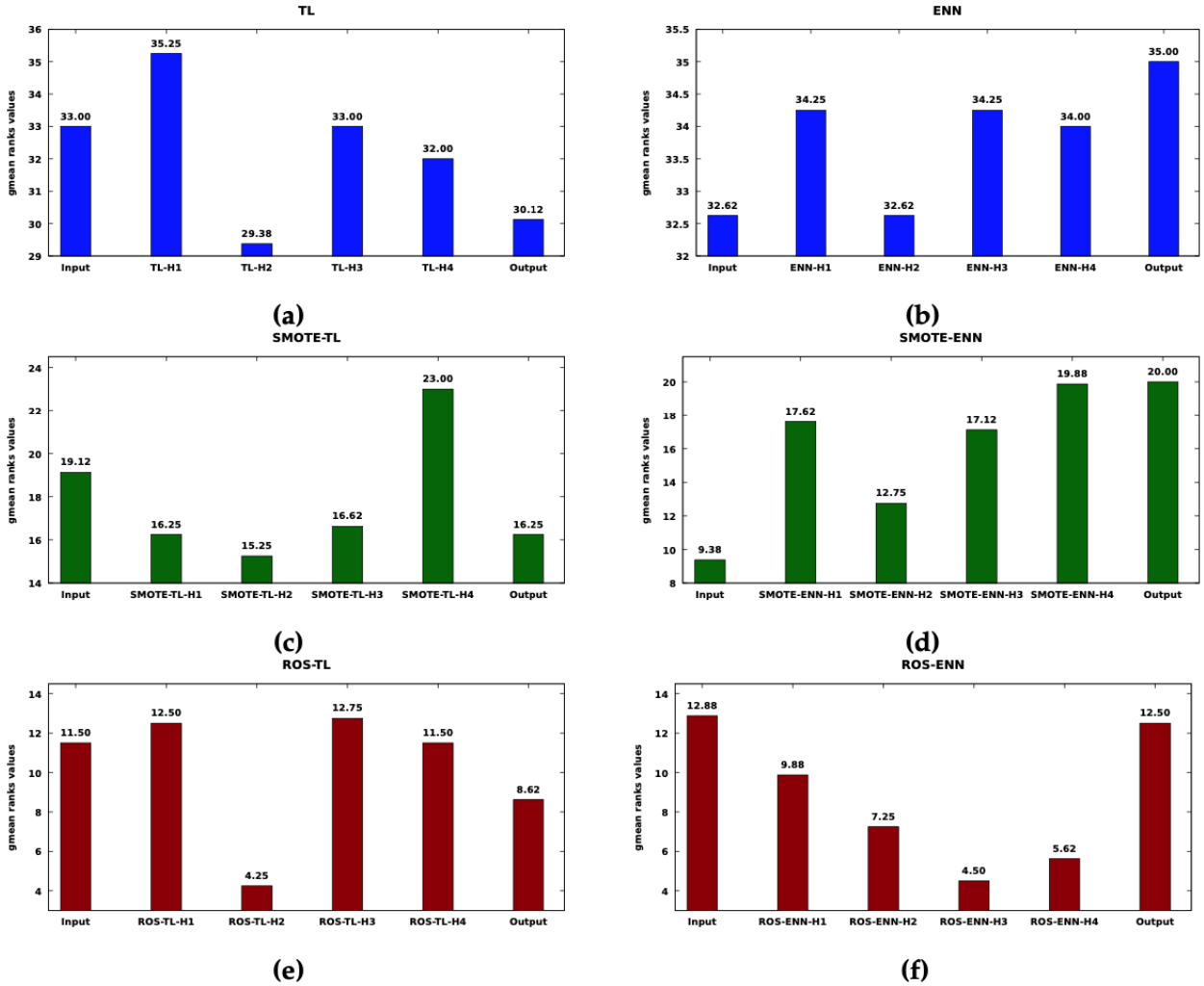


Fig. 3. Ranks of classifier performance by layer of each editing method studied in this work. H1, H2, H3, and H4 correspond to DL-MLP hidden layers, where H1 is the first hidden layer, H2 the second one, and so on (see Table 4 *Input* and *Output* correspond to H0, and H5 layers or input and output DL-MLP layers, respectively.

