Neural Processing Letters (2019) 50:1937–1949
https://doi.org/10.1007/s11063-018-09977-1

# Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function

**Yuri Sousa Aurelio**[1] · **Gustavo Matheus de Almeida**[1] · **Cristiano Leite de Castro**[1] · **Antonio Padua Braga**[1]

Published online: 10 January 2019

**Abstract**

This paper presents a novel approach to deal with the imbalanced data set problem in neural networks by incorporating prior probabilities into a cost-sensitive cross-entropy error function. Several classical benchmarks were tested for performance evaluation using different metrics, namely G-Mean, area under the ROC curve (AUC), adjusted G-Mean, Accuracy, True Positive Rate, True Negative Rate and F1-score. The obtained results were compared to well-known algorithms and showed the effectiveness and robustness of the proposed approach, which results in well-balanced classifiers given different imbalance scenarios.

**Keywords** Multilayer perceptron · Imbalanced data · Classification problem · Back-propagation · Cost-sensitive function

## 1 Introduction

The number of samples commonly differs from one class to another in classification problems. This problem, known as the imbalanced data set problem [1–7], arises in most real-world applications. The point is that most current inductive learning principles resides on a sum of squared errors that do not take priors into account, which generally results in a classification bias towards the majority class.

One possible approach to handle this problem is to consider an alternative criterion to the overall learning error [4,8,9]. Other solution is the use of data resampling, which indirectly modifies the selection probability of the patterns during the learning phase. According to the Bayesian decision theory, the effect of changing the prior probabilities is analogous to set a new decision boundary for a probability-based classifier [10]. Many data resampling techniques have been proposed in the Literature, as for example, "Synthetic Minority Oversampling Technique" (SMOTE) [11], "Weighted Wilson's editing" (WWE) [12] and "Adaptive Synthetic Sampling" (ADASYN) [13]. However, it has been shown that the classifier performance depends on both an *ad hoc* parameter setting (eg, percentage of data to be

✉ Yuri Sousa Aurelio
yurisousa@ufmg.br

[1] Federal University of Minas Gerais, Belo Horizonte, MG, Brazil

under- or over-sampled in a class and scale of local neighborhood, to mention a few) and the choice of the classifier itself. Experiments in [8,14] suggested that well-known resampling techniques do not lead to performance improvement in Multi-Layer Perceptron (MLP) neural networks, even in the case of optimized parameter settings. Another solution is the use of ensemble learning, that has shown improvements for MLPs [15,16]. Ensemble extensions for imbalanced learning consider changes in the pattern probability function during the training phase. Such change has an effect on the model selection criterion, since the lowest overall error rate, as in Adaboost [17], gives way to a balanced decision among the classes accuracy rates given the consideration of the respective priors. Since the ensemble approach is based on a combination of different hypotheses (eg, MLPs neural networks), it usually leads to longer training times. This is the case especially when MLPs are used as weak learners.

This paper presents a novel approach to deal with the imbalanced data set problem in neural networks by incorporating prior probabilities into a cost-sensitive cross-entropy error function. The usual overall error formulation for MLPs is explicitly modified to incorporate unequal misclassification costs [18,19]. Unlike other cost-sensitive approaches in MLP learning [8,20], each class contribution in the cross-entropy error function is weighted by its respective class prior probability. This approach results on well-balanced decision boundaries.

The remainder of this paper is organized as follows. Section 2 describes the learning problem using the cross-entropy error function, and Sect. 3 presents the modified cost-sensitive cross-entropy error function by considering the prior probabilities of the classes. The methodology, experiments and results are shown and discussed in Sect. 4. Final considerations are given in Sect. 5.

## 2 The Learning Problem

In classification problems, considering the learning set $\mathbf{S} = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y} \,|\, i = 1, \ldots, N\}$, the output labels $\mathbf{y}_i$ given the inputs $\mathbf{x}_i$ are generated by an unknown function $f(\mathbf{x})$. The objective is to estimate it as close as possible by means of a model $f(\mathbf{x} \,|\, \theta)$, where $\theta$ is the parameter set. Instead of adopting an empirical risk, often based on the Mean Squared Error (MSE) metric, another way to estimate $\theta$ is through the cross-entropy error function (Eq. 1), where $\hat{y} = f(\mathbf{x} \,|\, \theta)$ is the model output, given the learning of the ($\mathcal{X}$,$\mathcal{Y}$)-mapping function.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \big[ - y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i) \big] \tag{1}$$

This function was chosen in place of the MSE one since it is convex and more suitable for calculating posterior probabilities in the case of neural networks [21]. When $y_i = 0$, it reduces to $-\log(1 - \hat{y}_i)$, and otherwise, with $y_i = 1$, to $-\log(\hat{y}_i)$. Whatever the case, the error decreases logarithmically as $\hat{y}_i$ tends to $y_i$ (Fig. 1). Moreover, since the curves are symmetrical, the error reduction happens at the same logarithmic rate for both classes (that is, $y_i = 0$ and $y_i = 1$). For a balanced learning problem, the error from $[-\log(\hat{y}_i)]$ will be proportional to that from $[-\log(1 - \hat{y}_i)]$, once each term will account for 50% of the total error $J(\theta)$ for a given model output $\hat{y}$. However, in the case of imbalanced data, the term associated to the majority class will have larger influence on $J(\theta)$. This occurs because the overall error, which is a sum of the individual terms, is minimized regardless of the class that generated the error.

