MOX-Report No. 81/2021

Feature Selection for Imbalanced Data with Deep Sparse Autoencoders Ensemble

Massi, M.C.; Gasperoni, F.; Ieva, F.; Paganoni, A.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

Received -; | Revised -; | Accepted -

DOI: xxx/xxxx

ARTICLE

Feature Selection for Imbalanced Data with Deep Sparse Autoencoders Ensemble

Michela C. Massi[1,2] | Francesca Gasperoni[3] | Francesca Ieva[1,2] | Anna Maria Paganoni[1,2]

[1]MOX Laboratory for Modeling and Scientific Computing, Department of Mathematics, Politecnico di Milano
[2]CHDS - Center for Health Data Science, Human Technopole
[3]MRC-Biostatistics Unit, University of Cambridge

Correspondence
*Michela Carlotta Massi, Department of Mathematics, Politecnico di Milano, Via Edoardo Bonardi 9, 20133 Milano, Italy.
Email: michelacarlotta.massi@polimi.it

Summary

Class imbalance is a common issue in many domain applications of learning algorithms. Oftentimes, in the same domains it is much more relevant to correctly classify and profile minority class observations. This need can be addressed by Feature Selection (FS), that offers several further advantages, s.a. decreasing computational costs, aiding inference and interpretability. However, traditional FS techniques may become sub-optimal in the presence of strongly imbalanced data. To achieve FS advantages in this setting, we propose a filtering FS algorithm ranking feature importance on the basis of the Reconstruction Error of a Deep Sparse AutoEncoders Ensemble (DSAEE). We use each DSAE trained only on majority class to reconstruct both classes. From the analysis of the aggregated Reconstruction Error, we determine the features where the minority class presents a different distribution of values w.r.t. the overrepresented one, thus identifying the most relevant features to discriminate between the two. We empirically demonstrate the efficacy of our algorithm in several experiments, both simulated and on high-dimensional datasets of varying sample size, showcasing its capability to select relevant and generalizable features to profile and classify minority class, outperforming other benchmark FS methods. We also briefly present a real application in radiogenomics, where the methodology was applied successfully.

KEYWORDS:
Feature Selection, Imbalanced Data, AutoEncoder, Minority Class Profiling, Ensemble Methods

1 | INTRODUCTION

A well-known problem of many real life applications of statistical models and machine learning algorithms is class imbalance [5]. Examples can be found in many sensitive domains such as medicine [37], especially in case of rare disease classification tasks [22], fraud detection [46], fault detection [54], cyber security [49] and many others [2]. All these domains share the same peculiarity: the importance of correctly identifying and profiling the minority class. In these contexts, a false negative is usually much more expensive than a false positive. A straightforward example comes from the medical field, where a missed diagnosis in many cases is extremely risky for the patient's health and costly for the healthcare system [4, 43].

[0]**Abbreviations:** DSAEE, deep sparse autoencoder ensemble; FS, feature selection; AE, autoencoder; DSAE, deep sparse autoencoder; RE, reconstruction error; ML, machine learning; FSDS, feature selection dataset; CDS, classification dataset; LR, logistic regression; DT, decision tree; SVM, support vector machine; NB, naive bayes; NN nearest neighbor; AUROC, area under the roc curve; RFE, random feature elimination; LT, late toxicity

Moreover, on top of precise classification of minority class observations, domain experts are oftentimes interested in understanding which specific features (i.e. characteristics of their patients, or customers, etc.) should be kept under control or to investigate to drive decisions or invest in future research. The importance of identyfying the discriminant characteristics of the minority class is particularly evident in the clinical field, where an inaccurate feature selection can lead to an inaccurate diagnosis [25]. This observation holds for Genome Wide Association Studies for precision medicine [24], where the clinical interest lies on detecting the traits that are associated to a specific disease [6]. Answering to this question, rather than merely classifying observations, gets harder as the number of features and the non-linearity of their interrelationships rises, driving growth in models' complexity. One way of addressing this need is through Feature Selection (FS) techniques.

In general, FS helps in identifying highly influential features that provide intrinsic information and discriminant property for class separability, while decreasing computational costs, aiding inference and giving better understanding on model representation [17, 45]. However, it has been argued that traditional FS techniques become sub-optimal or even prejudicial to classification effectiveness when the classes are strongly imbalanced [48, 53]. In [48], the authors demonstrate through a simulation study how the overlapping of the classes' distributions after feature selection increases because of the strong bias towards the majority class, hindering classification performance. Therefore, to achieve the advantages granted by FS, a method tailored to address imbalanced settings without affecting classification accuracy is desirable.
Indeed, we argue that a FS robust to class imbalance can address both needs for accurate classification of the underrepresented class and for the identification of the specific pieces of information that are the most relevant for its identification. In other words, by selecting the most informative features to discriminate between classes, such a FS method can serve as a useful tool for the task of *minority class profiling*. In Section 5.5 we will briefly describe a real case study where the methodology presented in this work successfully played this role in a complex research setting. Nonetheless, although FS for imbalanced classification is recently gaining momentum, the number of reported works on the subject is still limited [2]. Few contributions dealt with this multi-faceted problem [22].

For these reasons, in this work we focused on developing a novel FS method tailored to identify relevant features to discriminate the minority from the majority class in strongly imbalanced binary classification settings. In order to accomplish this task, in this paper we propose a filtering algorithm that ranks feature importance on the basis of a Deep Sparse AutoEncoders Ensemble (DSAEE).

From a methodological standpoint, the value provided by our proposal comes from the combination of two aspects: on the one hand, the choice of a particular type of AutoEncoder (AE) [23] as underlying model, on the other, the inclusion of this model within an ensemble algorithm.
Indeed, AEs are Neural Network (NN) models capable of flexibly capturing non-linear relationships among features [21]. These models have been exploited as feature selectors but, to the best of our knowledge, never tailored to class imbalance (cfr. Section 2.2). Here we claim they can be effectively exploited as feature selectors specifically for an imbalanced setting if we consider the duality between imbalanced minority class classification and outlier detection. Indeed, as the minority class is rare w.r.t. the majority one, its observations might be considered outliers w.r.t. the normal population (inliers) constituted by the overrepresented class. AEs were previously recognized as powerful reconstruction-based outlier detection methods [10, 12, 27, 34, 40, 42] that rely on scoring outliers by aggregating the Reconstruction Error (RE) for each observation. In this work, we propose to repurpose this reconstruction-based outlier detection approach to solve the problem of feature selection in imbalanced settings instead. Indeed, we apply an AE trained only on majority class observations to reconstruct both majority and minority classes: from the aggregation of the REs for each feature within each class, we determine which features are associated to the highest differences between the REs of the minority class w.r.t. the majority class - thus identifying the most relevant features to discriminate between the two classes.
However, there exists the risk that a single AE fails in capturing the correlations among features, especially in high dimensional settings [12], and a natural variance in results that might depend on the data, the design of the model and the local search for parameters typical of many Machine Learning (ML) methods. By using an ensemble approach, as the one proposed in this work, and taking a central estimator of the RE, like the mean or the median, this variance is reduced [10, 14]. Nevertheless, in order to make ensemble learning methods work, the individual ensemble components must be adequately diverse [10, 42]. This is achieved in our proposition by designing the algorithm s.t. each ensemble component can capture different aspects of the underliyng majority class distribution. In particular, the novelty of our approach resides in fostering this diversity among

components through (i) a sampling procedure tailored for imbalanced settings that builds different training and test sets to supply to each learner, and (ii) a sparsity constraint imposed on the models.

In light of the above, the contributions of this work are multiple. We enlarge the limited literature on FS tailored to deal with the daunting real-life issue of class imbalance. We do that by presenting an algorithm that repurposes the power of AEs as outlier detectors for reconstruction-based minority class-specific feature selection, which is a novelty for AE-based feature selectors in general. Finally, we robustify the selection thanks to its ensemble approach, designed to foster diversity of components and accuracy on minority class.

The remainder of the paper is organized as follows. In Section 2 we discuss some related works, strenghtening our positioning w.r.t. other approaches; in Section 3 we provide some background on DSAEs, then we describe and discuss the proposed DSAEE algorithm in detail. In Section 4 we describe a simulation study on synthetically generated data, devoted to prove the concepts underlying our proposed approach, while in Section 5 we detail a series of experiments on several benchmark datasets of varying sample size and dimensionality: firstly we validate the good performance of the selected feature subset despite the dimensionality reduction (Section 5.2), then, we compare our proposed methodology with other state-of-the-art and more traditional FS methods (Section 5.3). Additionally, we display some visualizations of the selected features to demonstrate their meaningfulness in discriminating minority from majority class (Section 5.4) and finally we briefly describe an application on real clinical data (Section 5.5). In Section 6 we highlight some relevant considerations on the proposed approach, and conclude with some final remarks and possible extensions.

2 | RELATED WORKS

As stated in the introduction, we aim at presenting a novel FS method tailored to tackle class imbalance. Indeed, the method is designed to select a subset of informative features to reduce the impact of the strong imbalance between minority and majority classes on the classification performance. To frame the position of our proposal from a methodological point of view, in this section we will first describe other works that developed methods to this aim. Then, as we are exploiting AEs as building blocks of our ensemble method for FS, we will report on studies that utilized these models for this task, irrespective of the classes' distribution.

2.1 | Feature Selection for Imbalanced Data

In general, there are three approaches to apply FS algorithms in classification: wrapper, embedded and filter methods [51]. Wrapper methods [31] make the FS revolve around the optimization of the performance of a predetermined classifier: the feature subset that maximises the defined performance metric is selected. In an imbalanced setting, the choice of the optimization metric is crucial. Indeed, among the available examples in the literature, some exploited the area under the ROC curve as a metric to select the best mix of features [11], others the F-measure [2, 33, 51], while in [35] they exploit, among others, a balanced loss function which takes the weighted average of false positives and false negatives. Despite their optimal results in terms of classification accuracy, wrapper methods are generally computationally expensive, and there is no guarantee of reaching a global optimum.

Embedded methods [28] overcome this issue by determining the feature subset autonomously during classifier learning, by including for instance a regularization term in the loss function [38]. However, to the best of our knowledge, no embedded method has been designed specifically to tackle class imbalance. An hybrid embedded and wrapper approach is instead proposed in [32]. Nonetheless, all the aforementioned methods are strictly bounded to a specific classifier.

Filter methods [41] are pre-processing algorithms that measure the usefulness of the feature subset for classification by working on the original data without involving any classifier. They usually rank features' importance on the basis of suitable metrics, some specifically tailored for imbalanced classification problems [13, 48, 53]. Our proposal belongs to this classifier-agnostic type of algorithms.

2.2 | AutoEncoder-based Feature Selection

We will now provide a brief overview of how AutoEncoders (AEs) were employed as feature selectors in the available literature. As mentioned, AEs [23] are a particular class of NNs widely used for learning of data representations [7], dimensionality reduction [23] and outlier/anomaly detection [1, 10, 12, 27, 34, 40, 42]. This powerful representation learning method has been recently exploited for reconstruction-based feature selection as well. For instance, in [9] AEs are exploited as an unsupervised feature selection method, masking input features and using the Reconstruction Error (RE) of masked input features to compute feature weights in a moving average manner. In [21] the authors combine AE regression and a weight penalization on the input layer: feature importance is then derived from the value of the weights associated to each feature. Another sparsity-based unsupervised approach can be found in [16] and [50]. Finally, in the most recent work in [8], the authors propose the Concrete AutoEncoder Feature Selector (CAEFS), that exploits the Concrete distribution to differentiate through the reconstruction loss and selects input features to minimize it.
All these approaches share an unsupervised setting and have demonstrated their potential as feature selectors against other state of the art techniques. Nonetheless, they all train one AE model only, incurring in the risks discussed in Section 1. Moreover, they all are FS methods designed for balanced classification. This balanced selection of features was argued potentially harmful in strongly imbalanced settings [48, 53]. What distinguishes our DSAEE from the available examples of AE-based feature selectors, is the ensemble approach to the problem - where each of the AE is one of a set of weak learners - and the tailoring of each model's training procedure inspired by outlier detection methods, to approach specifically imbalanced datasets.

3 | DSAE ENSEMBLE (DSAEE) FOR MINORITY CLASS FEATURE SELECTION

In Section 3.2 we provide some background on the DSAE components and we detail the regularization we impose on the models to foster the diversity among each component. In Section 3.3 we detail how the proposed algorithm encapsulates each component into a tailored training procedure to identify the most relevant features to discriminate minority class in imbalanced settings.