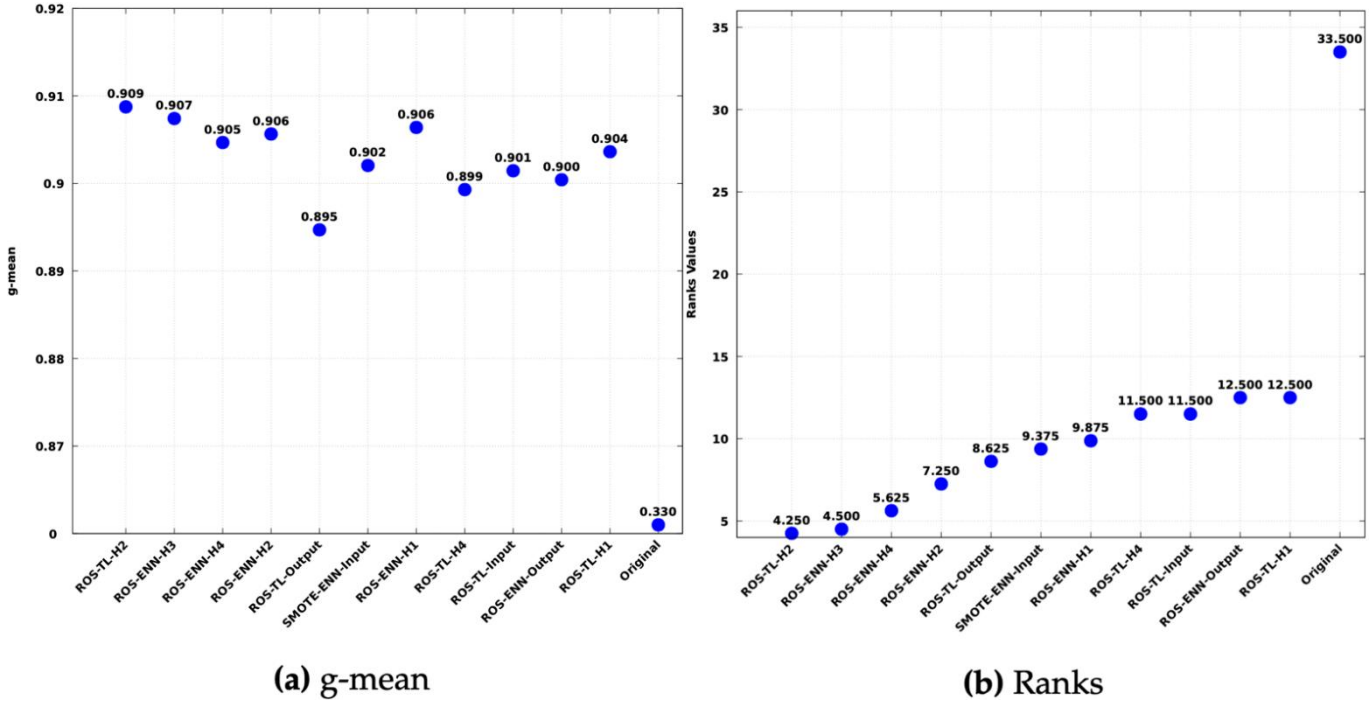
After carefully analyzing the results presented in Figure 3, below is a more precise analysis, recalling that lower *rank* values (according to the Friedman test applied to the g-mean metric) indicate better performance:

- Tomek Links: The input layer shows a *rank* value of 33.00, while the H2 layer (29.38) improves this result. H1 (35.25) and H3 (33.00) perform slightly worse than the input, while H4 (32.00) and the output (30.12) maintain values close to or below that of the input.
- Edited Nearest Neighbors: The input layer (32.62) outperforms H1 (34.25), H3 (34.25), H4 (34.00), and the output (35.00). H2 (32.62) is similar to the input layer.
- SMOTE-TL: The worst location appears to be H4 (23.00), while layers H1 (16.25), H2 (15.25), H3 (16.62), and the output (16.25) exhibit more favorable values than the input (19.12).
- SMOTE-ENN: The best performance is concentrated in the input layer (9.38). In comparison, layers H1 (17.62), H2 (12.75), H3 (17.12), H4 (19.88), and the output (20.00) yield higher *ranks*.
- ROS-TL**:** The input layer (11.50) offers competitive results, although layer H2 (4.25) and the output (8.62) surpass it. H1 (12.50) and H3 (12.75) lag slightly behind. H4 (11.50) equals the input.
- ROS-ENN: All hidden layers improve the value of the input layer (12.88). In particular, H3 (4.50) and H2 (7.25) stand out due to their lower ranks, while the output layer (12.50) achieves a slightly worse value than the input.

These findings demonstrate that editing methods significantly vary in performance depending on the network layer where they are applied. Especially when combining over-sampling methods (ROS or SMOTE) with under-sampling or noise removal methods (TL or ENN), the hidden layers often present a transformed feature space that, in specific configurations, enhances the network's capacity to handle scenarios with high class imbalance. Thus, according to the ranks analysis, the best results typically appear in intermediate layers, such as **H2** or **H3**, depending on the specific combination of sampling techniques applied.

4.2 Performance comparison of studied editing methods

Fig. 4 presents a comparison of the effectiveness of several editing methods. Axis *x* displays the top eleven editing methods, including the classifier performance on the original dataset (i.e., the dataset without preprocessing). The suffixes H# (in the names of the methods) denote the respective layers of the DL-MLP (see Table 4); for example, H1 represents the first hidden layer, H2 indicates the second hidden layer, and so on, while the suffixes *Input* and *Output* refers to the input and output layers, or H0 and H5 layers. Axis *y* in Fig. 4a illustrates the average *g-mean* values across all datasets, and axis *y* in Fig. 4b shows their respective ranks. Fig. 4 complements Fig. 3, and both are useful for explaining the behavior of the editing methods discussed in section 2.3.



(a) g-mean **(b)** Ranks

Fig. 4. Performance comparison of the eleven best editing methods, including the original dataset, represented as (a) average g-mean values across all hyperspectral datasets and (b) rank values obtained by applying the Friedman test over the g-mean results.

Figure 4(a) shows that the *g-mean* for the original dataset is *0.330*, indicating that the class imbalance problem significantly impacts these methods. In other words, the classifier's performance is notably influenced by class imbalance, even when the quality of the training dataset is enhanced. The experimental results demonstrate that when ROS balances the *TRD*, the *g-mean* values exceed *0.900*, with differences across various configurations. This highlights the apparent necessity to balance the class distribution, as the top eleven results, including the original *TRD* without edited processing, reveal that the first three positions are associated with editing methods applied in the hidden layers (ROS-TL-H2, ROS-ENN-H3, and ROS-ENN-H4), achieving *g-mean* values of *0.909*, *0.907*, and *0.905,* respectively. The following positions were achieved through several ROS configurations to confirm the benefits of editing the TRD used by DL-MLP. These results indicate a trend toward improved performance when the TRD is edited within the hidden layer compared to when it is not edited or edited in the input or feature spaces.