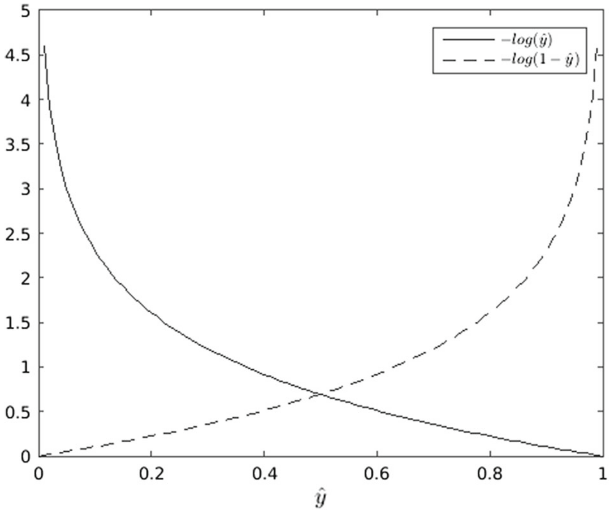

**Fig. 1** Illustration of the cross-entropy error function $J(\theta)$ for a range of model outputs $\hat{y}$ given $y_i = 1$ (solid line) and $y_i = 0$ (dashed line)

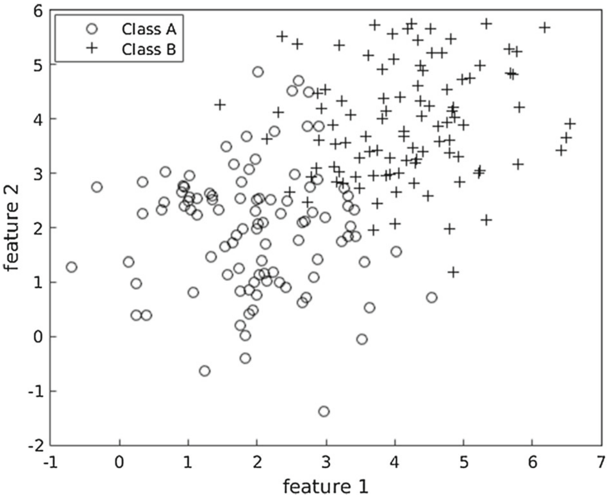

**Fig. 2** A two-Gaussian problem

As an example, consider a binary classification problem with classes A and B, each one containing 200 samples (Fig. 2). A MLP neural network with two inputs, two hidden neurons and one output, was then identified using the classical cross-entropy error function (Eq. 1). Also, consider imbalanced scenarios with class A having 5, 50 and 100 random samples among the original 200 instances. To evaluate the contribution of each class on the cross-entropy error function, the ratio $R$ (Eq. 2) was calculated along 1000 iterations. Figure 3 depicts the obtained results.

$$R = \frac{-\mathbf{y}\log(\hat{\mathbf{y}})}{-(1-\mathbf{y})\log(1-\hat{\mathbf{y}})} \tag{2}$$

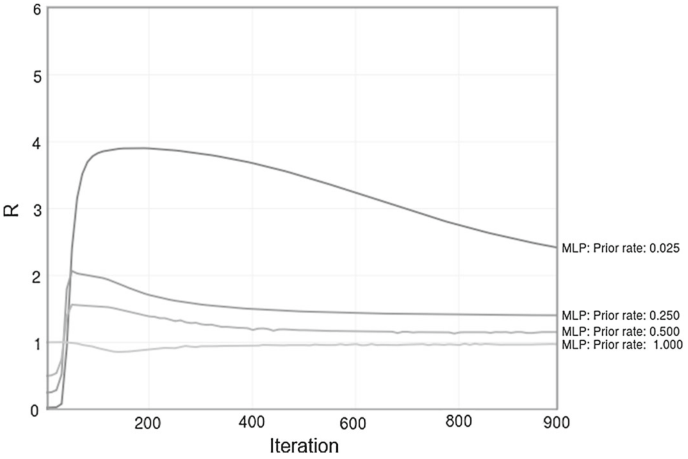

**Fig. 3** Cross-entropy imbalance ratio $R$ (Eq. 2) during learning of balanced and imbalanced data sets

It can be observed that $R$ is approximately constant when the prior probabilities of the classes are equal, since each class contributes equally to this ratio. The effect of the imbalance levels can also be observed; the greater it is, the more $R$ tends to stabilize at a higher value. The discrepancy between both priors is penalized in the computation of $R$ mainly in the initial iterations. This behavior led to the proposal of the cost-sensitive approach presented in the next section.

## 3 Cost-Sensitive Cross-Entropy Error Function Approach

The discrepancy of the error rates between balanced and imbalanced data may be treated considering the optimal decision rule given in Eq. 3 [22]. An approximate unit ratio between $[-\, y'^{(j)}_i \log(\hat{y}'^{(j)}_i)]$ and $[-\,(1 - y'^{(j)}_i) \log(1 - \hat{y}'^{(j)}_i)]$ is expected for balanced problems; however, as shown in Fig. 3, the ratio will reflect the priors. It also tends to stabilize in one, once the contribution of the minority class tends to become more influential as the number of iterations increases compensating the priors.

$$f_0(x) = \begin{cases} 1, & \text{if } \frac{p(\mathbf{x}|\mathbf{y}=1)}{p(\mathbf{x}|\mathbf{y}=0)} \geq \frac{p(\mathbf{y}=0)}{p(\mathbf{y}=1)} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