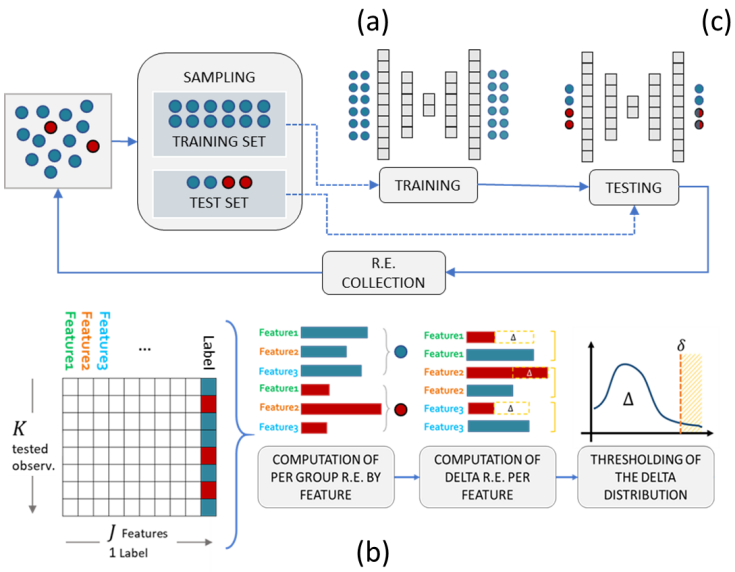
3.1 | Problem Statement

3.2 | Background: AutoEncoders and Deep Sparse AutoEncoders

An AE [23] is a NN trained to attempt to copy its input to its output. Let the matrix $\mathbf{X} \in \mathbb{R}^{N\times J}$ be the input data, $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ set of N training vectors i ($i \in \{1, ..., N\}$), characterized by J features. The shallow version of an AE is constituted by an input layer with J nodes, a hidden layer with H (with H usually smaller than J) nodes that describes a *code* used to represent the input, and an output layer of size J. The network can be seen as constituted by two parts: an encoder and a decoder. The encoder function $\mathbf{h}_i = f(\mathbf{W}\mathbf{x}_i + \mathbf{b})$, encodes each input vector $\mathbf{x}_i$ into an encoded version of itself of size H. Here f is usually non-linear and is referred to as *activation function*, $\mathbf{W} \in \mathbb{R}^{H\times J}$ is called *weight matrix* and $\mathbf{b}$ is a H-dimensional *bias* vector. The decoder maps back the encoded vector to the J-dimensional space in most cases using a squashing non-linear function $\hat{\mathbf{x}}_i = g(\mathbf{W}'\mathbf{h}_\mathbf{i} + \mathbf{b}')$, with parameters $\mathbf{W}' \in \mathbb{R}^{J\times H}$ and $\mathbf{b}' \in \mathbb{R}^{J}$. The model is trained through gradient descent of the loss function $L(\mathbf{x}, \hat{\mathbf{x}})$; where L is typically the Mean Squared Reconstruction Error (MSRE), i.e. the mean squared Euclidean distance between the input values and the reconstructed values for each observation. Each training observation $\mathbf{x}_i$ is thus mapped to a corresponding $\mathbf{h}_i$ which is then mapped to a reconstruction $\hat{\mathbf{x}}_i$ s.t. $\hat{\mathbf{x}}_i \approx \mathbf{x}_i$. In general, we can define an AE as a map $\phi(\mathbf{x}_i) : \mathbb{R}^J \to \mathbb{R}^J$, such that $\phi(\mathbf{x}_i) = g(\mathbf{W}' f(\mathbf{W}\mathbf{x}_i + \mathbf{b}) + \mathbf{b}')$ and the optimal representation of $\mathbf{x}_i$, $\hat{\mathbf{x}}_i$, is given by:

$$\hat{\phi}(\mathbf{x}_i) = \arg\min_{\phi} ||\mathbf{x}_i - \phi(\mathbf{x}_i)||^2 = \arg\min_{\phi} \sum_{j=1}^{J} (\mathbf{x}_{ij} - \phi(\mathbf{x}_{ij}))^2.$$

To expand the shallow network to a *deep* version, the formulation is similar, with the output of one layer being the input of the following layer. In this case, the map ϕ will be the results of multiple compositions.
Usually, AEs are built with constraints that force them not only to replicate the input, but to learn effective representations of such input in the hidden layer. One way to obtain useful representations from the autoencoer is to introduce sparsity in the code layer (Sparse AutoEncoders - SAE) by imposing a regularization term in the loss function. In order to do that, the model includes a sparsity penalty $\Omega(\mathbf{h})$ on the hidden layer (or the most internal layer in case of deep architectures) $\mathbf{h}$, additionally to



(c)

Algorithm 1: DSAEE for Minority Class Feature Selection

Input: $\mathbf{X} \in \mathbb{R}^{N \times J}$ data matrix with N observations and J features, $\mathbf{y} \in \mathbb{R}^{N}$ target vector
Output: Features Set F

$\mathbf{X}_O, \mathbf{X}_M \leftarrow \{\mathbf{x}_i | y_i = 1\}, \{\mathbf{x}_i | y_i = 0\}$
$\mathbf{y}_O \leftarrow \{\mathbf{y}_i | \mathbf{x}_i \in \mathbf{X}_O\}$
$minority\_sample\_size \leftarrow$ number of rows in $\mathbf{X}_O$
Set $\mathbf{Q}$ `// empty array`
repeat
  $\mathbf{X}_{M,test} \leftarrow$ Sample $minority\_sample\_size$ observations from $\mathbf{X}_M$;
  $\mathbf{y}_{M,test} \leftarrow \{y_i | \mathbf{x}_i \in \mathbf{X}_{M,test}\}$;
  $\mathbf{X}_{test}, \mathbf{y}_{test} \leftarrow concatenate(\mathbf{X}_O, \mathbf{X}_{M,test}), concatenate(\mathbf{y}_O, \mathbf{y}_{M,test})$;
  $\mathbf{X}_{train} \leftarrow \{\mathbf{x}_i \in \mathbf{X}_M | \mathbf{x}_i \notin \mathbf{X}_{test}\}$;
  Train DSAE on $\mathbf{X}_{train}$;
  $\hat{\mathbf{X}}_{test} \leftarrow$ DSAE reconstruction of $\mathbf{X}_{test}$;
  $\mathbf{R} \leftarrow L(\mathbf{X}_{test}, \hat{\mathbf{X}}_{test})$;
  Label $\mathbf{R}$ with $\mathbf{y}_{test}$;
  $\mathbf{Q} \leftarrow concatenate(\mathbf{Q}, \mathbf{R})$
until $number\ of\ iterations \leq B$;
$\mathbf{Q}_{maj}, \mathbf{Q}_{min} \leftarrow \{l_p \in \mathbf{Q} | y_p = 0\}, \{l_p \in \mathbf{Q} | y_p = 1\}$
$l_{min}, l_{maj} \leftarrow Column\ mean\ of\ \mathbf{Q}_{min}, Column\ mean\ of\ \mathbf{Q}_{maj}$
$\Delta \leftarrow l_{min} - l_{maj}$
Define threshold δ on the distribution of Δ values.
$F \leftarrow \{j | \Delta_j > \Delta_\delta, j \in \{1, .., J\}\}$

FIGURE 1 Training and FS of DSAEE algorithm. Panel (a) is a schema of the sampling procedure, repeated for each ensemble component. Blue and red dots represent majority and minority class' observations respectively. Panel (b) represents the steps of the algorithm from the concatenated $\mathbf{Q}$ matrix of RE to the feature selection. Blue bars represent average RE by feature for majority class observations, while red bars represent average RE on minority class. Finally, panel (c) reports the pseudo-code of the whole DSAEE algorithm.

the reconstruction error:

$$L_i = L(\mathbf{x}_i, \hat{\mathbf{x}}_i) + \Omega(\mathbf{h}_i).$$

The regularization can take various forms. In a deep architecture (Deep Sparse AutoEncoder - DSAE), let us consider $\mathbf{h}_i^{(l)}$ as the activation of the most internal hidden layer (l) for the i-th observation vector $\mathbf{x}_i$, i.e. the value of the function $\mathbf{h}_i^{(l)} = f^{(l)}(\mathbf{W}^{(l)}\mathbf{h}_i^{(l-1)} + \mathbf{b}^{(l)})$. One way of obtaining a sparse representation is to add a penalty term that penalizes the L_1 norm of the vector $\mathbf{h}_i^{(l)}$ for each observation i, controlled by a parameter λ, i.e.

$$L_i = L(\mathbf{x}_i, \hat{\mathbf{x}}_i) + \lambda |\mathbf{h}_i^{(l)}|. \tag{1}$$

The parameter λ can be optimized through grid search or can be arbitrarily chosen in the design phase of the model.
This penalization term forces the model to *activate* the minimum number of hidden nodes to reconstruct the input. Paired with the input sampling described below, it increases the diversity among each learner in the ensemble. Moreover, it reduces the need for tailored choices or expensive optimization to define the proper architecture.

3.3 | The Ensemble Algorithm

Let us consider the binary supervised learning setup with a training set of N (input, target) pairs $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$, where y_i is the target that takes values is $\{0, 1\}$ and $\mathbf{X} \in \mathbb{R}^{N \times J}$ is the input matrix. We consider the supervised learning to be imbalanced, thus the number of observations in the minority class ($O = \{\mathbf{x}_i | y_i = 1\}$) is relevantly smaller than the number of observations in the majority class ($M = \{\mathbf{x}_i | y_i = 0\}$). We assume that our observations $\mathbf{x}_i$ are the realizations of two random vectors. Specifically, the observations from the minority class are the realizations of $\mathbf{X}_i | Y_i = 1 \sim \mathcal{D}_1$, while the ones from the majority class are the realizations of $\mathbf{X}_i | Y_i = 0 \sim \mathcal{D}_0$. $\mathcal{D}_0$ and $\mathcal{D}_1$ are two unknown conditional multivariate probability distributions in $\mathbb{R}^J$. The idea behind this project is that it is possible to distinguish between $\mathcal{D}_0$ and $\mathcal{D}_1$ thanks to a determined set of features F, with $|F| < J$ (from now on the notation $|\cdot|$ will represent the cardinality of a set) and our final objective consists in identifying this set F.

With the intention of building an ensemble of B different *learners* from which to aggregate information to rank features, we first develop a tailored sampling procedure, inspired by the *outlier detection* approaches, to train each learner on a different sample of data selected with the rationale detailed in the following, and schematized in Figure 1 (a).

In particular, we define $\mathbf{X}_O \in \mathbb{R}^{|O|\times J}$ as the features related to the minority class and $\mathbf{X}_M \in \mathbb{R}^{|M|\times J}$ as the ones associated to the majority class. From $\mathbf{X}_O$ and $\mathbf{X}_M$ and the respective outcomes $\mathbf{y}_O$, and $\mathbf{y}_M$ we generate a training set $\mathbf{X}^b_{train}$ and a test set $\mathbf{X}^b_{test}$, where $b \in \{1, \dots, B\}$. Each test set contains $P = 2|O|$ data points, including all the minority class observations, and an equal number of majority ones randomly drawn from M. The training set is instead composed by the majority class data excluded from the test set.

This structure of the two datasets allows us to train each DSAE learner in an unsupervised fashion only on the overrepresented population, and to test their performance when facing both majority and minority class examples, so that we can compare the RE made on the two populations. The rationale behind this sampling procedure is based on the fact that DSAEs trained to reconstruct *normal* observations only (i.e. the majority class) will make higher RE when tested on *outlier* observations (i.e. minority class examples) never experienced during training. Following the notation in Section 3.2 and considering the different generating mechanisms behind the minority and majority populations, we expect that the optimal representations of $\mathbf{X}_i|Y_i = 0$ and $\mathbf{X}_i|Y_i = 1$ are different and we define them as $\hat{\phi}_0(\cdot)$ and $\hat{\phi}_1(\cdot)$, respectively. Training the DSAE on $\mathbf{X}^b_{train}$ by minimizing the loss function formulated in (1) leads to the estimate of $\hat{\phi}_0(\mathbf{X}^b_{train}|\mathbf{Y}^b_{train} = \mathbf{0})$. This representation is optimal, since we are sampling all units from the majority class. This consideration does not hold for the test set $\mathbf{X}^b_{test}$. Indeed, $\hat{\phi}_0(\cdot)$ is the optimal representation only for half of the data point, while it is suboptimal for those data points selected from the minority class. By definition, if a data point i belongs to the minority class, then Equation 2 holds, while if a data point r belongs to the majority class, then Equation 3 holds.

$$||(\mathbf{X}^b_{test,i}|Y^b_{test,i} = 1) - \hat{\phi}_0(\mathbf{X}^b_{test,i}|Y^b_{test,i} = 1)||^2 \geq ||(\mathbf{X}^b_{test,i}|Y^b_{test,i} = 1) - \hat{\phi}_1(\mathbf{X}^b_{test,i}|Y^b_{test,i} = 1)||^2 \tag{2}$$

$$||(\mathbf{X}^b_{test,r}|Y^b_{test,r} = 0) - \hat{\phi}_0(\mathbf{X}^b_{test,r}|Y^b_{test,r} = 0)||^2 \leq ||(\mathbf{X}^b_{test,r}|Y^b_{test,r} = 0) - \hat{\phi}_1(\mathbf{X}^b_{test,r}|Y^b_{test,r} = 0)||^2 \tag{3}$$

Furthermore, we expect that for the two generic data points i and r defined above Equation 4 is satisfied.

$$\sum_{j=1}^{J}((X^b_{test,ij}|Y^b_{test,i} = 1) - \hat{\phi}_0(X^b_{test,ij}|Y^b_{test,i} = 1))^2 \geq \sum_{j=1}^{J}((X^b_{test,rj}|Y^b_{test,r} = 0) - \hat{\phi}_0(X^b_{test,rj}|Y^b_{test,r} = 0))^2 \tag{4}$$

The most discriminating set of features, F, between the majority and the minority classes are the ones that contribute the most to the RE difference (elements in the sum reported in Equation 5).

$$\sum_{j=1}^{J}((X^b_{test,ij}|Y^b_{test,i} = 1) - \hat{\phi}_0(X^b_{test,ij}|Y^b_{test,i} = 1))^2 - ((X^b_{test,rj}|Y^b_{test,r} = 0) - \hat{\phi}_0(X^b_{test,rj}|Y^b_{test,r} = 0))^2 \tag{5}$$

To get a robust ranking procedure, we extend this reasoning from a simple comparison between the REs associated with two data points towards all the RE evaluated on the B test sets. The REs evaluated for a specific feature j for a specific data point p of the b-th test set, $\boldsymbol{l}^b_p$, is defined in Equation 6:

$$\boldsymbol{l}^b_p = \{((X^b_{test,pj}|Y^b_{test,pj}) - \hat{\phi}_0(X^b_{test,pj}|Y^b_{test,pj}))^2\}^J_{j=1}. \tag{6}$$

The RE made on the whole test set b are collected in the RE matrix $\mathbf{R}^b \in \mathbb{R}^{\mathrm{P}\times\mathrm{J}}$, whose p-th row is given by $\boldsymbol{l}^b_p$ and $p \in \{1, \dots, P = 2|O|\}$.