The rank analysis presented in Fig. 4b reinforces these findings, indicating that methods utilizing hidden layer editing consistently achieve lower ranks, with ROS-TL-H2 and ROS-ENN-H3 receiving ranks of *4.250* and *4.500,* respectively. The

Original dataset records the highest rank (*33.500*), showcasing the considerable improvement attained through the proposed editing methods. The effectiveness of combining ROS with hidden layer editing is particularly evident, as *g-mean* values for modified approaches consistently exceed the baseline of the original configuration, maintaining values above *0.895* across most modified implementations. This performance difference underscores the significance of addressing class imbalance through advanced editing techniques, especially those operating in the hidden layer space of deep learning architectures.

4.3 Editing methods analysis performance on individual hyperspectral image datasets

In Fig. 5, the g-mean values are shown for each hyperspectral image dataset, where the x-axis represents the eleven best data sampling methods plus the original dataset, and the y-axis represents the corresponding g-mean values. The classifier's performance on each dataset is distinguished by color: red for Salinas, green for PaviaU, blue for Indian Pines, and purple for Pavia.

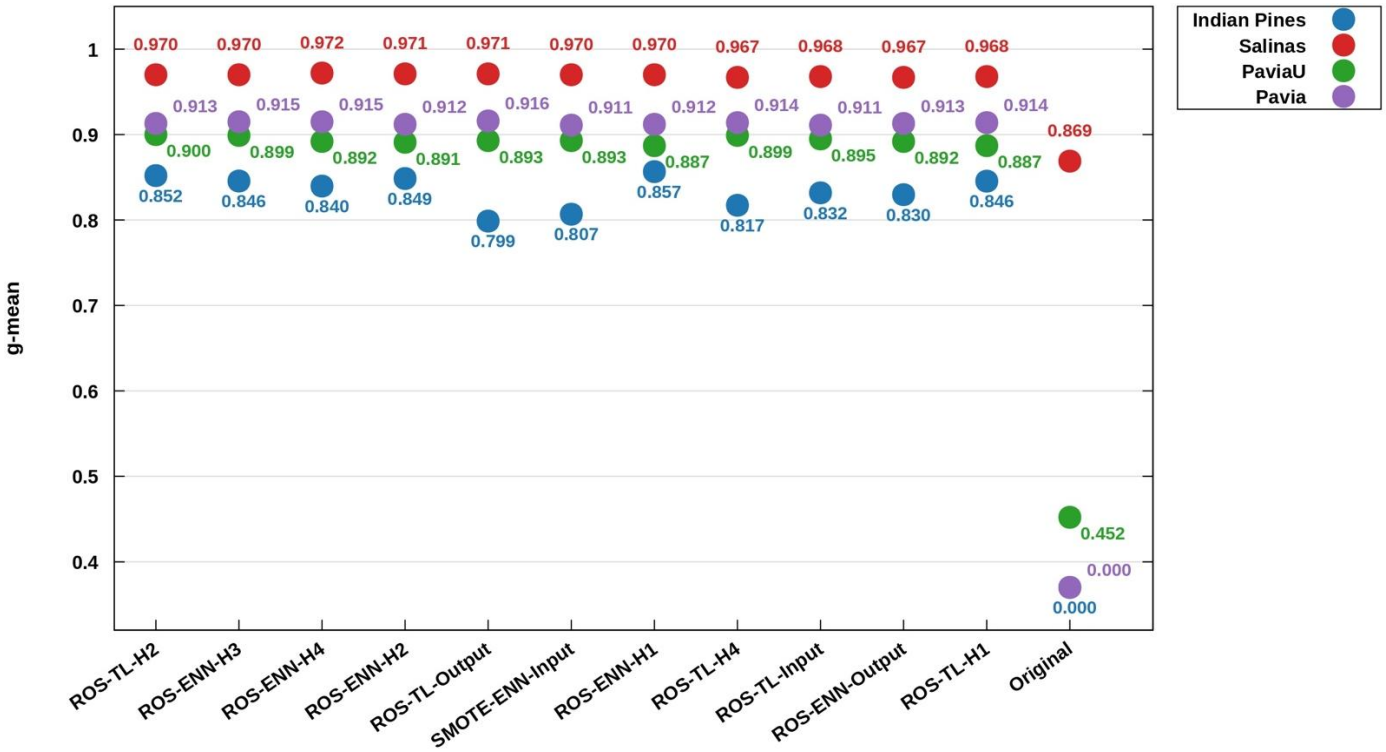


Fig. 5. g-mean values per hyperspectral dataset using the eleven best sampling configurations, including the performance of the original dataset.

It is observed in Fig. 5 that values of *g-mean =0.0* were obtained for the Indian Pines and Pavia datasets when the *original TDS* was used to train the classifier. This critical finding provides substantial evidence of the severe impact of class imbalance on classifier performance, particularly in complex hyperspectral scenarios. In contrast, the Salinas dataset demonstrates notably better performance without processing (original TDS, *g-mean* = 0.869), which aligns with its characteristics as the less imbalanced dataset in our study. PaviaU presents an intermediate performance with the original dataset (*g-mean = 0.452*), further supporting the correlation between imbalance severity and classification degradation.