It is clear that the *prior* probabilities ratio $\frac{p(\mathbf{y}=0)}{p(\mathbf{y}=1)}$ plays an important role in the classification balance between classes, and then it could be used to compensate the imbalance. One way to accomplish this is to incorporate this ratio into the cross-entropy error function, as shown in Eqs. 4 and 5, where $N$ is the number of samples of the positive class ($y = 1$) and $M$ is the total number of samples.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \big[-\, y_i \log(\hat{y}_i)\lambda - (1 - y_i) \log(1 - \hat{y}_i)(1 - \lambda)\big] \tag{4}$$

$$\lambda^{(j)} = \left(\frac{N}{M}\right)^{-1} \tag{5}$$

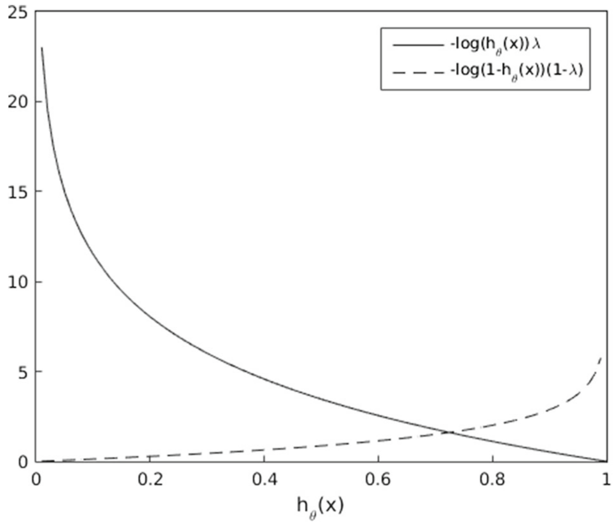

**Fig. 4** Minority class output obtained with the proposed approach based on the cross-entropy error function $J(\theta)$ for a range of model outputs $h_\theta(\mathbf{x})$ given $\mathbf{y_i} = 1$ (solid line) and $\mathbf{y_i} = \mathbf{0}$ (dashed line)

As an example, considering that $\frac{N^{\text{Class A(1)}}}{N^{\text{Class B(0)}}} = 0.20$, Fig. 4 shows that the magnitude of $[-\, y_i \log(\hat{y}_i)\lambda]$ decays faster than that of $[-\,(1 - y_i)\log(1 - \hat{y}_i)(1 - \lambda)]$. The gradient descent can also be applied to Eq. 4 in order to obtain $\partial J(\theta)/\partial\theta^{(n-1)}$ and $\partial J(\theta)/\partial z^{(n-1)}$, as shown by Equations from 6 to 12, where $n$ is the last layer of neurons and $\hat{y}_i = f(\mathbf{x}|\theta, z)$. Using the conventional cross-entropy error function (Eq. 1), it can be noted that changing only the error from the output layer, that is, $[\delta^{(n)} = g(z^{(n)}) - y$ by $\delta(n) = \big(qg(z^{(n)}) - \lambda y\big) + yg(z^{(n)})(\lambda - q)]$, is enough to meet all other equations. Applying this approach to the previous two Gaussian example (Sect. 2), Fig. 5 shows that the cross-entropy error rate (dashed line) is kept almost constant when considering the priors, which results in a balance between the classes. The gradient descent with the Rprop algorithm [23] was used in all experiments in the next section. Although the present approach is pattern-based, there is no constraint in the present formulation for a further matrix representation of the problem, as the one presented by [24].

$$\frac{\partial J(\theta)}{\partial\theta^{(n-1)}} = \Big[\Big(qg(z^{(n)}) - \lambda y\Big) + yg(z^{(n)})(\lambda - q)\Big] a^{(n-1)} \tag{6}$$

$$\frac{\partial J(\theta)}{\partial z^{(n-1)}} = \delta^{(n)}\theta^{(n-1)} g(z^{(n-1)})(1 - g(z^{(n-1)})) \tag{7}$$