Then we concatenate by rows the B matrices $\mathbf{R}^b$, building the final RE matrix $\mathbf{Q} = \{\mathbf{R}^b\}^B_{b=1} \in \mathbb{R}^{\mathrm{PB}\times\mathrm{J}}$, where PB is the total number of tested observations and J is the number of features. For a schema of the algorithm described in the following, refer to Figure 1 (b).

In order to select the most representative features to discriminate between minority and majority class, we subdivide the rows of $\mathbf{Q}$ in two matrices: one composed by minority class RE (see Equation 7) and the other by majority class RE (see Equation 8).

$$\mathbf{Q}_{min} = \{((X_{test,pj}|Y_{test,pj} = 1) - \hat{\phi}_0(X_{test,pj}|Y_{test,pj} = 1))^2\}^{j=\{1,..,J\}}_{p=\{1,..,T\}}. \tag{7}$$

$$\mathbf{Q}_{maj} = \{((X_{test,pj}|Y_{test,pj} = 0) - \hat{\phi}_0(X_{test,pj}|Y_{test,pj} = 0))^2\}^{j=\{1,..,J\}}_{p=\{1,..,T\}}. \tag{8}$$

According to the model design, both $\mathbf{Q}_{maj}$ and $\mathbf{Q}_{min}$ belong to $\mathbb{R}^{T\times J}$, where $T = |O|B$ is the number of both minority and majority class examples, after splitting $\mathbf{Q}$ in two. . From these matrices we can estimate the vectors of average RE per feature j per class: $\boldsymbol{l}_{min}$ and $\boldsymbol{l}_{maj}$, both belonging to $\mathbb{R}^J$, where each element is computed as

$$l_{j,min} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{Q}_{tj,min}, \tag{9}$$

$$l_{j,maj} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{Q}_{tj,maj}, \tag{10}$$

Once we have computed the class specific average REs per feature, we can proceed to the feature selection by studying how the RE of each feature varies between classes. We can accomplish this goal by extending the concept expressed in Equation 5 to Equation 11, where we evaluate $\boldsymbol{\Delta}$ as the J-elements vector resulting from the difference of the average RE on the minority class, $\boldsymbol{l}_{min}$, and the one on the majority class, $\boldsymbol{l}_{maj}$.

$$\boldsymbol{\Delta} = \boldsymbol{l}_{min} - \boldsymbol{l}_{maj}. \tag{11}$$

Finally, we are able to identify the set of discriminating features F, by *ranking* the elements of $\boldsymbol{\Delta}$. Indeed, higher values of Δ_j are associated to those features that have been reconstructed more accurately by the DSAE on the majority class (low RE), but are not reconstructed accurately on the minority class (high RE). Furthermore, it is impossible that we incur in a degenerate case, such as all RE equal to 0, because of the bottleneck and the sparsity constraint imposed in each DSAE.

To identify an exact set F we need to define a threshold $\delta \in (0, 1)$, as it is often required in FS ranking method. Specifically, Δ_δ is the δ-th quantile evaluated on the distribution of $\{\Delta_j\}_{j=1}^{J}$. We therefore select all those features j whose average RE difference is above the pre-defined threshold:

$$F = \{j|\Delta_j > \Delta_\delta, j \in \{1, .., J\}\}. \tag{12}$$

From the original dataset $\mathbf{X}$ we can extract a subset of features whose size can be tuned according to the problem at hand to either analyze per se or feed to any classifier.

There is an inverse relation between δ and the number of selected features: the higher the δ, the lower the number of selected features.

Algorithm 1 in Figure 1 (c) reports the pseudo-code of the whole FS procedure.

3.4 | Computational Complexity

Each DSAEE component has a complexity $O(nwe)$ dependent on n (the number of observations in the data matrix), w (the number of weigths in the network) and e (the number of *epochs*, or iterations in the training).

The complexity of the training of an Ensemble of DSAEs becomes $\sim O(Bnwe)$, growing linearly with the number of B trained models. Both the number of B employed components and the architectural choices impacting w and e can be optimized to reduce training time and improve results as well. Moreover, the ensemble training can be easily parallelized, thus significantly cutting training time.

4 | SIMULATION STUDY

In order to verify the hypothesis underlying our approach we run an extensive simulation study. In particular we were interested in verifying (i) the capability of the DSAEE to induce and capture two significantly separated distributions of average RE for majority and minority classes and (ii) the consequent usefulness of $\boldsymbol{\Delta}$ distribution to select relevant features to separate the two. Additionally, we aimed at testing (iii) the relevance of the sparsity constraint in determining the separation between the aforementioned distributions and finally (iv) the robustness of this approach to the complexity of the task, both in terms of class separability and class imbalance.

In this section we will first describe the defined simulation setting and then detail the results of the experiments we run for the validation of the aforementioned aspects.

4.1 | Experimental Setting

Simulated Data

To validate the assumptions underlying the DSAEE approach to feature selection, while justifying the application of a complex non-linear model (i.e. the AE components) to the task, we needed a generative model for our simulated data that reproduced complex multi-dimensional relationships among features. To this aim, we relied on the generative principles designed by Guyon *et al.* [19] to generate simulated data in NIPS2003 Feature Selection Challenge, adapted and implemented within Python scikit-learn library[1]. Specific details on the generative process can be found in [19] and in Appendix A.1.

In a nutshell, the data generating algorithm allows to define a number of informative features ($J_I = |F_{Inf}|$) and an hyparameter s, hereby defined as *class separation*, defining how far apart the classes lie in the multidimensional space they live in. In particular, for each class, J_I features are drawn independently from $N(0, 1)$ and then randomly linearly combined within each class to add covariance. The clusters are then placed on the vertices of a J_I-dimensional hypercube, with sides of length proportional to s. Additionally, further sets of features can be defined to increase the complexity of the FS task. Indeed, the total number of features for each simulated dataset is defined as $J_{SIM} = J_I + J_R + J_U$, where J_R is the number of Redundant Features (i.e. linear combination of the F_{Inf} set), and J_U useless features drawn at random.

In particular, for all the experiments described here we generated datasets with $N = 1500$ and $J_{SIM} = 150$, of which $J_I = 25$ and $J_R = 25$. Notably, all datasets were characterized by strong imbalance: 95% of the generated data belongs to majority class, 5% (75 observations only) to minority class. The class separation hyperparameter s was set according to the specific experimental goal.

Implementation details and evaluation metrics

To test our assumptions we defined a simple and fixed *toy architecture* across all experiments, to make them comparable. Each AE component's encoder has one 100-nodes hidden layer, followed by a bottleneck layer of 50 nodes. This choice of a fixed AE structure is meant to demonstrate the robustness of our method to poorly optimized architectures, aided by the sparsity constraint. Specific hyperparameter details for each experiment described in this section can be found in in Appendix A.2 and in Table B7 therein.

To quantify and evaluate the separation of the class-specific RE distributions, we decided to adopt non parametric methods to avoid strict assumptions on the distributions of RE. In particular, we exploited the Wasserstein Distance [44] to quantify the difference in shape of the two empirical distributions, and the Wilcoxon paired signed-rank test, to evaluate whether the two related samples (i.e. same features' average RE observed in two groups) come from the same distribution (two-sided test), or there is a stochastic order between the two distributions (one-sided test). Note that this test is particularly meaningful in our context as it assumes there is information in the magnitudes and signs of the differences between two samples (i.e. $\mathbf{\Delta}$ in our context) and studies the distribution of those differences treated as a rank: it is indeed by ranking features on the basis of $\mathbf{\Delta}$ that we select the most relevant ones. For this study we performed both two sided and one sided tests and reported their statistics and p-values.

Finally, we introduce a metric named Feature Selection Performance (FSP) to evaluate the capability of the algorithm to produce a meaningful ranking of features and select the most informative features to discriminate the two classes first. Indeed, FSP is computed as $FSP_\delta = |IRF|_\delta / |F|_\delta$, where $|IRF|_\delta$ is the total number of Informative and Redundant Features selected with the threshold δ, and $|F|_\delta$ is the total number of selected features with the same threshold. Note that in this *useful-to-selected* ratio, both informative and redundant features are considered useful, since they both carry information to separate the classes and there is no point in preferring one over the other for any downstream task. On the other hand, this metric quantifies the precision in discarding noisy features.

4.2 | Distribution of $\mathbf{\Delta}$ elements and Relevant Feature Selection

To validate points (i) and (ii) as mentioned at the beginning of this section, we generated 10 independent datasets with $s = 2$, and 10 with $s = 0.1$. We assigned to all J_I features the first 25 positions in the dataset, followed by the 25 J_R features. For each dataset, we trained and tested $B = 20$ DSAE components and collected the RE.

For illustrative purposes, in Figure 2 we plot the distribution of the average RE of the two classes ($\boldsymbol{l}_{min}$ and $\boldsymbol{l}_{maj}$) and the distribution of the respective $\mathbf{\Delta}$ for the first simulated datasets, for both values of s. While $s = 2$ corresponds unsurprisingly

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html

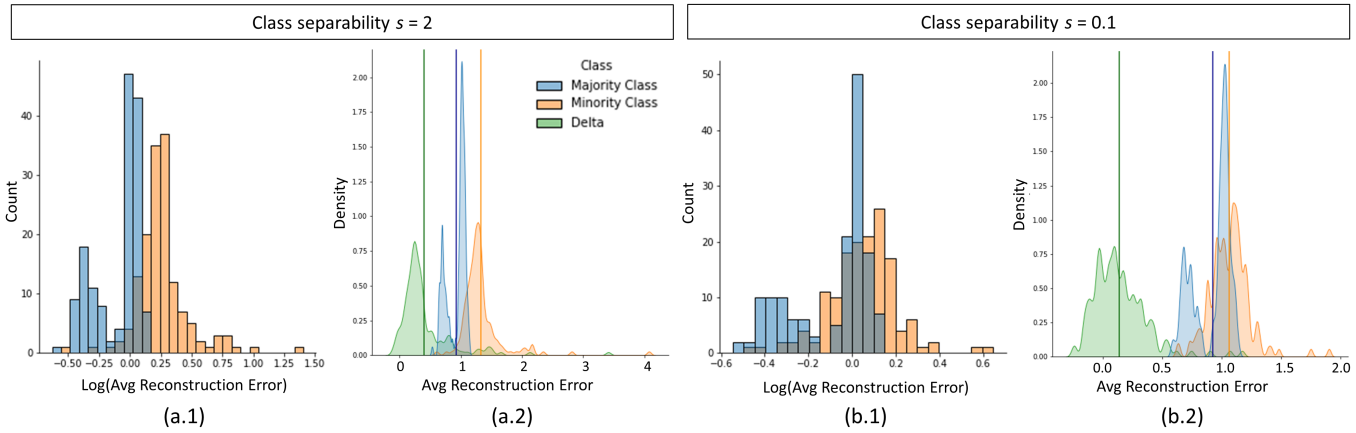


FIGURE 2 Distributions of the average RE associated to the majority class ($\boldsymbol{l}_{maj}$) in blue and to the minority class ($\boldsymbol{l}_{min}$) in orange, for the experiments described in Section 4.2. Panel (a.1) reports the histogram of log(average RE), while (a.2) displays average RE density functions of majority class (blue), minority class (orange) and $\boldsymbol{\Delta}$ distribution (green), for the first of the 10 datasets generated with $s = 2$. Vertical lines represent the mean. Panels (b.1) and (b.2) display the same results for $s = 0.1$

to a larger distance between the classes' RE distributions, minority class average RE ($\boldsymbol{l}_{min}$) distributions have a longer right tail in both cases nonetheless. Moreover, note that majority class average RE ($\boldsymbol{l}_{maj}$) distributions are bimodal, as their RE is affected by the large amount of noise and useless features ($J_U = 100$) in the dataset. A first lower peak should correspond to the 50 informative features, while the higher peak is induced by the larger pool of 100 noisy features. Indeed, to maximize reconstruction accuracy, the AE is induced to learn how to reconstruct informative features first, as they carry all the needed information to reconstruct the characterizing traits of the class it is trained on (i.e. majority class observations). This behavior leads to the desired higher values of Δ_j, associated with j-th informative covariate.
For the datasets displayed in Figure 2 , Wasserstein Distance was 0.407 $s = 2$ and 0.395 for $s = 0.1$, with $W_{one-sided}$ equal to 11,244 and 11,214 respectively, both with a p-value close to zero. These results support our claim on DSAEE's capability to induce significantly different RE distributions, irrespective of the complexity of the task. Wilcoxon tests' results and Wasserstein distances for all the 10 datasets of each experiment can be found in Appendix A.3 in Tables A2 and A3 .

Once verified that the DSAEE can induce class separability in terms of RE distributions, we seek to validate the use of quantile thresholding on the $\boldsymbol{\Delta}$ distribution to select the most relevant features. To this aim, we pick again the first generated datasets, we compute the two $\boldsymbol{\Delta}$ and we report in Table 1 the FSP metric for a range of δ thresholds. Note that in the easiest setting ($s = 2$) up to the low threshold value of $\delta = 0.75$ the DSAEE picks only meaningful features, and for $s = 0.1$ the algorithm keeps a very high useful-to-selected ratio nonetheless.