The systematic analysis reveals that over-sampling methods (ROS and SMOTE) substantially enhance classifier performance across most datasets. This improvement is particularly evident in the performance of ROS-TL-H2 and ROS-ENN-H4, which achieve the highest *g-mean* values across all datasets: Salinas (*0.970* - 0.972), Pavia (*0.913 - 0.915*), PaviaU (*0.900 - 0.892*), and Indian Pines (*0.852 - 0.840*). The performance enhancement is most pronounced in the more imbalanced datasets, where the original classifier struggles significantly.

A deeper examination of the results demonstrates that combining oversampling and editing methods in the hidden or output space consistently produces superior results across all datasets compared to non-mixed approaches. This synergistic effect is particularly noteworthy for ROS variants, which maintain remarkably stable performance levels: g-mean values consistently

exceed *0.800* for Indian Pines, *0.890* for PaviaU, *0.910* for Pavia, and *0.960* for Salinas. These robust results highlight the effectiveness of hybrid approaches in addressing class imbalance challenges.

The performance patterns observed across various datasets provide valuable information about the relationship between dataset characteristics and method effectiveness. For example, while Salinas exhibits relatively high performance with minimal processing, the more challenging Indian Pines dataset demonstrates the critical need for advanced balancing techniques. These performance variations across datasets emphasize the importance of considering dataset-specific characteristics when selecting and implementing class-balancing strategies.

Furthermore, the consistently superior performance of ROS-based methods, particularly when combined with hidden layer editing techniques, suggests that this approach effectively addresses the fundamental challenges of class imbalance while preserving essential spectral-spatial relationships within the hyperspectral data. This observation has significant implications for designing and implementing future classification systems for hyperspectral image analysis.

The results illustrated in Fig. 5 demonstrate that class imbalance can significantly degrade classifier performance in highly unbalanced datasets such as Indian Pines and Pavia. In contrast, performance remains acceptable even without applying over-sampling techniques in databases with less imbalance, such as Salinas. The implementation of hybrid methodologies combining over-sampling and editing (either in the hidden or output layer) substantially increases g-mean values across all analyzed scenarios, which corroborates the efficacy of these strategies in addressing the imbalance problem in hyperspectral contexts. Mainly, "*Deep Edition*" demonstrates a more pronounced impact in those datasets characterized by more significant imbalance (such as Indian Pines), given that in these cases, the classifier experiences more severe degradation when using the original data without implementing any compensatory balancing strategy.

5 Discussion

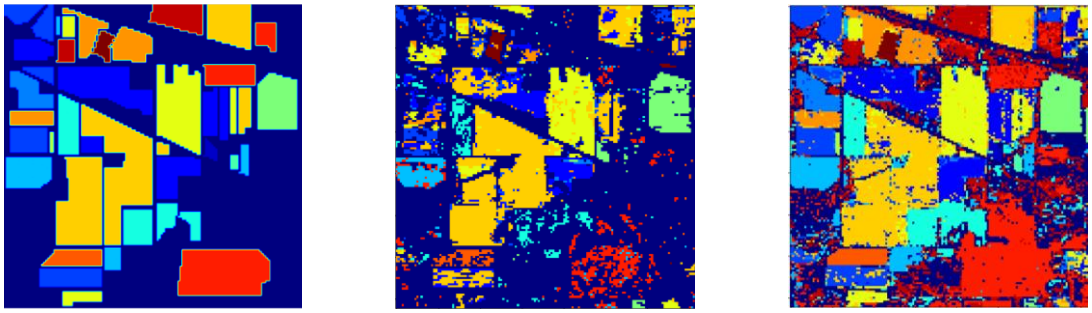
The experimental results demonstrate that when classifying highly multi-class imbalanced hyperspectral image datasets, editing sampling strategies (ENN and *TL*) achieve optimal performance when applied in the hidden neural network space or output layer, combined with an oversampling method (as shown in Fig. 4). Although *ENN* and *TL* have traditionally been used to address class imbalance issues [17], [30], [35], their effectiveness in dealing with data overlap and noise is still relevant in contemporary research [51], [52], [53], [54], [55], [56], [57].

A key innovation in this work is the application of (*ENN* and *TL*), combined with (*ROS* and *SMOTE*), in the hidden neural network space rather than in the conventional feature space utilized in previous Nearest Neighbor rule contexts. This approach shows significant potential for enhancing model performance when handling overlapping and noisy data, which are common challenges in real-world machine learning applications, significantly when worsened by class imbalance issues, as noted in other works [31], [58].

Visual analysis reveals a critical trade-off: improvements in DL-MLP performance for minority classes can adversely affect majority class classification. This observation reinforces the need for effective strategies to balance performance across all classes. Regarding computational efficiency, while the deep edition data sampling approach is more complex than traditional feature space editing (requiring hidden space values from the DL-MLP, as detailed in section 2.3), the additional computational cost is justified by the significant accuracy improvements achieved for minority classes.