$$\frac{\partial J(\theta)}{\partial z^{(n-1)}} = \delta^{(n-1)} \tag{8}$$

$$q = (1 - (N/M))^{-1} \tag{9}$$

$$qg(z^{(n)}) - \lambda y = \gamma \tag{10}$$

$$yg(z^{(n)})(\lambda - q) = \beta \tag{11}$$

$$\gamma + \beta = \delta \tag{12}$$

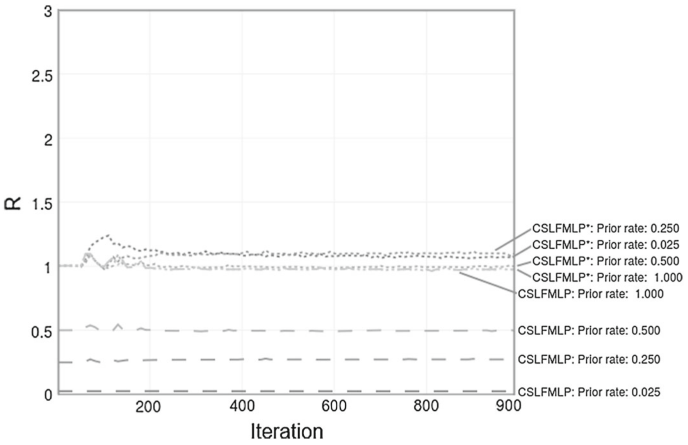

**Fig. 5** Cross-entropy rate for MLP

## 4 Results and Discussion

### 4.1 Experimental Study

An empirical study was conducted using 16 data sets from the UCI repository. The proposed cost-sensitive error function (CSEFMLP) was compared to five well-known techniques, namely SMOTE [11], weighted Wilson's editing (WWE) [12], Rprop [23], SMOTE + Tomek Links (SMTTL) [25], and RAMOBoost [26]. Table 1 depicts their main characteristics. After a preprocessing as in [8], twenty different trials were carried out for each data set by shuffling their original indexes. In sequence, they were split into training subset (70%), which was employed in a 7-fold cross-validation procedure for model selection, and test subset, used for performance evaluation. The following metrics were employed with this aim, namely the Kubat's G-Mean metric, that takes into account a balance between true positive and true negative rates given by $\sqrt{TPr \cdot TNr}$ [27]; the Area under the ROC curve (AUC), that also considers how well positive classes are ranked [28]; the Adjusted Geometric-Mean, a recent metric that proposes a balance between Specificity and Sensitivity favoring the latter [29]; the Accuracy, to show that it can be a trick metric if evaluated alone in unbalanced data problems; the True Positive Rate (TPR) and the True Negative Rate (TNR), that play an important role given the type of balance is being pursued by the model; and the F1 score, which is the harmonic mean of the precision and recall. The results for TPR and TNR were obtained considering the model with the greater accuracy.

### 4.2 Non-parametric Test

Model comparison procedures usually employ parametric tests. However, in this case, a non-parametric one is more adequate [30], namely the Nemenyi post-hoc statistical test ($F_F$) (Eq. 13), which is derived from the Friedman statistics ($\chi^2_F$) (Eq. 14) [31]. This test allows a simultaneous comparison of multiple classifiers ($L$) given multiple data sets ($M$). The null-hypothesis $H_0$ states that all algorithms perform similarly. In this case, they present equal average ranks ($R_j$), where $R_j = \frac{1}{M}\sum_{i=1}^{M} r_i^j$, $1 \leq j \leq L$, is the average rank of the $j$th algorithm given all data sets.

**Table 1** Characteristics of the data sets

| Data set | Alias | No. of features | $n_1$ | $n_2$ | $n_1/(n_1+n_2)$ |
|---|---|---|---|---|---|
| Ionosphere | iono | 34 | 126 | 225 | 0,359 |
| Pima Indians diabetes | pid | 08 | 268 | 500 | 0,349 |
| German credit | gmn | 24 | 300 | 700 | 0,3 |
| WP breast cancer | wpbc | 33 | 47 | 151 | 0,237 |
| Vehicle (4 vs. all) | veh | 18 | 199 | 647 | 0,235 |
| SPECTF heart | hrt | 44 | 55 | 212 | 0,206 |
| Segmentation (1 vs. all) | seg | 19 | 30 | 180 | 0,143 |
| Glass (7 vs. all) | gls7 | 10 | 29 | 185 | 0,136 |
| Euthyroid (1 vs. all) | euth | 24 | 238 | 1762 | 0,119 |
| Satimage (4 vs. all) | sat | 36 | 626 | 5809 | 0,097 |
| Vowel (1 vs. all) | vow | 10 | 90 | 900 | 0,091 |
| Abalone (18 vs. 9) | a18-9 | 08 | 42 | 689 | 0,057 |
| Yeast (9 vs. 1) | y9-1 | 08 | 20 | 463 | 0,041 |
| Car (3 vs. all) | car | 06 | 69 | 1659 | 0,04 |
| Yeast (5 vs. all) | y5 | 08 | 51 | 1433 | 0,034 |
| Abalone (19 vs. all) | a19 | 08 | 32 | 4145 | 0,008 |

$$F_F = \frac{(M-1)\chi_F^2}{M(L-1)-\chi_F^2} \tag{13}$$

$$\chi_F^2 = \frac{12M}{L(L+1)}\left(\sum_{j=1}^{L} R_j^2 - \frac{L(L+1)^2}{4}\right) \tag{14}$$

In case of rejection of the null hypothesis, another statistical test should be carried out to quantify the differences among the algorithms [30]. The most usual is the *Bonferroni-Dunn post-hoc* test [32]. Two classifiers are considered not similar if the difference between their average ranks is greater than a critical difference ($CD$; Eq. 15), where $q_\alpha$ is based on the Student statistic.

$$CD = q_\alpha\sqrt{\frac{L(L+1)}{6M}} \tag{15}$$

### 4.3 Results

Tables 2, 3, 4, 5, 6, 7 and 8 summarize the results for the considered metrics, namely G-Mean, AUC, Adjusted G-Mean, Accuracy, True Positive Rate (TPR), True Negative Rate (TNR) and F1 score, respectively. Given a data set, the highest classification score is highlighted in bold. The good performance of the proposed approach in comparison to well-known classifiers can be observed in general. For G-Mean, Adjusted G-Mean and TPR, it is the classifier that presents the highest number of best ratings, and for AUC, this number is similar to Rprop. For Accuracy, the scores are generally close to the highest classification scores mainly obtained from Rprop and RAMOboost, even though this metric is not a focus on unbalanced data problems. For TNR, a worse rating is expected since the proposed approach searches for a good balance between TPR and TNR, which led to a good average rank for TPR.