4.3 | On Sparsity of the Hidden Layer

Having demonstrated our hypotheses on class-specific RE and $\boldsymbol{\Delta}$ distributions, we aimed at evaluating the value added by the sparsity constraint on the hidden layer of the AE to the task of separating minority and majority average REs. Indeed, we claim that a proper level of sparsity in the model allows for better separation of the distributions, while reducing the need for an extensive architecture optimization of the AE. To do that, we kept the same *toy architecture* and run a series of experiments, varying the λ parameter (i.e. the weight associated to the L1 constraint). In particular, for each value of λ, $\lambda \in [0, 1]$, we generated 10 independent datasets with class separation $s = 1$, and we collected all the aforementioned metrics. In Table 2 are reported the results aggregated across the 10 datasets, additional metrics can be found in Appendix A.3 in Table A4 .
Note that despite the λ value, the Wilcoxon tests confirm the separation of the two classes in terms of RE. However, both null regularization (i.e. $\lambda = 0$) and strong regularization (w.r.t. average RE values, thus impacting the optimization significantly, i.e. $\lambda = 1$) provide sub-optimal results. Conversely, a peak can be noted for the following values of λ: $\{0.001, 0.05, 0.1\}$, supporting

	Class separability $s = 2$			Class separability $s = 0.1$		
δ	$\|F\|$	$\|IRF\|$	FSP [%]	$\|F\|$	$\|IRF\|$	FSP [%]
0.6	60	41	68.33	60	41	68.33
0.7	45	37	82.22	45	37	82.22
0.75	38	34	89.47	38	33	86.84
0.8	30	30	100	30	29	96.67
0.85	23	23	100	23	22	95.65
0.9	15	15	100	15	14	93.33
0.95	8	8	100	8	8	100
0.99	2	2	100	2	2	100

TABLE 1 Feature selection results of the first simulated dataset for $s = \{2, 0.1\}$, for different δ thresholds. $|F|$ is the total number of features selected; the column $|IRF|$ reports the total number of selected features that are either informative or redundant; the column *FSP* reports the useful-to-selected ratio.

	d_{Wass}		$W_{\text{one-sided}}$log(p-val)	
λ	mean	std	mean	std
0	0.182	0.031	-34.491	6.143
$1e^{-06}$	0.182	0.032	-34.044	5.985
$1e^{-05}$	0.179	0.035	-33.953	5.915
$1e^{-04}$	0.185	0.035	-36.089	6.709
0.001	**0.205**	**0.032**	**-43.535**	**6.347**
0.05	**0.248**	**0.04**	**-43.687**	**6.794**
0.1	**0.255**	**0.055**	**-37.225**	**8.953**
0.25	0.165	0.016	-3.63	2.126
0.5	0.166	0.017	-3.241	2.089
1	0.164	0.018	-3.602	2.446

TABLE 2 Average RE distributions distances (Wasserstein) and Wilcoxon one-sided log(p-values) for varying $L1$ penalization. Each row represents aggregated results over 10 independently generated datasets.

	d_{Wass}		$W_{\text{one-sided}}$log(p-val)	
s	mean	std	mean	std
0.01	0.158	0.024	-30.583	4.412
0.1	0.154	0.023	-29.44	4.563
0.3	0.162	0.021	-31.916	3.182
0.5	0.178	0.025	-34.926	4.595
0.7	0.202	0.034	-39.363	6.634
1	0.247	0.047	-47.598	7.132
1.5	0.352	0.075	-55.35	3.738
2	0.486	0.108	-58.713	1.855
3	0.783	0.165	-59.697	0.076
5	1.271	0.168	-59.72	0.019
10	2.133	0.292	-59.726	0.006

TABLE 3 Average RE distributions distances (Wasserstein) and Wilcoxon one-sided log(p-values) for varying class separability (s). Each row represents aggregated results over 10 independently generated datasets.

our hypothesis that a properly balanced regularization does affect the separability of RE distributions, pushing minority class average RE distribution towards higher error values.

4.4 | On Robustness to Class Separability and Class Imbalance

Real world data may be arbitrarily complex and majority and minority classes may get extremely hard to separate, once validated our assumptions on the proposed approach, we aimed at testing the robustness of the DSAEE to class separability. To do that we exploited once again our simulation framework, generating 10 independent datasets for each value of s in a wide range from 0.01 (i.e. almost overlapping classes) to 10, setting λ to 0.05. In Table 3 we report all the tested s values and the resulting Wasserstein Distance metrics and Wilcoxon test log(p-values). Further details and Two-sided test results can be found in Appendix A.3 in Table A5 , aggregated across 10 datasets. Irrespective of inter-class distance, the distribution of minority class RE remains

skewed to the right w.r.t. majority class' distribution. Moreover, all measures behave as expected, growing almost monotonically as classes are generated further apart (see Table 3).
Finally, we aimed at testing the DSAEE robustness to class imbalance. Note that all the analyses described so far were performed in a rather imbalanced setting, with minority class observations representing the 5% of each dataset only. However, in order to prove the applicability of our proposal on extreme class imbalance, we designed a final experiment with a range of minority class percentages. In particular, for each majority-minority class splits in [0.9-0.1, 0.95-0.05, 0.97-0.03, 0.99-0.01] we generated 10 datasets, we measured the usual metrics on the average RE distributions, we performed feature selection and evaluated FSP by setting $\delta = 0.9$. To generate all datasets we set $s = 1$ and we trained the DSAEE running $\boldsymbol{B} = 15$ components. All results are reported in Table A6 in Appendix A.3. Notably, the class-specific distributions are consistently different, and the useful-to-selected features ratio remains extremely high. Indeed, with a threshold of $\delta = 0.9$ the algorithm selects 15 features, on average all informative up to minority class sample size percentage of 1% (i.e. 15 observations): in this case 0.89 ± 0.09 are informative to separate the classes.

5 | EXPERIMENTS WITH BENCHMARK AND REAL DATA

Besides validating our assumptions on simulated data, we aimed at studying the performance of our proposed algorithm on datasets mimicking real life scenarios and complexities. However, in this case we had no access to a precise definition of the informative features set, therefore we focused on the classification performance of the selected subset, another fundamental aspect for any FS method.
In particular, we were interested in testing the capability of our algorithm to select even extremely small subsets of features while keeping the classification performance sufficiently high, especially on the minority class. This evaluation was carried out in settings of varying dimensionality and sample size (see Section 5.2). Moreover, we compared the performance of our method against some benchmark FS algorithms (Section 5.3) and finally, we investigated in an interpretable and visual way the meaningfulness of the selected features and their capability to provide useful insights to discriminate between minority and majority classes (Section 5.4). To conclude, we also provide a brief description of a real data application in the challenging field of radiogenomics (see Section 5.5). Through this analysis, we highlight the relevant impact that we are bringing in terms of minority class profiling in complex real-life research scenarios.

5.1 | Datasets and Performance Measures

For all the aforementioned numerical experiments we decided to adopt freely distributed datasets to make results accessible and reproducible. Moreover, some peculiar characteristic of each of the exploited data allowed us to showcase different aspects of our algorithm and discuss its potential when applied to multifaceted scenarios. Note that the datasets exploited in our experiments were not originally imbalanced and in most cases they were meant for multiclass classification problems. As a consequence, a preliminary subsetting of the chosen data was conducted. In the following, we will list the adopted datasets and describe in details the dataset-building choices we made for each of them.
For all datasets, we selected one of the classes as the majority class, and we undersampled another class to represent the minority category. From the derived datasets, we extracted one subset on which we applied our feature selection method (*Feature Selection* DataSet - FSDS), while the remaining was held out to evaluate the classification accuracy of the selected features (*Classification* DataSet, CDS). In Table 4 we report all datasets, their composition, and the type of experiment they were exploited for.

1. **ISOLET** [15] (number of observations $N = 370$; number of features $\boldsymbol{J} = 617$). It consists of preprocessed speech data of people pronouncing the names of the letters in the English alphabet, and is widely used as a benchmark in the feature selection literature. Each feature is one of the 617 quantities produced as a result of the preprocessing. We chose class 'A' as the majority class, and 'B' as the minority one. Given the small number of observations available per class, this dataset allowed us to test the applicability of our algorithm in high dimensionality and small sample size settings.

2. **GISETTE** [18] ($N = 3,300$; $\boldsymbol{J} = 5,000$) This dataset was built for NIPS2003 feature selection challenge. The whole dataset contained 6,000 observations equally split between classes, with 5,000 features (50% of which are probes with no predictive power). We created 5 datasets including all 3,000 majority class observations and 300 randomly sampled minority class observations (9.05%), and we splitted them into FSDS and CDS according to a 75/25 ratio.

DATASET	CLASS	FSDS - N (%)	CDS - N (%)	EXPERIMENTS
ISOLET	'A'	225 (81.9%)	75 (80.7%)	FEATURE SUBSET PERFORMANCE
	'B'	52 (18.1%)	18 (19.3%)	BENCHMARK
GISETTE	'0'	2,250 (90.91%)	750 (90.91%)	FEATURE SUBSET PERFORMANCE
	'1'	225 (9.09%)	75 (9.09%)	
EP. SEIZURE	'N'	6,440 (94.96%)	2,760 (94.96%)	FEATURE SUBSET PERFORMANCE
	'Y'	350 (5.04%)	150 (5.04%)	
F-MNIST	T-SHIRTS	5,250 (95.90%)	1,750 (95.90%)	FEATURE SUBSET PERFORMANCE
	PULLOVERS	225 (4.10%)	75 (4.10%)	BENCHMARK
	T-SHIRTS	6,000 (95.3%)	1,000 (92.2%)	INTERPRETABILITY
	COATS	300 (4.7%)	50 (7.8%)	
MNIST	'1'	6,742 (93.1%)	1,000 (92.2%)	INTERPRETABILITY
	'7'	500 (6.9%)	50 (7.8%)	

TABLE 4 Feature Selection Dataset (FSDS) and Classification Dataset (CDS) composition for the datasets adopted in the experiments.

3. **Epileptic Seizure** [3] ($N = 11,500; 7,300$; $J = 178$). In this functional dataset, each data point represents 178 seconds of EEG recording for one of the 500 patients in the study. Each of the 178 features is the value of the EEG at that time-stamp. The label indicates whether the EEG is recording seizure activity ('Y') or not ('N'). This dataset was originally imbalanced, but we decided to increase the complexity by subsampling minority class further (cfr. Table 4).

4. **Fashion MNIST** [47] ($N = [7,350; 7,300]$; $J = 784$). This dataset is composed by 28x28 grayscale images of clothing. To test our model we built two datasets with different imbalance rates. *T-shirts* were selected as the majority class, and *coats* as the minority one for the first dataset ($\sim 5\%$ of the whole dataset, with 7,350 total observations), while *pullovers* for the second ($\sim 4\%$ with $N = 7,300$).

5. **MNIST** [29] ($N = 8,292$; $J = 784$). This dataset is composed by 28x28 grayscale images of hand-written digits. We selected two quite overlapping classes to test our model: the '7' digit class as the minority class and the '1' digit class as the majority one. This dataset, together with the two extracted from Fashion MNIST, simulate a setting of extreme imbalance (below $95:5$ ratio) and moderately high dimensionality, but with a large sample size.

It should be noted that the proposed algorithm is meant to be applied to features that do not present any dependence (i.e. the order of the features is irrelevant). Its applicability to image datasets is guaranteed by the fact that all images are centered, allowing us to meaningfully treat each pixel as an independent feature. The choice to add image datasets to these experiments derives from both their dimensionality and the clear readability of their results, that allow for visually investigating the selected features by representing them as pixels.

To evaluate the classification performance in an imbalanced setting, we decided not to adopt the classical accuracy on both classes. Instead, we chose the Sensitivity metric (i.e. the ratio of true positives and the sum of true positives and false negatives for observations belonging to the minority class) and the Area Under the Receiver Operating Characteristic (AUROC), that estimates the performance of a binary classifier comparing false positive rates with true positive rates and is a widely used metric to evaluate model's capability to correctly classify both classes, especially in imbalanced settings.

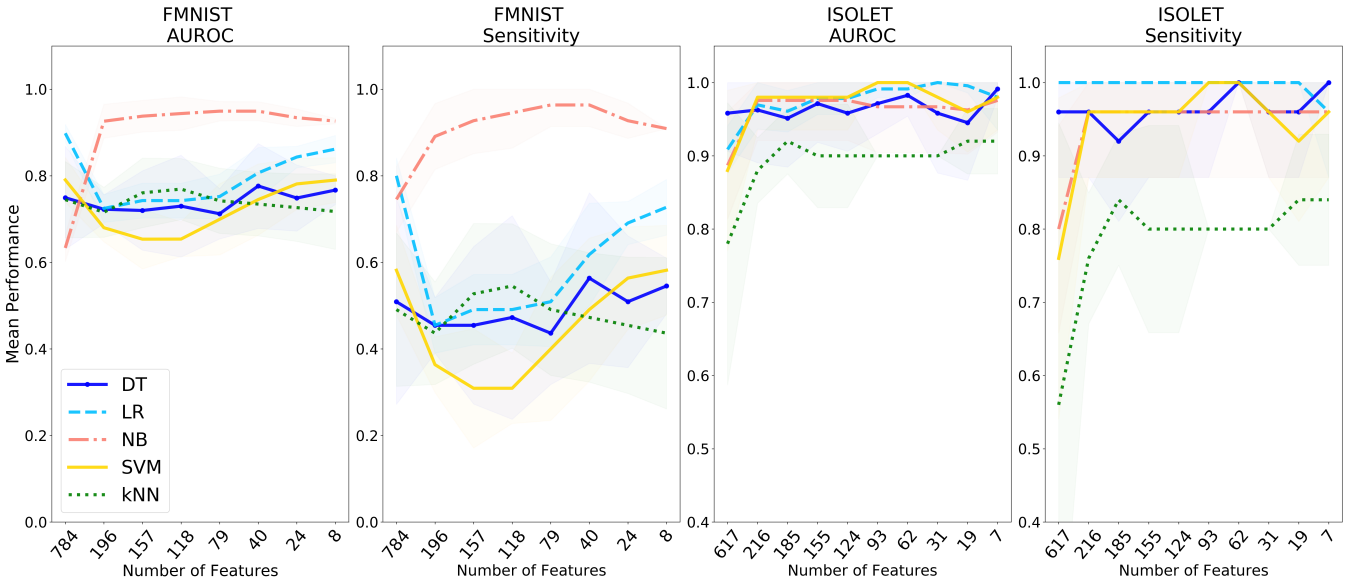


FIGURE 3 AUROC and Sensitivity improvement in the classification of FMNIST dataset (first two plots) and ISOLET dataset (second couple of plots) with different variables' subsets. The lines represent the average performance on 5 trials. Standard errors define the lighter areas around the line. The first number of features in each plot is the performance with the whole original features set.