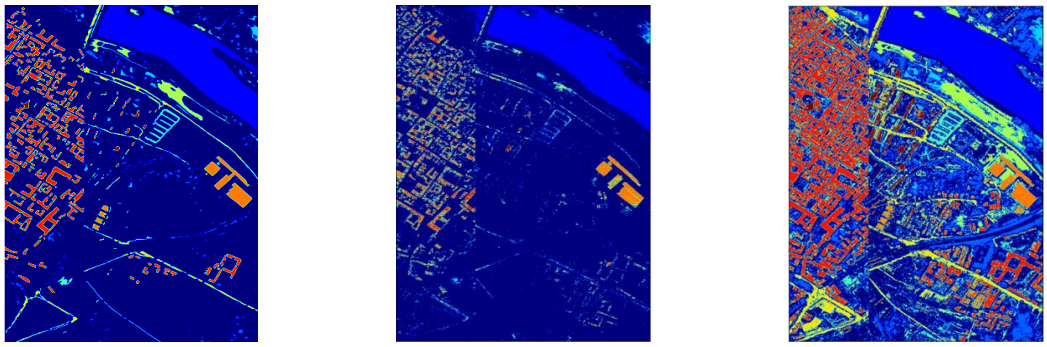
Figures 6 and 7 illustrate the impact of the proposed deep edition data sampling approach on classification performance for the Indian and Pavia datasets, respectively. In both cases, subfigure (a) represents the ground truth, providing a reference for evaluating model predictions. Subfigure (b) presents the classification results obtained by the DL-MLP model trained with the original training dataset (TDS) without preprocessing. These results highlight the challenges associated with class imbalance and noisy data, particularly the misclassification of minority classes. In contrast, subfigure (c) in both figures reveals the improvements achieved by applying the ROS-TL-H2 approach, which leverages a combination of ROS and TL within the hidden neural network space. The visual comparison reveals a notable enhancement in the delineation of minority class regions, indicating reduced misclassification errors and improved representation of underrepresented classes. However, this refinement can sometimes lead to trade-offs in majority class classification. Despite this, the overall balance in class prediction is significantly improved, reinforcing the effectiveness of deep edition data sampling in mitigating the adverse effects of class imbalance while maintaining high classification accuracy.

Although the Deep Edition strategy significantly improves classification performance (particularly for underrepresented classes) it also introduces a higher computational burden. This is primarily due to the need to extract hidden layer activations, execute the editing process across multiple levels of abstraction, and retrain the DL-MLP with the resulting dataset. Therefore, training time is effectively doubled, and memory consumption increases compared to conventional sampling approaches applied in the input or output spaces. These computational requirements may pose limitations in large-scale or time-sensitive applications, especially when resources are constrained. Therefore, while the performance gains are substantial, a balanced evaluation considering both classification accuracy and computational cost is essential for assessing the practical applicability of the proposed method.



(a) Ground truth (b) Original (c) ROS-TL-H2

Fig. 6. Visual comparison of the classes identified in different representative cases of Indian dataset. (a) Ground truth, (b) Maps obtained by DL-MLP trained with the original TDS, i.e., without preprocessing, and (c) Map produced by DL-MLP using the ROS-TL-H2 approach.



(a) Ground truth (b) Original (c) ROS-TL-H2

Fig. 7. Visual comparison of the classes identified in different representative cases of the Pavia dataset. (a) Ground truth, (b) Maps obtained by DL-MLP trained with the original TDS, i.e., without preprocessing, and (c) Map produced by DL-MLP using the ROS-TL-H2 approach.

Table 5 is a tabular representation that systematically synthesizes the fundamental findings of the study, facilitating the identification of performance patterns across different methodological categories and emphasizing the quantitative advantages of deep edition techniques in hidden layers for addressing class imbalance in hyperspectral image datasets.

Table 5. Comparative performance analysis of data sampling and editing methods for imbalanced classification.

Approach	Technique	Best g-mean	Best rank	Optimal layer	Improvement vs. Original	Best Result (Approach)
No Preprocessing (Original)	No preprocessing	0.330	33.500	–	–	– (not applicable)
Traditional Under-sampling	TL (Tomek Links)	0.498	30.120	Input/H2	+50.9%	ENN (0.512)
	ENN (Edited NN)	0.512	32.620	Input/H2	+55.2%	

Traditional Oversampling	ROS (Random Oversampling)	0.781	12.880	Input	+136.7%	ROS (0.781)
	SMOTE	0.764	9.380	Input	+131.5%	
Hybrid Methods (Feature Space)	ROS-TL-Input	0.835	11.500	Input	+153.0%	ROS-TL-Input (0.835)
	ROS-ENN-Input	0.827	12.880	Input	+150.6%	
Output Space Editing	ROS-TL-Output	0.876	8.620	Output	+165.5%	ROS-TL-Output (0.876)
	ROS-ENN-Output	0.868	12.500	Output	+163.0%	
Deep Edition (our approach)	ROS-TL-H2	0.909	4.250	H2	+175.5%	ROS-TL-H2 (0.909)
	ROS-ENN-H3	0.907	4.500	H3	+174.9%	

To highlight the distinctions between the proposed method and existing approaches, Table 6 systematically compares the primary strategies for addressing class imbalance in hyperspectral image classification. This comparison encompasses a comprehensive analysis that spans both the application space and mechanisms, precisely identifying the advantages and limitations of each approach.

Table 6. Comparison of the Main Approaches for Imbalanced Classification.