**Table 2** Average values for G-Mean

| Base | Rprop | SMOTE | SMTTL | WWE | RAMOBoost | CSEFMLP |
|---|---|---|---|---|---|---|
| iono | 85.64 | 87.49 | 88.65 | 85.55 | **89.75** | 85.56 |
| pid | 70.73 | 74.30 | 74.32 | 74.46 | 73.08 | **74.57** |
| gmn | 69.20 | 70.64 | 70.22 | 70.64 | 67.87 | **71.08** |
| wpbc | 64.44 | 66.76 | 64.37 | 63.25 | **68.53** | 67.34 |
| veh | 96.87 | 96.66 | 96.60 | 96.74 | **97.72** | 97.41 |
| hrt | 66.41 | 68.36 | 67.57 | **73.39** | 67.43 | 68.66 |
| seg | 99.57 | 99.44 | 99.68 | 99.51 | **99.76** | 99.37 |
| gls7 | 91.00 | 89.92 | 90.14 | **92.27** | 90.45 | 91.61 |
| euth | 89.99 | 91.21 | 91.23 | 91.32 | 89.42 | **91.73** |
| sat | 74.60 | 77.40 | 77.77 | 80.21 | 76.27 | **87.58** |
| vow | 97.82 | 97.24 | 98.27 | 98.04 | **99.30** | 98.65 |
| a18-9 | 74.17 | 84.37 | **84.58** | 76.64 | 74.61 | 83.77 |
| y9-1 | **74.27** | 70.96 | 70.08 | 73.57 | 74.10 | 73.40 |
| car | 94.87 | 93.04 | 91.78 | 96.75 | 95.85 | **99.19** |
| y5 | 51.36 | 77.93 | 78.52 | 66.46 | 63.94 | **79.72** |
| a19 | 14.18 | 75.89 | 75.90 | 25.08 | 41.13 | **76.96** |
| av. Rank | 4.56 | 4.00 | 3.56 | 3.25 | 3.44 | 2.19 |

**Table 3** Average values for AUC

| Base | Rprop | SMOTE | SMTTL | WWE | RAMOBoost | CSEFMLP |
|---|---|---|---|---|---|---|
| iono | 89.90 | 91.86 | 92.03 | 93.02 | **93.50** | 91.58 |
| pid | **82.88** | 82.63 | 82.65 | 82.70 | 80.54 | 82.79 |
| gmn | **78.49** | 77.71 | 77.56 | 78.20 | 74.27 | 78.43 |
| wpbc | 73.30 | 70.82 | 71.19 | 72.54 | **74.13** | 74.02 |
| veh | 99.43 | 98.87 | 99.36 | 98.75 | 99.25 | **99.76** |
| hrt | **81.58** | 77.96 | 77.54 | 80.65 | 79.32 | 78.70 |
| seg | 99.98 | 99.97 | **99.99** | 99.91 | 99.90 | 99.98 |
| gls7 | **95.90** | 95.61 | 95.28 | 95.09 | 93.81 | 95.08 |
| euth | 95.53 | 95.67 | 95.58 | 95.35 | 96.32 | **96.81** |
| sat | 92.50 | 91.98 | 92.27 | 92.85 | 93.18 | **94.52** |
| vow | 99.87 | 99.85 | 99.80 | 99.73 | 99.75 | **99.87** |
| a18-9 | **94.56** | 93.76 | 93.69 | 94.18 | 89.16 | 93.33 |
| y9-1 | 79.90 | 77.30 | 81.18 | 82.05 | **84.09** | 83.63 |
| car | **99.71** | 99.44 | 99.62 | 98.67 | 98.32 | 99.64 |
| y5 | 85.65 | 85.76 | 85.78 | **86.50** | 85.79 | 86.25 |
| a19 | 83.33 | 84.19 | 84.14 | 83.29 | 75.76 | **85.13** |
| av. Rank | 2.81 | 4.19 | 3.81 | 3.75 | 4.00 | 2.44 |

**Table 4** Average values for Adjusted G-Mean

| Base | Rprop | SMOTE | SMTTL | WWE | RAMOBoost | CSEFMLP |
|---|---|---|---|---|---|---|
| iono | 83.21 | 86.02 | 85.17 | 83.46 | **86.32** | 83.92 |
| pid | 65.95 | 75.39 | **75.46** | 73.72 | 71.14 | 74.29 |
| gmn | 63.99 | **72.61** | 71.75 | 68.63 | 67.09 | 72.37 |
| wpbc | 59.23 | 62.84 | 61.32 | 63.23 | 61.60 | **64.77** |
| veh | 96.58 | 96.52 | 95.56 | 96.69 | 97.08 | **97.55** |
| hrt | 64.71 | 65.40 | 54.93 | **75.27** | 65.10 | 64.29 |
| seg | 99.39 | 99.56 | 99.51 | **99.59** | 99.56 | 99.38 |
| gls7 | 86.93 | 89.41 | **89.72** | 89.38 | 86.65 | 87.54 |
| euth | 87.38 | 90.30 | 90.28 | 89.55 | 86.76 | **91.00** |
| sat | 67.27 | 81.51 | 80.81 | 74.46 | 74.10 | **87.10** |
| vow | 97.33 | 96.49 | 97.69 | 97.59 | **99.99** | 98.60 |
| a18-9 | 65.70 | 79.89 | 80.09 | 69.61 | 60.82 | **81.16** |
| y9-1 | 66.03 | 68.34 | **68.74** | 65.28 | 67.28 | 65.74 |
| car | 92.66 | 93.65 | 93.82 | 98.05 | 93.55 | **99.41** |
| y5 | 42.18 | 75.26 | 75.45 | 53.58 | 50.27 | **77.28** |
| a19 | 8.30 | 74.94 | 75.30 | 10.48 | 40.81 | **78.36** |
| av. Rank | 5.31 | 2.75 | 3.00 | 3.75 | 4.00 | 2.44 |