5.2 | Classification Performance of Selected Feature Subsets

Dimensionality reduction has impacts on computational time and complexity, noise reduction, model significance and results interpretability, but all these improvements should not come at the cost of a good classification performance on the classes of interest. In particular, in research scenarios as those presented in Section 1, a minimum level of precision on Minority Class observations is desirable.

To test our algorithm, we applied it to the FSDS for various δ values ($\delta \in \{0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 0.97, 0.99\}$), selecting different subsets of variables. For each δ we created from the CDS a dataset containing the selected features only. The CDS was subsequently subdivided in training set and test set according to a 70-30 split that was held constant across all experiments.

On the obtained datasets we trained and tested five classifiers: Logistic Regression (LR), Decision Tree (DT), Support Vector Machines (SVM), Naive Bayes (NB) and Nearest Neighbor (NN) classifier. We chose to test different classifiers to verify whether our model-agnostic feature selection approach provided good results indepently of the subsequent classifier adopted. All algorithms were drawn from scikit-learn library for Python [39] and their hyperparameters were kept in default mode, unless differently stated. Note that we applied the same classifiers to all experiments without tailoring their parameters to the data at hand. This choice does not resemble a traditional classification process in a real-life scenario, where classifiers are optimized to improve the performance on the data at hand, but aimed at showcasing the impact of the feature subset selection alone.

Details on the code, the implementation and the specific architectural choices for the DSAEE are described and discussed in Appendix.

We tested the DSAEE feature selector on Isolet, Fashion-MNIST, Gisette and Epileptic Seizure datasets. Results for FMNIST and Isolet datasets are reported in Figure 3 , where performance metrics are averaged over 5 trials and the x-axes display the size of the selected feature sets.

On FMNIST Dataset (Figure3 , first two panels) most classifiers suffered the dimensionality reduction up until smaller subsets, when their performance started growing again. On the contrary, NB classifier had a steep improvement on both AUROC and Sensitivity, reaching almost perfect scores for subsets of extremely small dimensionality (8 features, $\sim 1\%$ of the original 28x28 image).

On Isolet Data, where the sample size is extremely small compared to the number of features and the minority class in the training set contains 52 observations only, the five classifiers performed most of the times as good as the baseline performance

	Naïve Bayes				SVM			
	AUROC		Sensitivity		AUROC		Sensitivity	
|F|	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**
178	0.932	0.029	0.89	0.057	0.907	0.027	0.827	0.055
54	0.924	0.031	0.876	0.064	0.894	0.018	0.800	0.035
45	0.926	0.017	0.880	0.034	0.890	0.019	0.791	0.037
36	0.924	0.018	0.876	0.037	0.890	0.021	0.791	0.043
27	0.922	0.014	0.871	0.029	0.881	0.018	0.773	0.037
18	0.912	0.022	0.853	0.046	0.867	0.035	0.742	0.071
13	0.906	0.022	0.840	0.046	0.852	0.038	0.711	0.079
9	0.898	0.027	0.822	0.052	0.835	0.035	0.680	0.071

TABLE 5 Classification results for Epileptic Seizure Dataset with NB and SVM classifiers. Mean and Standard deviations are averaged over 5 trials.

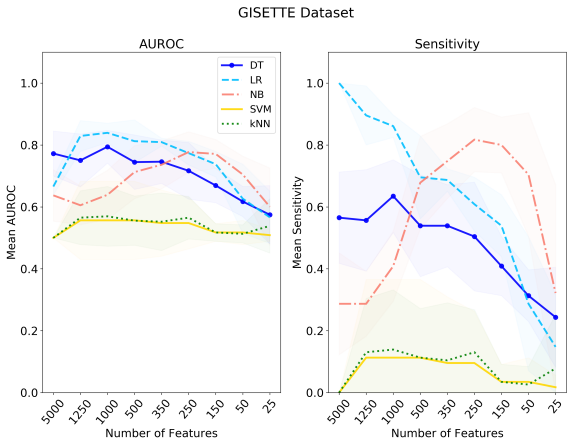


FIGURE 4 Classification results for GISETTE Dataset with all 5 classifiers. Mean and Standard deviations are averaged over 5 trials.

with all variables, despite the reducing size of the features subset. Sensitivity (Figure 3 fourth panel) increased substantially for KNN and SVM, while the LR classifier kept attaining an almost perfect score even as the cardinality ($|F|$) of the selected features set decreased substantially (while improving on AUROC score, as shown in Figure 3 third panel). In many cases, the classifiers obtained their best results as $|F|$ decreased.

In Table 5 we report the classification performance on the Epileptic Seizure Dataset. The first line summarizes baseline results. Note that we chose to include only NB and SVM classifiers, as LR, KNN and DT demonstrated a baseline performance that was too poor to meaningfully consider them for classification on this data. On the contrary, NB and SVM showed a high baseline performance despite the strong imbalance. Decreasing the amount of features used in classification did not hinder the performance, while reducing the dimensionality of the problem. For example, by reducing it to a third ($|F| = 54$), NB did not significantly reduce AUROC or Sensitivity metrics, while the performance for much smaller subsets ($|F| = \{18, 13, 9\}$) remains comparable with the baseline. In Figure 4 we report the results of the experiment on Gisette data. Note that this dataset was designed for feature selection benchmarking experiments, by including 2,500 predictive features and 2,500 probes. By looking at $|F|$ values on the x-axis, one can note that the feature subsets selected by the DSAEE are way smaller than the original number of noisy features. However, irrespective of the baseline performance with $|F| = 5,000$, all classifiers showed an increase in performance for some $|F|$ values. This could mean that the algorithm is first correctly excluding noisy features; then, among the informative predictors, it is progressively excluding correlated or reduntant features, identifying the most useful for the classification task at hand. This hypothesis is well supported by the behavior of NB classifier, that by design requires conditional independence to reach optimal classification [52]. In this experiment, NB yields a steep increase on both metrics for smaller $|F|$ values. Only LR suffered a steep decrease in Specificity, that was however balanced by the significant improvement in AUROC (meaning that the performance is better balanced between the two classes) for subsets between 1,000 and 250 features.

5.3 | Feature Selection Benchmarking Experiment

We selected the benchmark feature selection methods for the performance comparison with our DSAEE approach s.t. they would be representative of different types of algorithms. In particular, we included (i) Chi-squared, a supervised filtering feature selection method based on univariate χ^2 statistical tests, and (ii) Recursive Feature Elimination (RFE) [20] - supervised wrapper method, that when combined with SVM classifier (RFE-SVM) was proven one of the best performing methods in [35] for feature selection in imbalanced settings. Finally, we also included (iii) Concrete AutoEncoder Feature Selector (CAEFS), an unsupervised feature selection method based on AEs[2], that in [8] was proven superior to most related algorithms mentioned in

[2]The number of features selected is defined as the K nodes of the concrete layer. To compare the two models we trained the each DSAE learner and the CAEFS for the same number of epochs using the same batch size, and the architecture of the decoder was built equal to that of the DSAEs.

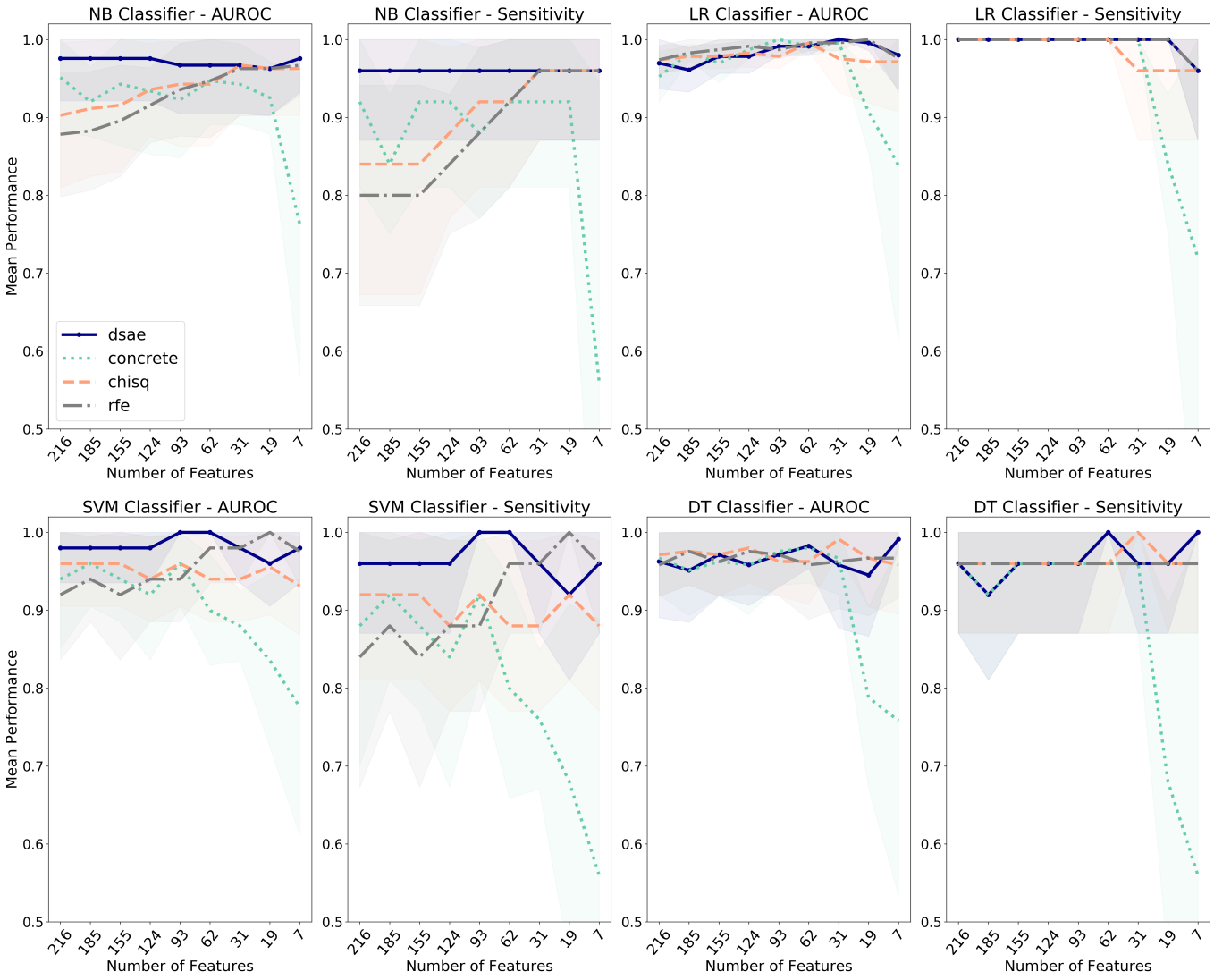


FIGURE 5 Classification benchmarking against other FS methods for ISOLET Datasets, for NB, LR, SVM and DT classifiers. Each classifier has one plot per metric (AUROC on the left, Sensitivity on the right)

Section 2.2.

All benchmark methods were applied to the FSDS imposing a number of selected features equal to the features selected by DSAEE for the different δ levels, then the subsets of selected features were extracted from the CDS to test classification accuracy. We compared the performance on Isolet dataset and Fashion MNIST dataset averaging on 5 trials for each experiment. In both cases we trained an ensemble of $B = 25$ DSAEs.

In Figure 5 we report the results on Isolet using four different classifiers, on Sensitivity and AUROC. Varying the threshold δ we selected a different subset of variables: the cardinality ($|F|$) of such subsets is reported on the x axes. For what concerns NB and SVM classifiers, the DSAEE performed better than the competitors for almost all variables subsets on both indicators. In particular, it significantly outperformed the unsupervised CAEFS for smaller subsets, while the major competition on the smallest dimensionalities was represented by the supervised RFE. Note that RFE-SVM is a feature selection method proved among the best performers for imbalanced settings [35], and the DSAEE either surpasses or reaches comparable performance levels in most cases (see the two plots in the left bottom part of Figure 5). Similar results were obtained with the NB classifier. Regarding LR classifier, all methods seemed to perform well on this dataset, but our methodology reaches an almost perfect score on Sensitivity irrespective of the threshold level, up until to only 7 variables, where the other AE-based FS method (CAEFS) lowered its average performance. These levels of Sensitivity and AUROC - irrespective of the adopted classifier - on a dataset with significantly small sample size and extremely high dimensionality testify in favour of the applicability of our methodology

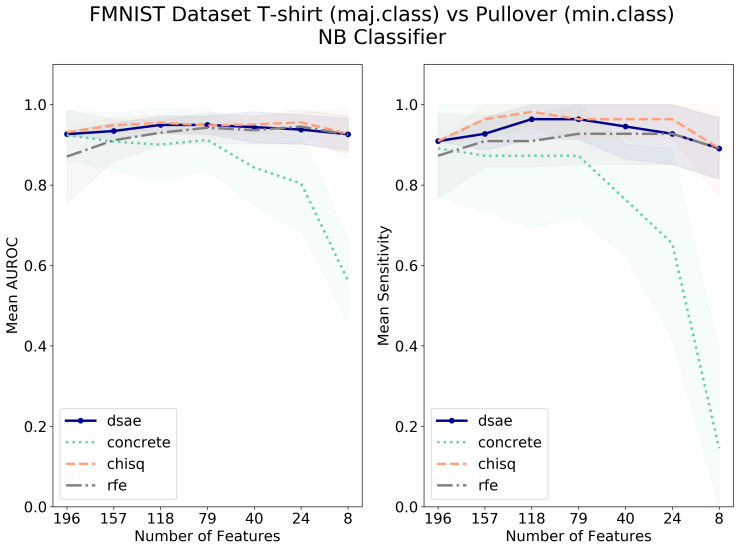


FIGURE 6 Classification benchmarking against other FS methods for Fashion MNIST dataset using NB classifier.