Author	Reference	Approach	Application Space	Primary Mechanism	Advantages	Limitations
Vuttipittayamongkol et al., 2021 and LeCun et al., 2015	[31], [29]	Traditional under-sampling methods (TL, ENN)	Feature space (Input)	Remove noisy or atypical samples in the original space.	Computational simplicity, preservation of original data structure.	Inability to capture nonlinear transformations, sensitivity to inherent noise.
Özdemir et al., 2021 and Zhong et al., 2021	[27], [26]	Traditional over-sampling (ROS, SMOTE)	Feature space (Input)	Generation of synthetic samples or duplication in the original space.	Better balance in class distribution and preservation of the original structure.	Risk of overfitting, generation of artificial instances in inappropriate regions.
Rendón et al., 2020	[16]	Output space editing	Neural network Output space	Application of editing techniques on neural network outputs.	Exploitation of high-level transformed features.	Limited to transformations of the final layer without exploiting intermediate representations.
Our work	--	Deep Edition (proposed)	Neural network hidden space.	Application of editing techniques on hidden layer representations.	Leveraging hierarchical abstractions, better class separation in transformed spaces.	Higher computational complexity, need for retraining.

Notably, the broader implications of these findings could extend to other deep-learning architectures, particularly those employing fully connected or dense layers [28]. For instance, Convolutional Neural Networks (*CNNs*), which typically include input, convolutional, pooling, and fully connected layers [59], could benefit from this approach. The dense layers in these networks effectively function as MLPs. Furthermore, the flexibility of modern DL architectures, including transfer learning approaches [60]

and auto-encoder pre-training methods [59], suggests the potential adaptability of our findings. Additionally, DL models are versatile and flexible in their architecture and training, which facilitates their design and construction [9]. However, while these results are promising, specific studies are necessary to verify the effectiveness of our proposal in other contexts.

6 Conclusions

This research addresses hyperspectral image classification with significant class imbalance using an innovative "Deep Edition" approach. Our investigation shows that applying editing techniques (ENN and Tomek Links) in the hidden layers of deep neural networks, combined with oversampling methods (ROS or SMOTE), leads to substantial improvements in classification performance compared to traditional methodologies.

Experimental analysis reveals that ROS-TL-H2 and ROS-ENN-H3 configurations achieve g-mean values of 0.909 and 0.907, respectively, representing 175.5% and 174.9% increments compared to the preprocessing-free scenario (0.330). These results significantly surpass the performance of traditional strategies: classical under-sampling methods barely exceed g-mean values of 0.50, while conventional oversampling (ROS, SMOTE) in the feature space reaches approximately 0.78 and 0.76, respectively.

Implementing hybrid techniques in the input space (ROS-TL-Input) improves performance up to 0.83-0.84, and their application in the output layer elevates this value to 0.87. However, the highest values are achieved in the transformed space of hidden layers, consistently exceeding 0.90. This finding underscores the advantage of intervening at intermediate levels of abstraction, where the transformed data structure facilitates more precise identification of noisy or atypical samples.

The proposed methodology introduces greater computational complexity due to the need for retraining after the editing phase. Nevertheless, the additional cost is justified by the substantial improvement in classification performance, particularly for minority classes, which are typically overlooked in high-imbalance scenarios.

Although the proposed method was implemented and evaluated using MLP architectures, its potential extension to other deep learning models, such as convolutional or recurrent networks, remains an open question since no empirical validation was conducted in this regard. This represents a current methodological limitation, and future work should investigate the adaptability of the editing scheme to architectures with different internal representations and learning dynamics.

Future research lines could be oriented towards computational optimization of the process to reduce retraining-associated costs, automated selection of the optimal layer for applying editing techniques through metaheuristics, evaluation in more complex neural architectures, and application of Deep Edition in other domains characterized by high-dimensional imbalanced data.

Deep Edition effectively addresses the class imbalance in hyperspectral image classification by leveraging intermediate representations generated by deep neural networks to enhance discriminative capacity in transformed spaces. Although the experimental results validate its effectiveness in this area, future studies are required to explore its applicability in other domains where class imbalance may influence the generalization capacity of deep learning models.

References

[1] G. Sumbul and B. DemIr, “A Deep Multi-Attention Driven Approach for Multi-Label Remote Sensing Image Classification,” *IEEE Access*, vol. 8, pp. 95934–95946, 2020, doi: 10.1109/ACCESS.2020.2995805.

[2] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, pp. 249–259, 2018, doi: 10.1016/j.neunet.2018.07.011.

[3] D. KoBmann, T. Wilhelm, and G. A. Fink, “Towards Tackling Multi-Label Imbalances in Remote Sensing Imagery,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 5782–5789. doi: 10.1109/ICPR48806.2021.9412588.

[4] X. Gao *et al.*, “An End-to-End Neural Network for Road Extraction From Remote Sensing Imagery by Multiple Feature Pyramid Network,” *IEEE Access*, vol. 6, pp. 39401–39414, 2018, doi: 10.1109/ACCESS.2018.2856088.

[5] X. Li *et al.*, "Dual attention deep fusion semantic segmentation networks of large-scale satellite remote-sensing images," *Int J Remote Sens*, vol. 42, no. 9, pp. 3583–3610, 2021, doi: 10.1080/01431161.2021.1876272.

[6] G. Sumbul *et al.*, "BigEarthNet-MM: A Large-Scale, Multimodal, Multilabel Benchmark Archive for Remote Sensing Image Classification and Retrieval [Software and Data Sets]," *IEEE Geosci Remote Sens Mag*, vol. 9, no. 3, pp. 174–180, 2021, doi: 10.1109/MGRS.2021.3089174.

[7] V. M. González-Barcenas, E. Rendón, R. Alejo, and R. M. Granda-Gutiérrez E. E.and Valdovinos, "Addressing the Big Data Multi-class Imbalance Problem with Oversampling and Deep Learning Neural Networks," in *Pattern Recognition and Image Analysis*, A. Morales, J. Fierrez, J. S. Sánchez, and B. Ribeiro, Eds., Cham: Springer International Publishing, 2019, pp. 216–224. doi: 10.1007/978-3-030-31332-6_19.