**Table 5** Average values for accuracy

| Base | Rprop | SMOTE | SMTTL | WWE | RAMOBoost | CSEFMLP |
|---|---|---|---|---|---|---|
| iono | 89.83 | 90.09 | 87.57 | 89.40 | **90.85** | 89.66 |
| pid | **75.66** | 73.20 | 73.01 | 75.20 | 73.32 | 74.69 |
| gmn | **74.82** | 68.44 | 68.41 | 73.17 | 71.47 | 69.61 |
| wpbc | 73.64 | 73.33 | 72.12 | 71.21 | **75.45** | 73.64 |
| veh | 97.09 | 96.91 | 97.06 | 96.42 | **97.62** | 97.16 |
| hrt | 77.64 | 75.28 | 75.17 | 74.16 | **78.54** | 75.06 |
| seg | 99.75 | 99.74 | 99.78 | 99.74 | **99.82** | 99.70 |
| gls7 | **95.71** | 95.14 | 95.00 | 94.57 | 94.29 | 95.00 |
| euth | **95.46** | 93.94 | 93.47 | 94.72 | 94.82 | 93.49 |
| sat | 92.61 | 92.12 | 91.98 | 91.72 | **93.47** | 88.03 |
| vow | 99.48 | 99.15 | 99.58 | 99.06 | **99.94** | 98.97 |
| a18-9 | **96.58** | 92.51 | 92.55 | 96.42 | 94.94 | 90.04 |
| y9-1 | 96.83 | 93.35 | 92.80 | 96.71 | **97.14** | 92.80 |
| car | 98.94 | 99.10 | 99.38 | 95.83 | **99.53** | 98.35 |
| y5 | **96.48** | 93.87 | 93.95 | 94.62 | 95.91 | 90.16 |
| a19 | **99.28** | 95.95 | 96.20 | **99.28** | 98.73 | 83.44 |
| av. Rank | 2.00 | 3.97 | 4.19 | 4.13 | 2.00 | 4.72 |

**Table 6** Average values for True Positive Rate

| Base | Rprop | SMOTE | SMTTL | WWE | RAMOBoost | CSEFMLP |
|---|---|---|---|---|---|---|
| iono | 77.74 | 82.14 | 82.74 | 80.95 | **83.33** | 76.31 |
| pid | 58.67 | **76.56** | **76.56** | 72.50 | 70.33 | 74.11 |
| gmn | 56.80 | 57.70 | 59.40 | **65.85** | 56.85 | 55.65 |
| wpbc | 44.67 | 48.00 | 47.33 | **52.67** | 47.67 | **52.67** |
| veh | 96.19 | 96.27 | 95.22 | 97.16 | 96.27 | **97.24** |
| hrt | 50.00 | 44.21 | 47.11 | **67.89** | 47.11 | 53.16 |
| seg | 99.45 | 99.36 | 99.55 | 99.27 | **99.64** | 99.05 |
| gls7 | 81.67 | 81.67 | 84.44 | **88.33** | 83.33 | 85.56 |
| euth | 84.31 | 85.63 | 80.63 | 87.25 | 79.13 | **89.81** |
| sat | 54.47 | 60.82 | 62.52 | 65.29 | 61.32 | **87.31** |
| vow | 96.17 | 95.50 | 95.17 | 96.50 | **99.17** | 98.50 |
| a18-9 | 55.00 | 53.21 | 54.29 | 60.36 | 49.64 | **73.57** |
| y9-1 | 56.67 | 47.50 | 45.00 | **57.50** | 53.33 | 54.17 |
| car | 86.96 | 86.96 | 88.48 | 97.61 | 91.52 | **100.00** |
| y5 | 4.71 | 32.65 | 44.41 | 32.65 | 29.71 | **60.88** |
| a19 | 0.00 | 12.00 | 10.00 | 0.00 | 3.50 | **58.50** |
| av. Rank | 4.66 | 3.97 | 3.69 | 2.53 | 3.75 | 2.41 |