Dataset		DSAE	CHISQ	RFE	CONCRETE	DSAE PARALL.
ISOLET	Average Time [min]	14.010	0.046	1.246	117.268	0.560
	Std [min]	5.927	0.005	0.048	78.034	0.237
FMNIST	Average Time [min]	7.687	0.087	22.852	0.618	0.256
	Std [min]	0.201	0.001	1.424	0.017	0.007

TABLE 6 Comparison of average runtime performance of all benchmark methods on the Isolet Dataset. The average time is computed considering total process time to select feature subsets for all δ thresholds, and averaged over 5 trials.

in many real-life scenarios where the collection of observations might be costly or difficult.

In Figure 6 we compare the performance of the DSAEE on FMNIST dataset using the best performing classifier in terms of performance improvement (Figure 3). Our algorithm confirmed its superiority w.r.t. the competing AE-based FS method, while keeping a comparable performance to the other benchmark algorithms, all set to a very high performance up until an extremely small feature subset (8 pixels from the original 784).

Note that, as can be noticed from Figure 3 , both datasets allowed for high prediction accuracy on both classes even before feature selection. This indicates that probably, despite the imbalanced setting, the two classes are sufficiently separated and consistently characterized to allow classifier to correctly separate them and generalize well just by seeing few examples of the underrepresented class. For this reason, it is not surprising to see all algorithms (especially the supervised ones) perform quite well on this feature selection and classification task. Nonetheless, although our ensemble algorithm is based on unsupervised learners, it consistently reached or surpassed the supervised approaches, and performed significantly better than the unsupervised one.

In Table 6 we report the total process runtime to complete all feature subsets selections (for all δ values) for the different algorithms, averaged over all trials. In the first column each of the trained DSAEs are processed in sequence, while in the last one we report the estimated average time to perform the algorithm's training in parallel. Even though the sequential training time is not prohibitive per se, its parallelized version outperforms the wrapper RFE and the other AE-based algorithm (CAEFS) by far, while enjoying the beneficial robustness of an ensemble framework.

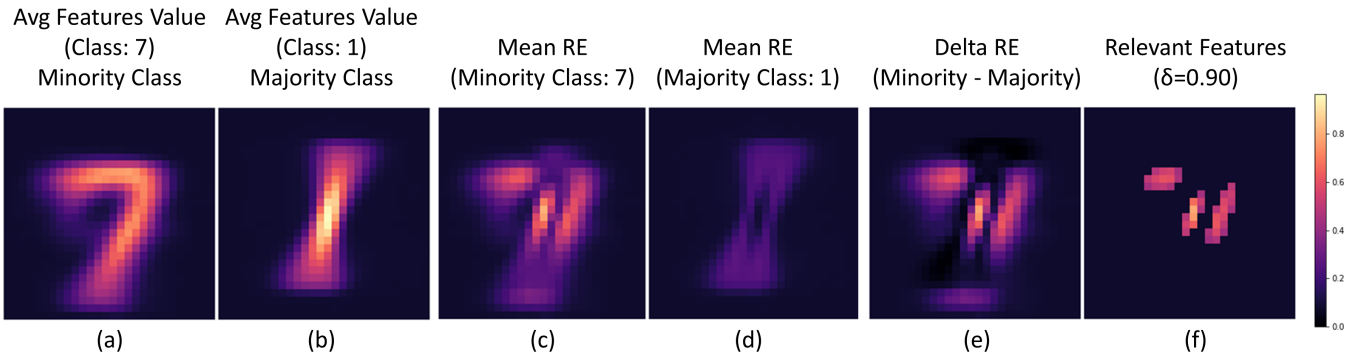


FIGURE 7 Results of the experiment on the 7 (minority) and 1 (majority) classes. In these 28x28 pixels images each pixel represents a feature. Subfigures (a) and (b) represent the mean of all values the two classes take in the FS dataset. The color scale is shared across all six subfigures. Subfigure (c) reports the average RE for the minority class, while (d) is the representation of the majority class average RE - Note that being the '1's class the majority one, the model learns to reconstruct precisely the center of the vertical line that draws the digit. The vector $\mathbf{\Delta}$ is reported in (e), while (f) depicts the selected variables with a threshold $\delta = 0.9$.

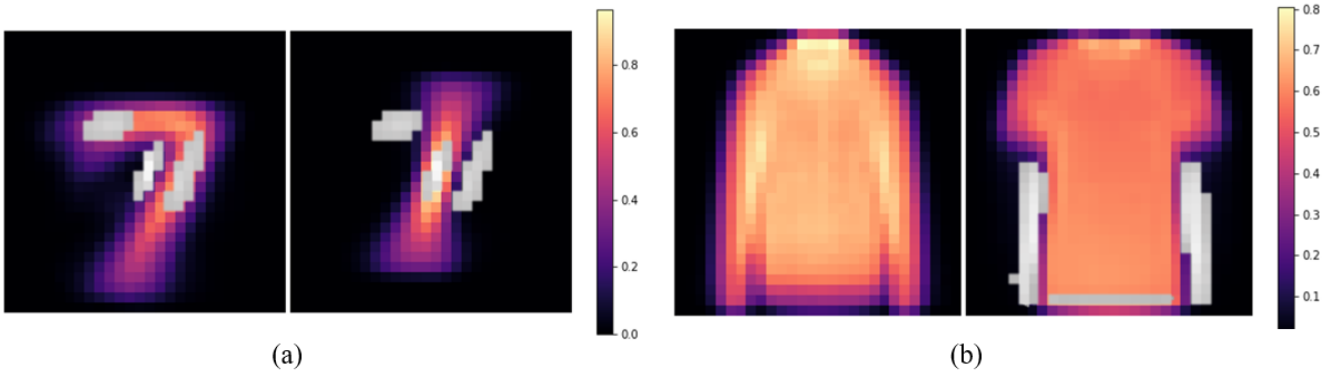


FIGURE 8 (a) MNIST Dataset. The most relevant identified features are represented in gray scale over the average minority (right) and majority class (left) representations. (b) Fashion MNIST Dataset. Here, it is clear how the most relevant features to distinguish *coats* from *t-shirts* are the pixels that compose the sleeves of the coat.

5.4 | Interpretability

One advantage of FS for classification lies in the increased interpretability of the subsequent algorithms and results. Indeed, identifying features that are the most informative, w.r.t. a target class within a dataset is an insightful information by itself in many application contexts. In the era of black-box classifiers, a reduction in the amount of information fed to these algorithms is per se a way of improving the interpretability of (and the control over) the obtained classifications. In the case of our proposed algorithm, the selected features are the subset of variables where the minority class distances the majority one the most.

In Figure 7 we report some visualizations from the MNIST Dataset that help in understanding the feature selection process performed by our algorithm. The small set of selected features for $\delta = 0.90$ (Figure 7 .f) is then overlapped (in gray scale) to the average representation of the two classes (Figure 8 .a). This visualization allows us to recognize how the selected features include all pixels where the minority class ('7' digits) have different characteristics w.r.t. the '1' digits class.

In Figure 8 .b we propose the same visualization for the Fashion MNIST dataset. Note that these features subsets were obtained in an highly imbalanced setting, as reported in Table 4 , but the selected features are extremely meaningful nonetheless.

5.5 | Case Study application in Radiogenomics

Class imbalance is a daunting issue in many real life applications, especially when dealing with medical and biological data [43] (cfr. Section 1). So far, we presented simulation studies and proofs of concept to demonstrate the generalizable potential of the proposed algorithm. However, the value of the presented approach lies in its demonstrated applicability to complex scenarios arising from real life research settings. Indeed, in this section we present a real data application of the DSAEE FS algorithm in the field of radiogenomics. A detailed report on the study can be found in [36]. However, because of the aforementioned reasons, we were interested in providing here a brief description nonetheless. Specifically, we focused on the long term outcomes of radiotherapy on patients suffering from prostate cancer. The final aim was to validate genetic locations (in the form of Single Nucleotide Polymorphisms, or SNPs) that can be associated with Late Toxicity (LT) outcomes. Experts were indeed interested in finding whether among the features (i.e. the SNPs) with high association to the 5 considered LT endpoints in previous studies on different cohorts, some could be validated as relevant for the cohort at hand ($\sim$ 1,700 patients with an incidence of the positive class always below 10% for each endpoint and a total number of 43 SNPs to evaluate). We applied our DSAEE on each of the 5 endpoints separately, and we selected SNPs with different δ thresholds ($\delta = \{0.7, 0.8, 0.9, 0.95\}$). This being an unsupervised setting it is hard to comment on precision of the results without the required clinical expertise. However, notably, for one of the endpoints (i.e. Late Urinary Frequency) 3 SNPs identified as relevant by our method for all δ values were previously mentioned in literature [26] as the most strongly associated to this endpoint.
This is an interesting application case in which FS methods are useful to profile minority class, and provide useful insights to researchers. As introduced in Section 1, our FS algorithm is indeed tailored to respond to similar needs and to deal with complex scenarios where the class of interest is extremely rare.

6 | DISCUSSION AND CONCLUSIONS

In this paper we presented a Deep Learning-based ensemble approach to select features for highly imbalanced classification tasks. The proposed approach exploits Deep Sparse AutoEncoders as *weak learners*, each trained to learn the *normal* patterns in majority class observations, and tested on both majority and minority class data. Diversity among components of the ensemble is fostered by a tailored sampling procedure and the sparsity constraint on the training loss function. Features are ranked averaging on the RE of the ensemble of learners to identify the most informative ones, where minority class distribution differs from majority class the most.
We performed a series of analyses, on simulated and real data, to validate our claims and test the potential of our DSAEE. First, in a simulation setting, we evaluated the capability of DSAEE algorithm to induce well separated average RE distributions of the two classes, and the value of their differential distribution (i.e. $\mathbf{\Delta}$) to identify the most informative features. This synthetic and controlled setting allowed us to support at least empirically all the hypotheses underlying our design choices, despite the lack of a complete theoretical framework regarding complex DL-based methods. Then, exploiting a wide range of benchmark data to mimick real research settings, we verified DSAEE's ability to avoid the degradation of classification performance induced by selecting feature subsets in the presence of strong imbalance [48]. We compared baseline performances including the entire original feature set, with that obtained with subsets of increasingly small dimensionality. We also benchmarked our method against other feature selection methods, demonstrating the superior or comparable performance of the DSAEE feature selector. Note that most of the algorithms we compared the DSAEE with had the advantage of being supervised, or even tailored to maximize prediction accuracy on minority class (RFE-SVM).
Our FS algorithm is tailored to manage extremely imbalanced settings with the aim of attaining all the advantages of FS methods without sacrificing too much on the classification performance by reducing the amount of information supplied to classifiers. First we validated DSAEEs capability to discard useless information in the first simulation experiment, then we evaluated its robustness in selecting informative features up to extreme imbalance levels. Additionally, when measuring classification performance on real data, in some cases we observed the algorithm identify subsets of the original features yielding an improved AUROC and/or Specificity (cfr. Figure 3 and Figure 4). In particular, the improved Specificity might be induced by the training procedure of each ensemble DSAE component: indeed, AEs by nature represent an approximation of the identity function and the applied model is compelled to learn the common characteristics of the data [42]. By training on majority class only, the learnt data distribution does not include the characterization aspects of minority class instances, thus generating higher reconstruction errors on those features. Moreover, the initial data sampling, once included in an ensemble framework, allows to extract reliable information even when the observations belonging to the minority class are limited. While creating

the different sampled training and test set for each ensemble component, the minority class is indeed studied against various subsets of the majority one, thus enhancing the informative power of the small underrepresented sample. Indeed, when tested against very small minority class samples in our simulation (i.e. 15 or 45 observations), the algorithm consistently selected relevant features carrying class-separating information.
On top of the sampling procedure we included to the training loss function of our components a sparsity penalty term, that besides fostering components' diversity reduces the need for lengthy optimization of the DSAEs' architecture. Indeed, the penalty term forces the number of active nodes in the hidden layer to adapt to the sample of training data, reducing autonomously the risk of learning trivial representations. We validated the value of sparsity in our simulation study, where the right regularization balance obtained the best separation of the two class-specific average RE distributions. Moreover, the same unoptimized *toy architecture* was capable of excluding all uninformative features from selection up until a quite low δ threshold value, irrespective of inter-class separability.
In addition to all the above, the DSAEE Feature Selection algorithm is a filtering method, meaning that it is agnostic to the classifier exploited to discriminate between classes. This may slightly hinder classification accuracy compared for instance to wrapper methods, but gains generalizability of the identified features. Moreover, when compared to wrapper methods, our approach does not incur in the risk of sub-optimal solutions in high-dimensional settings, where evaluating all possible combinations of features would be computationally intractable. When compared to embedded methods, our AE-based approach is capable of capturing nonlinear relationships among features. Kernel-based embedded feature selection methods were proposed to learn nonlinear representations [30], but they are limited by the fixed kernel, and the choice of the optimal kernel or combination of kernels is not straightforward.