[8] J. Błaszczyński and J. Stefanowski, "Local Data Characteristics in Learning Classifiers from Imbalanced Data," in *Advances in Data Analysis with Computational Intelligence Methods: Dedicated to Professor Jacek Żurada*, Cham: Springer International Publishing, 2018, ch. 2, pp. 51–85. doi: 10.1007/978-3-319-67946-4_2.

[9] X. Chen and X. Lin, "Big Data Deep Learning: Challenges and Perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014, doi: 10.1109/ACCESS.2014.2325029.

[10] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J Big Data*, vol. 6, no. 1, p. 27, 2019, doi: 10.1186/s40537-019-0192-5.

[11] P. Branco, L. Torgo, and R. P. Ribeiro, "A Survey of Predictive Modeling on Imbalanced Domains," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 1–50, 2016, doi: 10.1145/2907070.

[12] R. Wang and M.-O. Pun, "Robust Semisupervised Land-use Classification using Remote Sensing Data with Weak Labels," *IEEE Access*, p. 1, 2021, doi: 10.1109/ACCESS.2021.3109989.

[13] Z. Wu, Y. Gao, L. Li, J. Xue, and Y. Li, "Semantic segmentation of high-resolution remote sensing images using fully convolutional network with adaptive threshold," *Conn Sci*, vol. 31, no. 2, pp. 169–184, 2019, doi: 10.1080/09540091.2018.1510902.

[14] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *J Big Data*, vol. 5, no. 1, p. 42, Nov. 2018, doi: 10.1186/s40537-018-0151-6.

[15] W. C. Sleeman IV and B. Krawczyk, "Multi-class imbalanced big data classification on Spark," *Knowl Based Syst*, vol. 212, p. 106598, 2021, doi: 10.1016/j.knosys.2020.106598.

[16] E. Rendón, R. Alejo, C. Castorena, F. J. Isidro-Ortega, and E. E. Granda-Gutiérrez, "Data Sampling Methods to Deal With the Big Data Multi-Class Imbalance Problem," *Applied Sciences*, vol. 10, no. 4, 2020, doi: 10.3390/app10041276.

[17] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The Imbalanced Training Sample Problem: Under or over Sampling?," in *Structural, Syntactic, and Statistical Pattern Recognition*, A. Fred, T. M. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 806–814.

[18] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans Syst Man Cybern*, vol. SMC-2, no. 3, pp. 408–421, 1972, doi: 10.1109/TSMC.1972.4309137.

[19] I. Tomek, "Two Modifications of CNN," *IEEE Trans Syst Man Cybern*, vol. SMC-6, no. 11, pp. 769–772, 1976, doi: 10.1109/TSMC.1976.4309452.

[20] P. E. Hart, "The Condensed Nearest Neighbour Rule," *IEEE Trans Inf Theory*, vol. 14, no. 3, pp. 515–516, 1968, doi: 10.1109/TIT.1968.1054155.

[21] K. Cheng, C. Zhang, H. Yu, X. Yang, H. Zou, and S. Gao, "Grouped SMOTE With Noise Filtering Mechanism for Classifying Imbalanced Data," *IEEE Access*, vol. 7, pp. 170668–170681, 2019, doi: 10.1109/ACCESS.2019.2955086.

[22] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[23] A. Fernández, S. Garc\'{\i}a, F. Herrera, and N. V Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-Year Anniversary," *J. Artif. Int. Res.*, vol. 61, no. 1, pp. 863–905, Jan. 2018, doi: 10.1613/jair.1.11192.

[24] G. Kovacs, "Smote-variants: A python implementation of 85 minority oversampling techniques," *Neurocomputing*, vol. 366, pp. 352–354, 2019, doi: 10.1016/j.neucom.2019.06.100.

[25] F. Ghaseminik, H. Aghamohammadi, and M. Azadbakht, "Land cover mapping of urban environments using multispectral LiDAR data under data imbalance," *Remote Sens Appl*, vol. 21, p. 100449, 2021, doi: 10.1016/j.rsase.2020.100449.

[26] X. Zhong, Y. Quan, W. Feng, Q. Li, G. Dauphin, and M. Xing, "Imbalanced Multi-Class Classification of Hyperspectral Image Based on Smote and Deep Rotation Forest," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 2516–2519. doi: 10.1109/IGARSS47720.2021.9554960.

[27] A. Özdemir, K. Polat, and A. Alhudhaif, "Classification of imbalanced hyperspectral images using SMOTE-based deep learning methods," *Expert Syst Appl*, vol. 178, p. 114986, 2021, doi: https://doi.org/10.1016/j.eswa.2021.114986.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[30] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, Jun. 2004, doi: 10.1145/1007730.1007735.

[31] P. Vuttipittayamongkol, E. Elyan, and A. Petrovski, "On the class overlap problem in imbalanced data classification," *Knowl Based Syst*, vol. 212, p. 106631, 2021, doi: 10.1016/j.knosys.2020.106631.

[32] S. Haykin, *Neural Networks. A Comprehensive Foundation*, 2nd ed. New Jersey: Pretince Hall, 1999.