**Table 7** Average values for True Negative Rate

| Base | Rprop | SMOTE | SMTTL | WWE | RAMOBoost | CSEFMLP |
|---|---|---|---|---|---|---|
| iono | 94.67 | 92.67 | 93.53 | 94.80 | 96.07 | **96.27** |
| pid | **85.57** | 71.12 | 71.42 | 76.66 | 72.32 | 75.15 |
| gmn | **83.53** | 72.80 | 73.59 | 75.85 | 78.25 | 75.60 |
| wpbc | 80.98 | 79.22 | 80.20 | 75.29 | **83.04** | 79.31 |
| veh | 97.49 | 97.09 | 97.88 | 96.49 | **98.05** | 97.19 |
| hrt | 84.64 | 84.64 | 83.07 | 74.50 | **88.14** | 81.29 |
| seg | 99.68 | 99.73 | 99.74 | 99.64 | **99.83** | 99.73 |
| gls7 | **97.38** | 96.39 | 96.64 | 95.25 | 96.39 | 96.80 |
| euth | 96.85 | 94.31 | 94.89 | 95.62 | **96.89** | 93.38 |
| sat | 96.18 | 95.09 | 95.23 | 94.61 | **96.95** | 88.16 |
| vow | 99.62 | 99.65 | 99.57 | 99.28 | **99.92** | 98.97 |
| a18-9 | **99.06** | 95.59 | 95.66 | 98.47 | 97.66 | 91.99 |
| y9-1 | 98.97 | 95.39 | 95.71 | 98.68 | **99.32** | 93.81 |
| car | 99.47 | 99.58 | 99.62 | 95.33 | **99.75** | 98.38 |
| y5 | **99.76** | 96.60 | 95.95 | 97.66 | 98.34 | 92.09 |
| a19 | **100.00** | 97.02 | 96.86 | **100.00** | 99.44 | 84.36 |
| av. Rank | 2.25 | 4.38 | 3.81 | 4.16 | 1.84 | 4.56 |

**Table 8** Average values for F1-score

| Base | Rprop | SMOTE | SMTTL | WWE | RAMOBoost | CSEFMLP |
|---|---|---|---|---|---|---|
| iono | 84.15 | 85.37 | 85.88 | 85.28 | **87.10** | 81.16 |
| pid | 63.51 | 66.97 | 66.75 | 67.16 | 64.33 | **67.68** |
| gmn | 58.91 | 59.93 | 59.65 | 58.74 | 56.28 | **60.37** |
| wpbc | 42.49 | 46.26 | 45.21 | 47.93 | 48.32 | **48.79** |
| veh | 92.97 | 90.60 | 90.17 | 93.69 | **95.62** | 90.49 |
| hrt | 51.09 | 51.05 | 49.86 | **54.46** | 52.04 | 52.65 |
| seg | 99.25 | 99.19 | 99.17 | 98.77 | **99.39** | 91.59 |
| gls7 | **83.2** | 81.94 | 81.71 | 80.27 | 81.49 | 82.99 |
| euth | **77.96** | 71.89 | 70.59 | 77.10 | 77.48 | 62.59 |
| sat | 61.96 | 62.56 | 60.81 | 63.14 | **63.60** | 48.46 |
| vow | 96.66 | 95.16 | 95.32 | 93.70 | **99.24** | 69.21 |
| a18-9 | 54.35 | 44.06 | 43.40 | **54.98** | 50.06 | 35.38 |
| y9-1 | **62.87** | 26.94 | 29.29 | 62.79 | 58.47 | 41.23 |
| car | 89.07 | 88.05 | 89.12 | 60.98 | **91.51** | 39.79 |
| y5 | 33.12 | 34.66 | 31.59 | **39.17** | 29.26 | 25.08 |
| a19 | 0.00 | 5.04 | **5.16** | 0.71 | 3.42 | 4.49 |
| av. Rank | 3.31 | 3.50 | 4.00 | 3.25 | 2.75 | 4.19 |

The average ranks ($R_j$) are depicted in the last rows of Tables 2, 3, 4, 5, 6, 7 and 8. The lower the value, the better. The proposed approach yields to the lowest values for the G-Mean, AUC, Adjusted G-Mean and TPR metrics. Next, the Nemenyi post-hoc statistical test was accomplished for an overall performance evaluation of the classifiers, with $M = 16$ (number of data sets) and $L = 6$ (number of classifiers). The test statistics ($F_F$) for the G-Mean, AUC, Adjusted G-Mean, Accuracy, True Positive Rate, True Negative Rate and F1-score metrics are, respectively, equal to 3.3206, 2.4818, 6.8324, 10.1638, 4.1453, 9.3699 and 1.2870. Given the critical value $F_{F;5;75;\alpha=0.01} = 1.9256$, except for F1-score, the null-hypothesis that all algorithms perform similarly was rejected. The Bonferroni-Dunn post-hoc test was then used for evaluation of the proposed approach (CSEFMLP). Table 9 shows the pairwise differences given all classifiers. Values beyond the critical difference ($CD = 1.5385$; $\alpha = 0.1$) are highlighted in bold. According to the G-Mean metric, the CSEFMLP classifier is significantly better than Rprop and SMOTE and slightly better than SMTLL, RAMOBoost and WWE. Regarding the AUC metric, it outperforms the SMOTE and RAMOBoost classifiers and is slightly better than SMTTL and WWE. For the Adjusted G-Mean metric, it is better than Rprop and RAMOBoost and slightly better than WWE. Given the Accuraccy metric, the CSEFMLP performs better than Rprop and RAMOBoost and slightly better than SMOTE. According to the True Positive Rate metric, CSEFMLP outperforms Rprop and SMOTE and is slightly better than RAMOBoost and SMTTL, and to the True Negative Rate metric, it is significantly better than RAMOBoost and Rprop and slightly better than SMTTL. And for the F1-score, CSEFMLP performs statistically equal to all classifiers. This set of results shows the efficiency and robustness of the proposed approach, which is based on the cost-sensitive cross-entropy error function, to handle unbalanced data problems.

**Table 9** Bonferroni-Dunn post-hoc test

| CSLFMLP versus | | | | | |
|---|---|---|---|---|---|
| Metric | Rprop | SMOTE | SMTTL | WWE | RAMOBoost |
| *G-Mean* | **2.3750** | **1.8125** | 1.3750 | 1.0625 | 1.25 |
| *AUC* | 0.3750 | **1.7500** | 1.3750 | 1.3125 | **1.5625** |
| *Adj G-Mean* | **2.8750** | 0.3125 | 0.5625 | 1.0625 | **1.5625** |
| *Accuracy* | **2.7188** | 0.7500 | 0.5312 | 0.5938 | **2.7188** |
| *True Positive Rate (TPR)* | **2.2500** | **1.5625** | 1.2812 | 0.1250 | 1.3438 |
| *True Negative Rate (TNR)* | **2.3125** | 0.1875 | 0.7500 | 0.4062 | **2.7188** |
| *F1-score* | 0.8750 | 0.6875 | 0.1875 | 0.9375 | 1.4375 |

## 5 Conclusion

This work proposes a new approach, called CSEFMLP (Cost-Sensitive Cross-Entropy Error Function for MLP neural networks), to handle the common unbalanced classification problem. This method generally performs better or at least similarly to well-known classifiers considering a set of performance metrics for unbalanced problems, namely G-Mean, AUC, Adjusted G-Mean, Accuraccy, True Positive Rate, True Negative Rate and F1-score. In short, the obtained results demonstrate that the proposed approach is able to deal with unbalanced data.

**Acknowledgements** The authors would like to thank the funding agencies CNPq, FAPEMIG and CAPES for their financial support.

## References

1. Chawla NV, Japkowicz N, Kotcz A (2004a) Special issue on learning from imbalanced data sets. SIGKDD Explor 6(1):1–6
2. Chawla N, Japkowicz N, Kolcz A (2004b) Special issue on learning from imbalanced data sets. In: Editorial of the ACM SIGKDD explorations newsletter
3. He H, Garcia E (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
4. López V, Fernández A, García S, Palade V, Herrera F (2013) An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf Sci 250:113–141
5. Bhowan U, Johnston M, Zhang M, Yao X (2013) Evolving diverse ensembles using genetic programming for classification with unbalanced data. IEEE Trans Evol Comput 17(3):368–386
6. Frasca M, Bertoni A, Re M, Valentini G (2013) A neural network algorithm for semi-supervised node label learning from unbalanced data. Neural Netw 43:84–98
7. Wang L, Yang B, Chen Y, Zhang X, Orchard J (2017) Improving neural-network classifiers using nearest neighbor partitioning. IEEE Trans Neural Netw Learn Syst 28(10):2255–2267
8. Castro CL, Braga AP (2013) Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. IEEE Trans Neural Netw Learn Syst 24(6):888–899
9. Oh SH (2011) A statistical perspective of neural networks for imbalanced data problems. Int J Contents 7(3):1–5
10. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
11. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 321–357
12. Barandela R, Valdovinos RM, Sánchez JS, Ferri FJ (2004) The imbalanced training sample problem: under or over sampling? In: Structural, syntactic, and statistical pattern recognition. Springer, pp 806–814

13. He H, Bai Y, Garcia EA, Li S (2008) Adasyn: adaptive synthetic sampling approach for imbalanced learning. In: IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). IEEE, pp 1322–1328
14. Khoshgoftaar TM, Van Hulse J, Napolitano A (2010) Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors. IEEE Trans Neural Netw 21(5):813–830
15. Chen S, He H, Garcia EA (2010) Ramoboost: ranked minority oversampling in boosting. IEEE Trans Neural Netw 21(10):1624–1642
16. Sun Y, Kamel MS, Wong AK, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 40(12):3358–3378
17. Schapire RE, Singer Y (1999) Improved boosting algorithms using confidence-rated predictions. Mach Learn 37(3):297–336
18. Kukar M, Kononenko I (1998) Cost-sensitive learning with neural networks. In: ECAI, pp 445–449
19. Elkan C (2001) The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence. Lawrence Erlbaum Associates Ltd, pp 973–978
20. Alejo R, García V, Sotoca JM, Mollineda RA, Sánchez JS (2007) Improving the performance of the rbf neural networks trained with imbalanced samples. In: Computational and ambient intelligence. Springer, pp 162–169
21. Kline DM, Berardi VL (2005) Revisiting squared-error and cross-entropy functions for training neural network classifiers. Neural Comput Appl 14(4):310–318
22. Berger JO (2010) Statistical decision theory and Bayesian analysis, 2nd edn. Springer, New York
23. Riedmiller M, Braun H (1993) A direct adaptive method for faster back propagation learning: the rprop algorithm. In: IEEE international conference on neural networks. IEEE, pp 586–591
24. Zhu C, Wang Z (2017) Entropy-based matrix learning machine for imbalanced data sets. Pattern Recognit Lett 88:72–80
25. Tomek I (1976) Two modifications of cnn. IEEE Trans Syst Man Cybern 6:769–772
26. Provost F, Fawcett T (2001) Robust classification for imprecise environments. Mach Learn 42(3):203–231
27. Kubat M, Matwin S (1997) Addressing the curse of imbalanced trainingsets: one-sided selection. In: ICML, Nashville, USA, vol 97, pp 179–186
28. Fawcett T (2006) An introduction to roc analysis. Pattern Recognit Lett 27(8):861–874
29. Batuwita R, Palade V (2012) Adjusted geometric-mean: a novel performance measure for imbalanced bioinformatics datasets learning. J Bioinform Comput Biol 10(04):1250003
30. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(Jan):1–30
31. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701
32. Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56(293):52–64