In conclusion, with this work we are taking inspiration from different methodological domains to develop a novel filtering feature selection algorithm that is (i) robust thanks to its ensemble nature, (ii) capable to learn complex patterns in data because of its AE components, (iii) provides interpretable insights and (iv) is specifically tailored to tackle class imbalance. All these considerations promote the usefulness of our DSAEE feature selector in real-life contexts where data are imbalanced, minority class observations have great relevance, sample size is small, and interpretability of results is crucial. We provided a direct example in Section 5.5, where a real data application is briefly described.
Future works might be devoted to studying the applicability of the DSAE feature selector to imbalanced multi-class classification problems or to further develop the analysis of the RE distributions to select features.

ACKNOWLEDGMENTS

The Authors thank the ERA PerMed Cofund program, grant agreement No ERAPERMED2018-44, RADprecise - Personalized radiotherapy: incorporating cellular response to irradiation in personalized treatment planning to minimize radiation toxicity.

Financial disclosure

FG was funded by the UK Medical Research Council programme
MRC_MC_UU_00002/5.

Conflict of interest

The authors declare no potential conflict of interests.

APPENDIX

A SIMULATION STUDY SUPPLEMENTARY MATERIAL

In the following section we provide all the supplementary information regarding the simulation study experiments.

A.1 Simulated Data

The simulated data was constructed exploiting Python Scikit-learn implementation of the algorithm to generate MADELON dataset for NIPS2003 Variable Selection Challenge. Specific details on the pre-implemented code can be found at https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html.However, for the purpose of our analyses the algorithm was adapted from the original and specific choices were made, therefore in this section we will provide details about the generative mechanism of our simulated data.

In particular, the algorithm generating our simulated data takes as an input the number of features that will be informative to characterize the generated data-clouds ($J_I = |F_{inf}|$) representing the two classes of interest. As a first step, it creates J_I-dimensional gaussian clusters by sampling J_I independent features from a Normal Distribution $N(0, 1)$. In our implementation, each of the two classes are composed of one of those clusters, and class sample sizes are defined by assigning *class weights* as the proportion of N generated data assigned to each class. Then, some covariance is added within each cluster by multiplying by a random matrix A, with uniformly distributed random numbers between -1 and 1.
At this point, each class is placed at random on the vertices of a hypercube in a J_I-dimensional space. The length of the hypercube's sides is proportional to the *class separation* hyperparameter s. In particular, each side is $2s$ long, and larger s values place the clusters (i.e. the classes) further apart, making the classification task easier.
At this point, J_R features carrying redundant information are added, obtained by random linear combination of the informative features. Last, a set of J_U uninformative features is added, filled by random Gaussian noise from $N(0, 1)$.
In conclusion, the total number J_{SIM} of features comprise

- J_I informative features,
- J_R redundant features, and
- $J_{SIM} - J_I - J_R$ useless features drawn at random.

We set the order of the features in the output dataset so that the informative features are assigned to the first J_I columns, followed by the J_R reduntand features. The remainder of the dataset is filled with uninformative features.
As mentioned in the paper, for all our experiments we set $N = 1500$, $J_{SIM} = 150$, $J_I = J_R = 25$, s.t. each dataset contained 100 useless random probes. Class Weights and s were defined according to the specific experiment, as reported in the following sections and in the main corpus of the paper.

A.2 Implementation and Architectural Details

For the whole Simulation Study we exploited the same *toy architecture* for each AE component in the ensemble, that is reported in Figure A1 . In particular, each AE component was designed as a fully connected AE with an input layer, a 100-nodes hidden layer and a 50-nodes bottleneck layer. The decoder was composed of only one 100-nodes hidden layer followed by the output layer (150 nodes, as the input dimension). The specific activation function applied to the nodes of each layer is reported within parenthesis in Figure A1 .
Each component was trained for 500 epochs with a batch size of 300 observations. Further specific details for each experiment can be summarized as in Table A1 . The first two experiments presented in the paper in Section 4.2 are referred to as Baseline 1 and Baseline 2 in the table.

Experiment	s	B	λ	Class Weights
Baseline 1	2	20	0.05	[0.95-0.05]
Baseline 2	0.1	20	0.05	[0.95-0.05]
Varying L1		10	[1.0, 0.5, 0.25, 0.1, 0.05, 0.001, 0.0001, 0.00001, 0.000001, 0.0]	[0.95-0.05]
Robustness to s	[0.01, 0.1, 0.3, 0.5, 0.7, 1, 1.5, 2, 3, 5, 10]	10	0.05	[0.95-0.05]
Robustness to Imbalance	1	15	0.05	[0.9-0.1], [0.95- 0.05], [0.97-0.03], [0.99-0.01]

TABLE A1 Simulation Experiments details

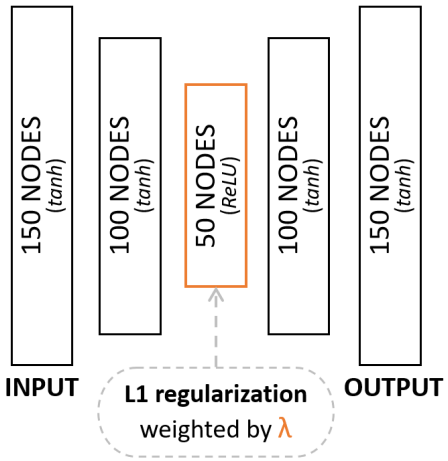


FIGURE A1 Toy Architecture exploited across the whole simulation study for each AE ccomponent in the ensemble.In parenthesis we report the activation function exploited in the specific layer.

A.3 Further Results

In Table A2 are reported the complete results for the experiment described in Section 4.2, here referred to as Baseline 1, with $s = 2$.
In Table A3 are reported the complete results for the experiment described in Section 4.2, here referred to as Baseline 2, with $s = 0.1$.
In Table A4 are reported the complete results for the experiment described in Section 4.3, testing the effect of L1 penalization.
In Table A5 are reported the complete results for the experiment described in Section 4.4, testing the robustness to class separability (s).
In Table A6 are reported the complete results for the experiment described in Section 4.4, testing the robustness to class imbalance, both in terms of class-specific RE distribution separation and informative feature selection (FS performance).

B DSAEE ARCHITECTURAL AND IMPLEMENTATION DETAILS FOR REAL DATA EXPERIMENTS

In the following section we provide the details of the architectural and implementation choices made on the DSAEs for the different experiments on real datasets, as described in Section 5. Note that these choices are provided for the sake of results' reproducibility, but they should not be considered a strict guideline about how the components in the ensemble should be built. Indeed, the DSAE is a fundamental building block of our methodology, but just as in any application of deep learning models, it should be customized to the problem at hand. For that reason, we did not focus all our effort in seeking for the lightest and fastest possible architecture. Our focus and concern was on the demonstration of the potentials of the methodology as a whole,

dataset id	Wasserstein	Wilcoxon p-value	Wilcoxon Stat	Wilc. two-sided p-value	Wilc. two-sided Stat
1	0.407	1,16E-09	81.0	5,79E-11	11244.0
2	0.583	2,35E-10	1.0	1,17E-10	11324.0
3	0.459	3,17E-11	16.0	1,59E-11	11309.0
4	0.313	6,29E-10	167.0	3,15E-09	11158.0
5	0.650	2,30E-10	0.0	1,15E-10	11325.0
6	0.440	2,44E-10	3.0	1,22E-10	11322.0
7	0.544	2,39E-10	2.0	1,20E-10	11323.0
8	0.402	4,28E-10	31.0	2,14E-10	11294.0
9	0.574	2,30E-10	0.0	1,15E-10	11325.0
10	0.506	2,30E-10	0.0	1,15E-10	11325.0

TABLE A2 Results for each of the 10 datasets generated for the first experiment described in Section 4.2, with $s = 2$

dataset id	Wasserstein	Wilcoxon p-value	Wilcoxon Stat	Wilc. two-sided p-value	Wilc. two-sided Stat
1	0,395	2,10E-10	111.0	1,05E-10	11214.0
2	0.576	2,30E-10	0.0	1,15E-10	11325.0
3	0.470	4,11E-11	29.0	2,06E-10	11296.0
4	0.317	9,11E-10	186.0	4,56E-10	11139.0
5	0.653	2,30E-10	0.0	1,15E-10	11325.0
6	0.478	3,17E-11	16.0	1,59E-11	11309.0
7	0.528	2,30E-10	0.0	1,15E-10	11325.0
8	0.405	3,11E-10	15.0	1,55E-10	11310.0
9	0.576	2,30E-10	0.0	1,15E-10	11325.0
10	0.506	2,30E-10	0.0	1,15E-10	11325.0

TABLE A3 Results for each of the 10 datasets generated for the first experiment described in Section 4.2, with $s = 0.1$

that exploits this well known building block within a novel algorithm to robustly identify features to separate the two classes.

The algorithm was developed in Python 3.6, using Keras and Tensowrflow as back-end. The code was run on Jupyter notebooks hosted on Google Colab Virtual Machines[3], with access to GPUs and Fast VMs thanks to the Pro subscription. The types of GPUs that are available in Colab vary over time. The GPUs available in Colab often include Nvidia K80s, T4s, P4s and P100s. There is no way to choose what type of GPU you can connect to in Colab at any given time.

An overview of the architectural and implementation choice of the proposed method for the four different datasets evaluated in this work is reported in Tab.B7 . In this table we give details about the Encoder (number of nodes for input and hidden layers, activation function per layer) and the Decoder (number of nodes in the hidden and output layers, activation function per layer). If a single type of function is reported (like tanh as activation function in the Decoder for ISOLET data), this means that we chose the same activation function across all layers. In the bottom part of Tab. B7 , we describe the number of epochs, the batch size and the parameter B of the algorithm.

The following parameters have been selected consistently across the four different datasets:

- the last hidden layer in the Encoder had an L_1 penalization on the activation of the 200 nodes, with $\lambda = 10e^{-5}$. The value of this hyperparameter was chosen between $\lambda = (10e^{-5}, 10e^{-10}, 10e^{-20})$ as the one that guaranteed a low reconstruction error, while favouring a sufficient penalization on the activation of the hidden nodes.

[3] https://colab.research.google.com/notebooks/intro.ipynb

	Wasserstein		Wilc. log(p-value)		Wilc. Stat		Wilc. two-sided log(p-value)		Wilc. two-sided Stat	
λ	mean	std	mean	std	mean	std	mean	std	mean	std
0	0.182	0.031	-33.798	6.143	1450	404.865	-34.491	6.143	9875	404.865
1.00E-06	0.182	0.032	-33.351	5.985	1480.3	410.934	-34.044	5.985	9844.7	410.934
1.00E-05	0.179	0.035	-33.26	5.915	1485.6	401.693	-33.953	5.915	9839.4	401.693
0.0001	0.185	0.035	-35.396	6.709	1349.2	451.416	-36.089	6.709	9975.8	451.416
0.001	**0.205**	**0.032**	**-42.842**	**6.347**	**884.4**	**380.175**	**-43.535**	**6.347**	**10440.6**	**380.175**
0.05	**0.248**	**0.04**	**-42.994**	**6.794**	**876.5**	**394.759**	**-43.687**	**6.794**	**10448.5**	**394.759**
0.1	**0.255**	**0.055**	**-36.532**	**8.953**	**1287.6**	**564.676**	**-37.225**	**8.953**	**10037.4**	**564.676**
0.25	0.165	0.016	-2.937	2.126	4694.6	437.508	-3.63	2.126	6630.4	437.508
0.5	0.166	0.017	-2.548	2.089	4808.2	493.277	-3.241	2.089	6516.8	493.277
1	0.164	0.018	-2.908	2.446	4723.1	506.049	-3.602	2.446	6601.9	506.049

TABLE A4 Complete results for Experiment with varying L1 penalization, as described in Section 4.3.

	Wasserstein		Wilc. log(p-value)		Wilc. Stat		Wilc. two-sided log(p-value)		Wilc. two-sided Stat	
s	mean	std	mean	std	mean	std	mean	std	mean	std
0.01	0.158	0.024	-29.89	4.412	1710.4	312.218	-30.583	4.412	9614.6	312.218
0.1	0.154	0.023	-28.747	4.563	1794.7	349.391	-29.44	4.563	9530.3	349.391
0.3	0.162	0.021	-31.222	3.182	1612.2	222.317	-31.916	3.182	9712.8	222.317
0.5	0.178	0.025	-34.233	4.595	1414.5	313.897	-34.926	4.595	9910.5	313.897
0.7	0.202	0.034	-38.67	6.634	1138.4	412.989	-39.363	6.634	10186.6	412.989
1	0.247	0.047	-46.905	7.132	653.1	405.855	-47.598	7.132	10671.9	405.855
1.5	0.352	0.075	-54.656	3.738	225.1	194.991	-55.35	3.738	11099.9	194.991
2	0.486	0.108	-58.02	1.855	51.4	94.669	-58.713	1.855	11273.6	94.669
3	0.783	0.165	-59.004	0.076	1.5	3.808	-59.697	0.076	11323.5	3.808
5	1.271	0.168	-59.026	0.019	0.4	0.966	-59.72	0.019	11324.6	0.966
10	2.133	0.292	-59.032	0.006	0.1	0.316	-59.726	0.006	11324.9	0.316

TABLE A5 Complete results for Experiment on Robustness to Class Separation s, as described in Section 4.4

	Wasserstein		Wilc. log(p-value)		Wilc. Stat		Wilc. two-sided log(p-value)		Wilc. two-sided Stat		FSP	
Class Weights	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
[0.9, 0.1]	0.261	0.05	-55.394	3.929	187.5	208.349	-56.087	3.929	11137.5	208.349	1	0
[0.95, 0.05]	0.244	0.042	-47.188	6.177	633.3	346.259	-47.881	6.177	10691.7	346.259	1	0
[0.97, 0.03]	0.246	0.04	-39.776	5.598	1065.9	351.498	-40.469	5.598	10259.1	351.498	0.987	0.028
[0.99, 0.01]	0.265	0.062	-27.001	5.822	1930.1	431.071	-27.694	5.822	9394.9	431.071	0.893	0.095

TABLE A6 Complete results for Robustness to class imbalance experiment, as described in Section 4.4. FS Performance (*FSP*) is defined as the proportion of useful features selected among all selected features with $\delta = 0.9$ (i.e. 15 features).