[33] A. Fawzi, S.-M. Moosavi-Dezfooli, P. Frossard, and S. Soatto, "Classification regions of deep neural networks," *CoRR*, vol. abs/1705.09552, 2017, [Online]. Available: http://arxiv.org/abs/1705.09552

[34] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016, [Online]. Available: http://arxiv.org/abs/1609.04747

[35] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in *Proc. 14th International Conference on Machine Learning*, Morgan Kaufmann, 1997, pp. 179–186.

[36] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002, doi: 10.3233/IDA-2002-6504.

[37] Y. Yan, M. Chen, M. Shyu, and S. Chen, "Deep Learning for Imbalanced Multimedia Data Classification," in *2015 IEEE International Symposium on Multimedia (ISM)*, 2015, pp. 483–488. doi: 10.1109/ISM.2015.126.

[38] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2999–3007. doi: 10.1109/ICCV.2017.324.

[39] J. Yue, L. Fang, H. Rahmani, and P. Ghamisi, "Self-Supervised Learning With Adaptive Distillation for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2021, doi: 10.1109/TGRS.2021.3057768.

[40] E. Pan, X. Mei, Q. Wang, Y. Ma, and J. Ma, "Spectral-spatial classification for hyperspectral image based on a single GRU," *Neurocomputing*, vol. 387, pp. 150–160, 2020, doi: 10.1016/j.neucom.2020.01.029.

[41] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral–Spatial Attention Network for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3232–3245, 2020, doi: 10.1109/TGRS.2019.2951160.

[42] B. Qolomany, "Parameters optimization of deep learning models using Particle swarm optimization," *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017.

[43] Y. Lee, S.-H. Oh, H. K. Song, and M. W. Kim, "Design rules of multilayer perceptrons," in *Defense, Security, and Sensing*, 1992.

[44] R. Alejo, J. M. Sotoca, and G. A. Casañ, "An Empirical Study for the Multi-class Imbalance Problem with Neural Networks," in *Progress in Pattern Recognition, Image Analysis and Applications*, J. Ruiz-Shulcloper and W. G. Kropatsch, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 479–486.

[45] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLoS One*, vol. 10, no. 3, p. e0118432, Mar. 2015, doi: 10.1371/journal.pone.0118432.

[46] M. Liu, M. Dong, and C. Jing, "A modified real-value negative selection detector-based oversampling approach for multiclass imbalance problems," *Inf Sci (N Y)*, vol. 556, pp. 160–176, 2021, doi: 10.1016/j.ins.2020.12.058.

[47] J. Qin, C. Wang, Q. Zou, Y. Sun, and B. Chen, "Active learning with extreme learning machine for online imbalanced multiclass classification," *Knowl Based Syst*, vol. 231, p. 107385, 2021, doi: 10.1016/j.knosys.2021.107385.

[48] J. Du, Y. Zhou, P. Liu, C.-M. Vong, and T. Wang, "Parameter-Free Loss for Class-Imbalanced Deep Learning in Image Classification," *IEEE Trans Neural Netw Learn Syst*, pp. 1–7, 2021, doi: 10.1109/TNNLS.2021.3110885.

[49] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.

[50] J. Carrasco, S. García, M. M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm Evol Comput*, vol. 54, p. 100665, 2020, doi: 10.1016/j.swevo.2020.100665.

[51] A. Bansal and A. Jain, "Analysis of Focussed Under-Sampling Techniques with Machine Learning Classifiers," in *2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2021, pp. 91–96. doi: 10.1109/SERA51205.2021.9509270.

[52] A. N. Tarekegn, M. Giacobini, and K. Michalak, "A review of methods for imbalanced multi-label classification," *Pattern Recognit*, vol. 118, p. 107965, 2021, doi: 10.1016/j.patcog.2021.107965.

[53] E. Kaya, S. Korkmaz, M. A. Sahman, and A. C. Cinar, "DEBOHID: A differential evolution based oversampling approach for highly imbalanced datasets," *Expert Syst Appl*, vol. 169, p. 114482, 2021, doi: 10.1016/j.eswa.2020.114482.

[54] S. Korkmaz, M. A. Şahman, A. C. Cinar, and E. Kaya, "Boosting the oversampling methods based on differential evolution strategies for imbalanced learning," *Appl Soft Comput*, vol. 112, p. 107787, 2021, doi: https://doi.org/10.1016/j.asoc.2021.107787.

[55] R. M. Pereira, Y. M. G. Costa, and C. N. Silla, "Handling imbalance in hierarchical classification problems using local classifiers approaches," *Data Min Knowl Discov*, vol. 35, no. 4, pp. 1564–1621, 2021, doi: 10.1007/s10618-021-00762-8.

[56] R. M. Pereira, Y. M. G. Costa, and C. N. Silla Jr., "MLTL: A multi-label approach for the Tomek Link undersampling algorithm," *Neurocomputing*, vol. 383, pp. 95–105, 2020, doi: 10.1016/j.neucom.2019.11.076.

[57] C. Jia and Y. Zuo, "S-SulfPred: A sensitive predictor to capture S-sulfenylation sites based on a resampling one-sided selection undersampling-synthetic minority oversampling technique," *J Theor Biol*, vol. 422, pp. 84–89, 2017, doi: 10.1016/j.jtbi.2017.03.031.

[58] Z. Li, M. Huang, G. Liu, and C. Jiang, "A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection," *Expert Syst Appl*, vol. 175, p. 114750, 2021, doi: 10.1016/j.eswa.2021.114750.

[59] Q. V Le, "A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks," 2015. [Online]. Available: http://lnnk.in/gadO

[60] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021, doi: 10.1109/JPROC.2020.3004555.