- the DSAE was trained with the Adam optimization algorithm ($learning\_rate = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). We decided not to optimize these hyperparameters because of computational time of the experiments. Therefore we kept them as the default standard suggested by literature when analyzing all the four different datasets.

TABLE B7 Details of the architectural and of the implementation are reported here for the four analysed datasets. The function tanh is the hyperbolic tangent, the ReLu is the Rectified Linear Unit function. *Enc.* section of the table reports the encoder architecture, while *Dec.* details the decoder. The bottom part of the table (*Train.*) reports details on the training procedure (number of epochs, batch size and number of ensemble components B).

		ISOLET	**GISETTE**	**MNIST**	**F-MNIST**	**EP. SEIZURE**
Enc.	Nodes	617-600-500-250-200	5000-1000-500-250-250	784-700-500-250-200	784-700-500-250-200	178-132-64-32
	Act. funct.	tanh-tanh-tanh-ReLu	sigmoid-sigmoid-ReLu-ReLu	sigmoid	tanh-tanh-tanh-ReLu	tanh
Dec.	Nodes	250-500-600-617	250-500-1000-5000	250-500-700-784	250-500-700-784	64-132-178
	Act. funct.	tanh	ReLu-sigmoid-sigmoid-sigmoid	sigmoid	tanh	tanh
Train.	Epochs	100	50	50	100	200
	Batch sz.	10	1000	100	100	1000
	B	25	25	50	50	30

References

[1] Aggarwal, C. C., 2015: Outlier analysis. *Data mining*, Springer, 237–263.

[2] Ali, A., S. M. Shamsuddin, A. L. Ralescu, et al., 2015: Classification with class imbalance problem: a review. *Int. J. Advance Soft Compu. Appl*, **7**, no. 3, 176–204.

[3] Andrzejak, R. G., K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, 2001: Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, **64**, no. 6, 061907.

[4] Annemans, L., K. Redekop, and K. Payne, 2013: Current methodological issues in the economic assessment of personalized medicine. *Value in Health*, **16**, no. 6, S20–S26.

[5] Anwar, N., G. Jones, and S. Ganesh, 2014: Measurement of data complexity for classification problems with unbalanced data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **7**, no. 3, 194–211.

[6] Austin, E., W. Pan, and X. Shen, 2013: Penalized regression and risk prediction in genome-wide association studies. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **6**, no. 4, 315–328.

[7] Baldi, P., 2012: Autoencoders, unsupervised learning, and deep architectures. *Proceedings of ICML workshop on unsupervised and transfer learning*, 37–49.

[8] Balın, M. F., A. Abid, and J. Zou, 2019: Concrete autoencoders: Differentiable feature selection and reconstruction. *Proceedings of the 36th International Conference on Machine Learning*, PMLR, Long Beach, California, USA, volume 97, 444–453.
URL http://proceedings.mlr.press/v97/balin19a.html

[9] Chandra, B. and R. K. Sharma, 2015: Exploring autoencoders for unsupervised feature selection. *2015 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 1–6.

[10] Chen, J., S. Sathe, C. Aggarwal, and D. Turaga, 2017: Outlier detection with autoencoder ensembles. *Proceedings of the 2017 SIAM international conference on data mining*, SIAM, 90–98.

[11] Chen, X.-w. and M. Wasikowski, 2008: Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 124–132.

[12] Chen, Z., C. K. Yeo, B. S. Lee, C. T. Lau, and Y. Jin, 2018: Evolutionary multi-objective optimization based ensemble autoencoders for image outlier detection. *Neurocomputing*, **309**, 192–200.

[13] Cuaya, G., A. Munoz-Meléndez, and E. F. Morales, 2011: A minority class feature selection method. *Iberoamerican Congress on Pattern Recognition*, Springer, 417–424.

[14] Dietterich, T. G., 2000: Ensemble methods in machine learning. *International workshop on multiple classifier systems*, Springer, 1–15.

[15] Fanty, M. and R. Cole, 1991: Spoken letter recognition. *Advances in Neural Information Processing Systems*, 220–226.

[16] Feng, S. and M. F. Duarte, 2018: Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation. *Neurocomputing*, **312**, 310–323.

[17] Guyon, I. and A. Elisseeff, 2003: An introduction to variable and feature selection. *Journal of machine learning research*, **3**, no. Mar, 1157–1182.

[18] Guyon, I., S. Gunn, A. Ben-Hur, and G. Dror, 2005: Result analysis of the nips 2003 feature selection challenge. *Advances in neural information processing systems*, 545–552.

[19] Guyon, I., S. Gunn, A. B. Hur, and G. Dror, 2006: Design and analysis of the nips2003 challenge. *Feature extraction*, Springer, 237–263.

[20] Guyon, I., J. Weston, S. Barnhill, and V. Vapnik, 2002: Gene selection for cancer classification using support vector machines. *Machine learning*, **46**, no. 1-3, 389–422.

[21] Han, K., Y. Wang, C. Zhang, C. Li, and C. Xu, 2018: Autoencoder inspired unsupervised feature selection. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2941–2945.

[22] He, J. and J. Carbonell, 2010: Coselection of features and instances for unsupervised rare category analysis. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **3**, no. 6, 417–430, doi:https://doi.org/10.1002/sam.10091. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.10091

[23] Hinton, G. E. and R. R. Salakhutdinov, 2006: Reducing the dimensionality of data with neural networks. *science*, **313**, no. 5786, 504–507.

[24] Hira, Z. M. and D. F. Gillies, 2015: A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, **2015**.

[25] Jung, L. C., H. Wang, X. Li, and C. Wu, 2020: A machine learning method for selection of genetic variants to increase prediction accuracy of type 2 diabetes mellitus using sequencing data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **13**, no. 3, 261–281.

[26] Kerns, S. L., L. Dorling, L. Fachal, S. Bentzen, P. D. Pharoah, D. R. Barnes, A. Gómez-Caamaño, A. M. Carballo, D. P. Dearnaley, P. Peleteiro, et al., 2016: Meta-analysis of genome wide association studies identifies genetic markers of late toxicity following radiotherapy for prostate cancer. *EBioMedicine*, **10**, 150–163.

[27] Kieu, T., B. Yang, C. Guo, and C. S. Jensen, 2019: Outlier detection for time series with recurrent autoencoder ensembles. *IJCAI*, 2725–2732.

[28] Lal, T. N., O. Chapelle, J. Weston, and A. Elisseeff, 2006: Embedded methods. *Feature extraction*, Springer, 137–165.

[29] LeCun, Y., C. Cortes, and C. J. Burges, 2010: *Mnist handwritten digit database*. Http://yann.lecun.com/exdb/mnist.

[30] Liang, Z. and T. Zhao, 2006: Feature selection for linear support vector machines. *18th International Conference on Pattern Recognition (ICPR'06)*, IEEE, volume 2, 606–609.

[31] Liu, H. and H. Motoda, 2007: *Computational methods of feature selection*. CRC Press.

[32] Liu, M., C. Xu, Y. Luo, C. Xu, Y. Wen, and D. Tao, 2017: Cost-sensitive feature selection by optimizing f-measures. *IEEE Transactions on Image Processing*, **27**, no. 3, 1323–1335.

[33] Liu, Y., Y. Wang, X. Ren, H. Zhou, and X. Diao, 2019: A classification method based on feature selection for imbalanced data. *IEEE Access*, **7**, 81794–81807.

[34] Ma, Y., P. Zhang, Y. Cao, and L. Guo, 2013: Parallel auto-encoder for efficient outlier detection. *2013 IEEE International Conference on Big Data*, IEEE, 15–17.

[35] Maldonado, S., R. Weber, and F. Famili, 2014: Feature selection for high-dimensional class-imbalanced data sets using support vector machines. *Information Sciences*, **286**, 228–246.

[36] Massi, M. C., F. Gasperoni, F. Ieva, A. M. Paganoni, P. Zunino, A. Manzoni, N. R. Franco, L. Veldeman, P. Ost, V. Fonteyne, et al., 2020: A deep learning approach validates genetic risk factors for late toxicity after prostate cancer radiotherapy in a requite multi-national cohort. *Frontiers in oncology*, **10**.

[37] Mazurowski, M. A., P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi, 2008: Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks*, **21**, no. 2-3, 427–436.

[38] Nie, F., H. Huang, X. Cai, and C. H. Ding, 2010: Efficient and robust feature selection via joint l2, 1-norms minimization. *Advances in neural information processing systems*, 1813–1821.

[39] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, 2011: Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, **12**, 2825–2830.

[40] Protopapadakis, E., A. Voulodimos, A. Doulamis, N. Doulamis, D. Dres, and M. Bimpas, 2017: Stacked autoencoders for outlier detection in over-the-horizon radar signals. *Computational intelligence and neuroscience*, **2017**.

[41] Sánchez-Maroño, N., A. Alonso-Betanzos, and M. Tombilla-Sanromán, 2007: Filter methods for feature selection–a comparative study. *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, 178–187.

[42] Sarvari, H., C. Domeniconi, B. Prenkaj, and G. Stilo, 2019: Unsupervised boosting-based autoencoder ensembles for outlier detection. *arXiv preprint arXiv:1910.09754*.

[43] Thabtah, F., S. Hammoud, F. Kamalov, and A. Gonsalves, 2020: Data imbalance in classification: Experimental evaluation. *Information Sciences*, **513**, 429–441.

[44] Vallender, S., 1974: Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, **18**, no. 4, 784–786.

[45] Wasikowski, M. and X.-w. Chen, 2009: Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on knowledge and data engineering*, **22**, no. 10, 1388–1400.

[46] Wei, W., J. Li, L. Cao, Y. Ou, and J. Chen, 2013: Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, **16**, no. 4, 449–475.

[47] Xiao, H., K. Rasul, and R. Vollgraf, 2017: *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms*. Https://research.zalando.com/welcome/mission/research-projects/fashion-mnist/.

[48] Yin, L., Y. Ge, K. Xiao, X. Wang, and X. Quan, 2013: Feature selection for high-dimensional imbalanced data. *Neurocomputing*, **105**, 3–11.

[49] Yousefi-Azar, M., V. Varadharajan, L. Hamey, and U. Tupakula, 2017: Autoencoder-based feature learning for cyber security applications. *2017 International joint conference on neural networks (IJCNN)*, IEEE, 3854–3861.

[50] Yu, L., Z. Zhang, X. Xie, H. Chen, and J. Wang, 2019: Unsupervised feature selection using rbf autoencoder. *International Symposium on Neural Networks*, Springer, 48–57.

[51] Zhang, C., G. Wang, Y. Zhou, L. Yao, Z. L. Jiang, Q. Liao, and X. Wang, 2017: Feature selection for high dimensional imbalanced class data based on f-measure optimization. *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, IEEE, 278–283.

[52] Zhang, H., 2005: Exploring conditions for the optimality of naive bayes. *International Journal of Pattern Recognition and Artificial Intelligence*, **19**, no. 02, 183–198.

[53] Zheng, Z., X. Wu, and R. Srihari, 2004: Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter*, **6**, no. 1, 80–89.

[54] Zhu, Z.-B. and Z.-H. Song, 2010: Fault diagnosis based on imbalance modified kernel fisher discriminant analysis. *Chemical Engineering Research and Design*, **88**, no. 8, 936–951.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

80/2021 Sollini, M., Bartoli, F., Cavinato, L., Ieva, F., Ragni, A., Marciano, A., Zanca, R., Galli, L., Pai
[18F]FMCH PET/CT biomarkers and similarity analysis to refine the definition of oligometastatic prostate cancer

78/2021 Bucelli, M.; Dede', L.; Quarteroni, A.; Vergara, C.
Partitioned and monolithic algorithms for the numerical solution of cardiac fluid-structure interaction

79/2021 Ferraccioli, F.; Sangalli, L.M.; Finos, L.
Some first inferential tools for spatial regression with differential regularization

76/2021 Ponti, L.; Perotto, S.; Sangalli, L.M.
A PDE-regularized smoothing method for space-time data over manifolds with application to medical data

77/2021 Guo, M.; Manzoni, A.; Amendt, M.; Conti, P.; Hesthaven, J.S.
Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities

73/2021 Marcinno, F.; Zingaro, A.; Fumagalli, I.; Dede', L.; Vergara, C.
A computational study of blood flow dynamics in the pulmonary arteries

75/2021 Cicci, L.; Fresca, S.; Pagani, S.; Manzoni, A.; Quarteroni, A.
Projection-based reduced order models for parameterized nonlinear time-dependent problems arising in cardiac mechanics

74/2021 Orlando,G.; Barbante, P. F.; Bonaventura, L.
An efficient IMEX-DG solver for the compressible Navier-Stokes equations with a general equation of state

71/2021 Franco, N.; Manzoni, A.; Zunino, P.
A Deep Learning approach to Reduced Order Modelling of parameter dependent Partial Differential Equations

72/2021 Fresca, S.; Manzoni, A.
POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition