Sādhanā (2020) 45:174
https://doi.org/10.1007/s12046-020-01411-4

Simultaneous two-sample learning to address binary class imbalance problem in low-resource scenarios

SRI HARSHA DUMPALA, RUPAYAN CHAKRABORTY and SUNIL KUMAR KOPPARAPU*

TCS Research and Innovation – Mumbai, Yantra Park, Tata Consultancy Services Limited, Mumbai, Pokharan Road 2, Subhash Nagar, Thane (West) 400 601, Maharashtra, India
e-mail: sunilkumar.kopparapu@tcs.com

MS received 20 September 2019; revised 13 March 2020; accepted 30 March 2020

Abstract. Binary class imbalance problem refers to the scenario where the number of training samples in one class is much lower compared with the number of samples in the other class. This imbalance hinders the applicability of conventional machine learning algorithms to classify accurately. Moreover, many real world training datasets often fall in the category where data is not only imbalanced but also low-resourced. In this paper we introduce a novel technique to handle the class imbalance problem, even in low-resource scenarios. In our approach, instead of, as is common, learning using one sample at a time, two samples are simultaneously considered to train the classifier. The simultaneous two-sample learning seems to help the classifier learn both intra- and inter-class properties. Experiments conducted on a large number of benchmarked datasets demonstrate the enhanced performance of our technique over the existing state of the art techniques.

Keywords. Low-resource data; Class imbalance; Majority class; Minority class; Simultaneous learning; Binary classification.

1. Introduction

Class imbalance refers to the scenario where one class (minority class, C_-) is severely under-represented compared with the other class (majority class, C_+) in terms of the available number of training samples. This problem is often encountered in domains ranging from computational biology and bioinformatics [1, 2] to image annotation [3] and document classification [4]. For example, in call center conversations, *happy* emotion hardly exists in customer's speech, thus becomes a minority class and makes it difficult to automatically classify emotion in speech using conventional machine learning algorithms [5, 6]. These conventional algorithms, which assume a balanced distribution of data, fail to effectively learn the minority class characteristics when the training data is imbalanced [7–9].

Several approaches have been proposed in literature to tackle the class imbalance problem, as it attracted much attention from machine learning and data mining communities [10, 11]. One of the common approaches adopted to address class imbalance problem is to use sampling techniques [12], which involves preprocessing the imbalanced data either by undersampling the majority class samples or by generating new synthetic or repetitive samples for the minority class [13]. Ensemble-based methods [14–16] have also been popularly adopted in literature. These methods use boosting or bagging algorithms to train multiple classifiers on different, yet, balanced datasets obtained by adopting sampling techniques. As an alternative to sampling techniques, cost-sensitive learning-based methods have also been proposed; here different costs are assigned to the same degree of errors occurring due to majority and minority class samples [17].

In this paper, we propose a novel approach called simultaneous two-sample learning ([MC]s2s-MLP) to address the class imbalance problem [18]. In our approach, we simultaneously consider two arbitrary samples from the training dataset to train the classifier. We refer to this data representation as simultaneous two-sample (s2s) representation [19]. As a result, the number of samples available for training quadratically increases with s2s representation compared with the number of samples in the original training dataset. In order to handle the s2s data representation format, we modify the architecture of the classifier accordingly. In this work, a multi-layered perceptron (MLP) is used as the base classifier. Further, to test the classifier, we propose a mechanism of combining the test sample with a set of a priori known reference samples (class label known) to generate the required s2s representation format of the test sample. On these s2s represented test samples, majority-voting-based decision is obtained using only a single classifier. We show through extensive experiments that our approach performs significantly better than the state of the art techniques addressing class

*For correspondence
Published online: 04 July 2020

imbalance problem. We also show that the proposed approach outperforms other techniques when the imbalanced training data is also limited (low-resource). The main contribution of this work is (a) a data reorganization technique to address the class imbalance problem and (b) identifying a strategy to obtain majority-voting-based output by considering only a single classifier.

The rest of the paper is organized as follows: In section 2, a brief review of the approaches proposed to address the class imbalance problem is provided. Our approach to address class imbalance problem, which also works in low-resource scenarios, is explained in section 3. We explain the experimental framework along with the results in section 4. Finally, we provide the summary and conclusions in section 5.

2. Related work

Real world scenarios produce datasets that are very often imbalanced. Most of the approaches proposed to tackle the class imbalance problem, in literature, can be broadly classified into three categories, namely (1) data-level approaches, (2) algorithm-level approaches and (3) hybrid approaches [20, 21]. Approaches to tackle data imbalance problems are shown in table 1.

Data-level approaches include sampling methods to preprocess the data by either undersampling the majority class [13] or oversampling the minority class samples to remove imbalance in data distribution [12]. Majority-weighted minority oversampling technique (MWMOTE) [25] and critical synthetic minority oversampling technique (CSMOTE) [28] are the recent synthetic minority-oversampling-based techniques. MWMOTE generates synthetic samples using the weighted informative minority class samples, whereas CSMOTE generates synthetic minority samples by considering only the border and the edge minority class samples.

In algorithm-level approaches the conventional classification algorithms are fine-tuned to improve the learning task, especially relative to the minority class. Cost-sensitive learning-based methods and ensemble-based methods are examples of this approach [21]. In cost-sensitive learning-based methods, the cost function of the learning algorithm is modified by assigning a different misclassification cost to the errors associated with the minority and majority class samples [22]. In cost-sensitive multi-layer perceptron (CSMLP), the MLP is directly trained on the imbalanced data by assigning a higher cost of misclassification to the minority class [16, 17].

Table 1. Approaches to tackle data imbalance problems.

Data-level	Algorithm-level	Hybrid-level
Oversampling [12]	CSMLP [22]	llvotes [23]
Undersampling [13]	RUSBoost [15]	GSVM-RU [24]
MWMOT [25]	EUS-Boost [26]	UCML [27]
CSMOTE [28]		s2s(proposed)

Ensemble-based methods are another example of algorithm-level approaches, which consider a combination of multiple classifiers so as to improve the generalization ability and to increase the classifier performance [20]. RUSBoost [15] is an ensemble-based method, which uses random undersampling (RUS) as a preprocessing step for boosting algorithms. EUSBoost [26], based on RUSBoost, is an ensemble-based method, which uses evolutionary undersampling to enhance the performance of an ensemble of classifiers on highly imbalanced datasets.

Many hybrid approaches combining data and algorithm-level approaches have also been proposed. Most ensemble-based methods achieve enhanced performance by combining boosting/bagging algorithms with sampling techniques [23]. IIvotes [23] is an ensemble-based method, which integrates selective preprocessing combining filtering and oversampling of imbalanced data with the Ivotes ensemble. Granular Support Vector Machines-Repetitive Undersampling algorithm (GSVM-RU) [24] is another hybrid technique integrating granular support vector machine (SVM) learning with undersampling methods. Uncorrelated cost-sensitive multiset learning (UCML) [27], a recently proposed hybrid approach, is based on multiset feature learning, which uses undersampling for data preprocessing and an ensemble of classifiers for decision making.

Our approach to address class imbalance, even in low-resource scenarios, is motivated by co-ordination learning [29], in which an arbitrary pair of samples, belonging to different classes, are considered to train conventional classifiers such as SVMs. Co-ordination learning seems to help classifiers to learn the discriminative characteristics between classes (inter-class) better, compared with the conventional format of considering only one sample (intra-class) at a time to train the models [29]. Our approach differs from co-ordination learning in two major aspects: (a) we simultaneously consider two samples, irrespective of the class they belong to, to train the models and (b) the procedure followed to test the trained models. Our approach, a type of hybrid approach, differs from other class imbalance approaches. While most hybrid approaches either use sampling techniques for preprocessing the data or an ensemble of classifiers to obtain the output label or a combination of both [24, 27], our approach addresses class imbalance problem by simultaneously considering two samples to represent the data; this approach works even in low-resource conditions. Further, a decision mechanism is used to obtain majority-voting-based output label using only a single base classifier but not an ensemble of classifiers.

3. Proposed approach

The proposed approach to address the class imbalance problem, which also works in low-resource conditions, is explained in three steps, namely (1) data representation, (2) classifier training and (3) classifier testing. In this work we

use a modified MLP architecture, as shown in figure 1(b), as the base classifier instead of the traditional MLP architecture shown in figure 1(a). We refer to the modified MLP architecture as simultaneous two-sample MLP (s2s-MLP).

3.1 *Data representation*

Consider a two-class classification task with $\mathcal{C} = \{C_1, C_2\}$ denoting the set of class labels, and let N_1 and N_2 be the number of samples corresponding to C_1 and C_2 classes, respectively. In general, to train a traditional MLP (figure 1(a)), the samples in the training set are provided as an input–output pair as follows:

$$\{\vec{x}_{ij}{}^T, C_i\}, \quad i = 1, 2 \text{ and } j = 1, 2, ..., N_i, \tag{1}$$

where $\vec{x}_{ij} \in \mathbb{R}^{1\times d}$ is a d-dimensional feature vector representing the j^{th} sample in the i^{th} class, and $C_i \in \mathcal{C}$ refers to the i^{th} class label; T denotes the transpose of a vector. Note that when $N_1 \approx N_2$ it is a balanced class problem. In the proposed s2s data representation, we simultaneously consider two samples as follows:

$$\left\{ \begin{bmatrix} \vec{x}_{ij}^T \\ \vec{x}_{kl}^T \end{bmatrix}, \begin{bmatrix} C_i \\ C_k \end{bmatrix} \right\}, \ \forall\, i,k = 1,2 \text{ and } j(l) = 1, 2, ..., N_i(N_k) \tag{2}$$

where $\vec{x}_{ij}$ and $\vec{x}_{kl} \in \mathbb{R}^{1\times d}$ refer to the d-dimensional feature vectors representing the j^{th} sample in the i^{th} class and l^{th} sample in the k^{th} class, respectively. $[C_i\ C_k]^T$ refers to the output labels of i^{th} and k^{th} class, respectively.

As can be observed, in s2s representation, we have an input feature vector of length $2d$, namely $[\vec{x}_{ij}\vec{x}_{kl}] \in \mathbb{R}^{1\times 2d}$, and output class labels as either $[C_1C_1]^T$, $[C_1C_2]^T$, $[C_2C_1]^T$ or $[C_2C_2]^T$. Note that by representing the data in the s2s format, the total number of samples in the training set increases from $(N_1 + N_2)$ to $(N_1 + N_2)^2$ samples; this increased number of training samples, as will be seen later, helps the s2s-MLP classifier. Additionally, s2s representation can be seen to provide the classifier with a representation that allows learning the intra-class and inter-class variations [29].

For representing the class imbalance condition better we represent the two classes as C_+ and C_- where C_+ denotes the majority class and C_- denotes the minority class, and the number of samples in each class as $N_1 = N_+$ and $N_2 = N_-$ such that $N_+ \gg N_-$. In s2s-based data representation (2), the number of training samples generated by simultaneously considering two samples is $(N_+ + N_-)^2$.

Figure 2 captures the distribution of the number of samples across different combinations in the proposed s2s data representation. Let majority class samples be $N_+ = 75$ and minority class samples be $N_- = 25$. The four blocks, as shown in figure 2, represent the four possible combinations that can be obtained by simultaneously considering two samples at a time from the majority and minority class samples, namely, majority–majority, majority–minority, minority–majority and minority–minority class combinations with sizes $N_+ \times N_+ = 5625$, $N_+ \times N_- = 1875$, $N_- \times N_+ = 1875$ and $N_- \times N_- = 625$, respectively. Note that the total number of possible training samples generated using s2s representation is $(N_+ + N_-)^2 = 10000$ (an increase from $(N_+ + N_-) = 100$ that would be available traditionally) samples. Note that the total number of s2s samples carrying at least one majority class information (s2s_{N_+}) is $(N_+ \times N_+) + (N_+ \times N_-) + (N_- \times N_+) = 9375$ while the total number of s2s samples carrying at least one minority class information (s2s_{N_-}) is $(N_- \times N_-) + (N_+ \times N_-) + (N_- \times N_+) = 4375$. Note that, in the s2s representation the class imbalance ratio (IR) has reduced from originally 3 to 2.14, namely

$$\frac{N_+}{N_-} = 3 \longrightarrow \frac{\text{s2s}_{N_+}}{\text{s2s}_{N_-}} = 2.14. \tag{3}$$

However, the class imbalance still persists. Note that an IR equal to one represents an ideal balanced class.

To overcome the class imbalance, we constrain the number of majority–majority (training) sample combinations by combining each of the N_+ samples corresponding

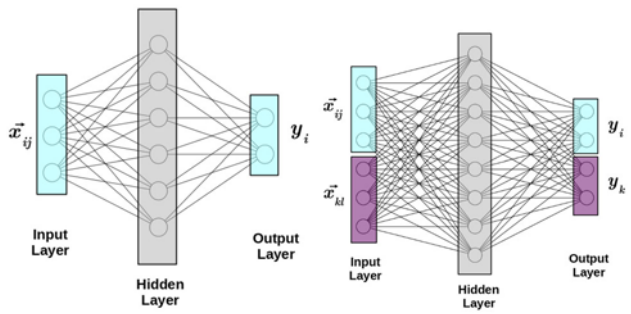

Figure 1. (a) Traditional MLP and (b) s2s-MLP architecture.

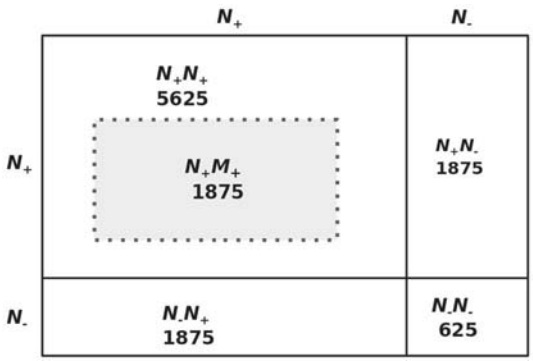

Figure 2. Distribution of class samples in s2s representation.

to class C_+ with only M_+ (where $M_+ = N_-$) randomly chosen samples corresponding to class C_+ (represented by the shaded portion bounded by a dotted box, labelled as N_+M_+, in figure 2). This constraint modifies the number of majority–majority training samples in the s2s data representation to $N_+ \times M_+ = 1875$, without affecting the number of samples in other class combinations. This constraint in s2s representation, called as majority–majority constrained s2s ([MC]s2s), results in a total of $[\text{MC}]\text{s2s}_{N_+} = (N_+ \times M_+) + (N_+ \times N_-) + (N_- \times N_+) = 5625$ samples carrying at least majority class information while maintaining a total of $[\text{MC}]\text{s2s}_{N_-} = \text{s2s}_{N_-} = 4375$ samples carrying at least one minority class information. Thus, the majority–majority constraint ([MC]) reduces the number of samples carrying majority class information and making the class IR

$$\frac{[\text{MC}]\text{s2s}_{N_+}}{[\text{MC}]\text{s2s}_{N_-}} = \frac{5625}{4375} = 1.28.$$

Note that the original IR of 3 (3) now is 1.28, which is achieved by [MC]s2s data representation. One can show (see Appendix I) that

$$\frac{[\text{MC}]\text{s2s}_{N_+}}{[\text{MC}]\text{s2s}_{N_-}} = \frac{3\left(\frac{N_+}{N_-}\right)}{\left(1 + 2\left(\frac{N_+}{N_-}\right)\right)}. \tag{4}$$

Clearly, IR of the [MC]s2s data representation (4) is always less than or equal to the initial IR (see figure 6), namely

$$\frac{[\text{MC}]\text{s2s}_{N_+}}{[\text{MC}]\text{s2s}_{N_-}} \leq \left(\frac{N_+}{N_-}\right). \tag{5}$$

Note that the [MC]s2s data representation borrows the advantages of the s2s data representation, namely (a) the increased number of training samples, (b) the ability of the samples to capture the intra- and inter-class properties (in the form of differences and similarities in the classes) and in addition (c) reduces the original class IR (5). These three important aspects seem to help [MC]s2s approach perform better than all the other state of the art approaches. This will be evident from our experiments.

3.2 *Classifier training*

MLP is one of the most common feed forward neural networks (FFNNs) that has been successfully used in various classification tasks. In this paper, we will consider MLP as the base classifier to validate our s2s data representation. To train MLP on s2s data representation, the s2s-MLP architecture, as shown in figure 1(b), is considered.

- Input layer has $2d$ linear units to accept the two samples simultaneously, namely, $[\vec{x}_{ij}\ \vec{x}_{kl}]$.
- The number of units in the hidden layer is selected empirically by varying the hidden units from 2 to $4d$ (twice the length of the input layer) and then selecting the number of units with the best performance on the validation set. Rectified linear units (ReLU) are used as the activation function for the hidden layers. We considered a single hidden layer in all our experiments.
- As can be observed from figure 1(b), the output layer has outputs $\vec{y}_i$ and $\vec{y}_k$ corresponding to the labels associated with the input feature vectors $\vec{x}_{ij}$ and $\vec{x}_{kl}$, respectively. Consequently, the output layer consists of twice the number of units considered in a traditional MLP architecture. Sigmoid activation function (not softmax) is used for the output units as the output labels.

Note that training s2s-MLP remains conceptually the same as that used to train a traditional MLP. The data represented in s2s format is directly provided as a training set to train the s2s-MLP. For a two-class classification problem there will be four units in the output layer, which can take a value from the set

$$\left\{ \begin{bmatrix}0\\1\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\0\\0\\1\end{bmatrix}, \begin{bmatrix}1\\0\\1\\0\end{bmatrix} \right\}$$

corresponding to the class labels

$$\left\{ \begin{bmatrix}C_+\\C_+\end{bmatrix}, \begin{bmatrix}C_+\\C_-\end{bmatrix}, \begin{bmatrix}C_-\\C_+\end{bmatrix}, \begin{bmatrix}C_-\\C_-\end{bmatrix} \right\},$$

respectively. For training an s2s MLP, we use the Adam algorithm with an initial learning rate of 0.001. Binary cross-entropy is used as the cost function. The batch size and other hyper-parameters are selected using the validation set.

3.3 *Classifier testing*

Generally, the feature vector corresponding to the test sample is provided as input to the trained MLP in the testing phase and the class label is decided based on the output obtained. In case of ensemble-based methods, the same test sample is provided as input to a bunch of classifiers and the final class label is obtained using majority voting on the outputs obtained from the ensemble of classifiers.

In [MC]s2s-MLP method, the feature vector corresponding to the test sample ($\vec{t}^T$) is converted to the s2s-based representation for testing the s2s-MLP. We obtain the s2s representation of a test sample by concatenating the test sample with a set of R pre-selected reference samples, whose class label is known a priori, as follows:

$$\begin{bmatrix} \vec{t} \\ \vec{r}_i \end{bmatrix}, i = 1, 2, ..., R \tag{6}$$

where $\vec{t} \in \mathbb{R}^{d\times 1}$, $\vec{r}_i \in \mathbb{R}^{d\times 1}$ refer to the d-dimensional feature vector corresponding to the test sample and i^{th} reference sample, respectively, and R refers to the number of reference samples considered. The pre-selected reference samples can belong to either majority or minority class.

Algorithm 1 Testing of S2S-MLP

1: **Given:**
R ▷ Number of reference samples;
$\{y_i^t, y_i^r\}_{i=1}^R$ ▷ Predicted label for test and the reference
$\{Y_i^r\}_{i=1}^R$ ▷ Ground truth of reference
C_+, C_- ▷ Class labels
2: **Determine:** $O^t \in C_+$ or C_-
3: **Initialize:** $N_+ \leftarrow 0$; $N_- \leftarrow 0$; $i \leftarrow 1$.

4: **while** $i \leq R$ **do**
5: **if** $(y_i^r == Y_i^r)$ **then**
6: **if** $(y_i^t = C_+)$ **then**
7: $N_+ \leftarrow N_+ + 1$
8: **else**
9: $N_- \leftarrow N_- + 1$
10: $i \leftarrow i + 1$

11: **if** $N_+ > N_-$ **then**
12: $O^t \in C_+$
13: **else**
14: $O^t \in C_-$

During s2s-MLP classifier testing, a test sample $\vec{t}$ is concatenated with R reference samples, say, $(\vec{r}_1, \vec{r}_2, ..., \vec{r}_R)$ to form R different instances of the same test sample $\vec{t}$, namely

$$\begin{bmatrix} \vec{t} \\ \vec{r}_1 \end{bmatrix}, \begin{bmatrix} \vec{t} \\ \vec{r}_2 \end{bmatrix}, ..., \begin{bmatrix} \vec{t} \\ \vec{r}_R \end{bmatrix}$$

to obtain the corresponding outputs, namely

$$\begin{bmatrix} y_1^t \\ y_1^r \end{bmatrix}, \begin{bmatrix} y_2^t \\ y_2^r \end{bmatrix}, ..., \begin{bmatrix} y_R^t \\ y_R^r \end{bmatrix}.$$

This output from s2s-MLP classifier is evaluated using Algorithm 1 to obtain the final output label $O^t \in \mathcal{C}$. As seen in Algorithm 1 we provide the outputs obtained from s2s-MLP, namely

$$\left\{ \begin{bmatrix} y_i^t \\ y_i^r \end{bmatrix} \right\}_{i=1}^R$$

along with ground-truth labels of the reference samples, namely, $\{Y_i^r\}_{i=1}^R$ as input to Algorithm 1. Note that only those s2s samples that have the reference sample classified correctly (see Line 5 in Algorithm 1) are used for determining the class of the test sample $\vec{t}$. The class label with maximum number of votes is considered as the output label O^t, i.e., if $N_+ > N_-$, then O^t is labelled as C_+, else O^t is given the label C_-. It is also important to note that the majority-voting-based decision is obtained using a single base classifier and not an ensemble of classifiers.

4. Experimental details

4.1 *Experimental framework*

Experiments are conducted on 20 benchmark binary-class imbalanced datasets, belonging to different domains, namely, computational biology and bioinformatics (Ecoli, Yeast, Haberman, Pima, NewThyroid and Abalone), document classification (Pageblk), sound classification (Vowel), image annotation (Vehicle) and object classification (Glass, Shuttle). All datasets (5-fold partitioned) are obtained from Keel dataset repository [30, 31]. Table 2 summarizes the details of the datasets represented traditionally, and using [mc]s2s-based representation. The datasets are listed in the increasing order of their IR. In this work, we refer to IR as the ratio of the number of samples carrying majority class information to the number of samples carrying minority class information. Datasets with IR between 1.5 and 9 are considered as relatively low imbalanced datasets, and datasets with IR above 9 are considered as high imbalanced datasets [31].

In table 2 the column d refers to the number of feature attributes representing each input sample; N_+ and N_- refer to the number of samples carrying majority and minority class information, respectively, in the traditional data representation. $[\text{mc}]\text{s2s}_{N_+}$ and $[\text{mc}]\text{s2s}_{N_-}$ refer to the number of samples carrying majority class information and minority class information in [mc]s2s data representation, respectively. Therefore IR in traditional data representation is computed as (N_+/N_-), and IR is computed as $([\text{mc}]\text{s2s}_{N_+}/[\text{mc}]\text{s2s}_{N_-})$ in [mc]s2s representation. We can observe from table 2 that in traditional representation, the majority class samples (N_+) heavily outnumber the minority class samples (N_-) in almost all the datasets. However, in [mc]s2s representation, $[\text{mc}]\text{s2s}_{N_+}$ and $[\text{mc}]\text{s2s}_{N_-}$ are comparable. Note that the actual number of samples corresponding to minority class is very low (for example just 9 minority class samples in case of Glass5) in most of the datasets (which makes it difficult for the classifiers to learn the characteristics of the minority class) but they have an increased representation in [mc]s2s representation.

For each dataset, we use 5-fold (the folds as provided in the Keel dataset repository are directly used) cross-validation approach to compare the performance of all the methods considered for analysis. In Keel dataset

Table 2. Imbalanced datasets used in our experimentation.

Low imbalanced datasets

		Traditional data representation			[MC]s2s data representation		
Dataset	d	N_+	N_-	$\left(\frac{N_+}{N_-}\right)$	[MC]s2s$_{N_+}$	[MC]s2s$_{N_-}$	$\frac{\text{[MC]s2s}_{N_+}}{\text{[MC]s2s}_{N_-}}$ (4)
Glass1	9	138	76	1.82	31464	26752	1.176
Ecoli0vs1	7	143	77	1.86	33033	27951	1.181
Pima	8	500	268	1.90	402000	339824	1.183
Glass0	9	144	70	2.01	30240	25080	1.206
Vehicle3	18	606	240	2.52	436320	348480	1.252
Haberman	3	222	84	2.68	55944	44352	1.261
Vehicle0	18	647	199	3.23	386259	297107	1.300
Ecoli1	7	259	77	3.36	59829	45815	1.306
NewThyroid1	5	180	35	5.14	18900	13825	1.367
Yeast3	8	1321	163	8.11	645969	457215	1.413

High imbalanced datasets

		Traditional data representation			[MC]s2s data representation		
Dataset	d	N_+	N_-	$\left(\frac{N_+}{N_-}\right)$	[MC]s2s$_{N_+}$	[MC]s2s$_{N_-}$	$\frac{\text{[MC]s2s}_{N_+}}{\text{[MC]s2s}_{N_-}}$ (4)
Vowel0	13	899	89	10.1	240033	167943	1.429
Ecoli4	7	313	23	13.84	21597	14927	1.447
Shuttle0vs4	9	1706	123	13.87	629514	434805	1.448
Pageblk	10	444	28	15.85	37296	25648	1.454
Abalone9vs18	8	690	41	16.68	84870	58261	1.457
Shuttle2vs4	9	409	20	20.5	24540	16760	1.464
Glass5	9	205	9	22.81	5535	3771	1.468
Yeast5	8	1440	44	32.78	190080	128656	1.477
Yeast6	8	1449	35	39.15	152145	102655	1.482
Abalone	8	4142	32	128.87	397632	266112	1.494

repository, each fold is obtained by randomly selecting the samples from the dataset but maintaining the same IR value across all folds as in the original dataset and there is no overlap between any of the two folds. Hence, at any time 80% of the data is used for training (75% as training set and 5% as validation set) and remaining 20% of the data is used for testing. The validation set is used for selecting network architecture and for hyper-parameter tuning. It is to be noted that the s2s representation is obtained separately on the train, validation and test sets. Majority–majority constrained [MC]s2s data representation is obtained on training set considering $M_+ = N_-$; for the test set, majority voting is used where $R = 20$ randomly selected majority class samples from the training set are used as the pre-selected reference set. The choice of R is obtained empirically by experimenting with different values of R. The performance is measured in terms of F-measure, namely

$$F_\beta = (1+\beta^2) \times \frac{\text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}. \quad (7)$$

In this paper we will consider the widely used F_1 ($\beta = 1$) and F_2 ($\beta = 2$) as performance metrics, as previously used in [27, 32]. The higher the values of F_β, the better the performance of the system. F_β values reported in this paper are the mean scores obtained from the 5-fold cross-validation.

4.2 *Experimental results*

The F_1 and F_2 values obtained by conventional MLP, cost-sensitive MLP (CSMLP) [16] and [MC]s2s-MLP (proposed approach) on low and high imbalanced datasets are provided in tables 3 and 4, respectively. It can be observed from tables 3 and 4 that [MC]s2s-MLP outperforms both MLP and CSMLP across all the class-imbalanced datasets in terms of both F_1 and F_2 metrics. This validates that the [MC]s2s data representation (increased number of training samples, decrease in IR, enabling ability to capture intra- and inter-class information) does help in better modelling the classifier in class imbalance scenarios. The performance of the proposed [MC]s2s-based approach is comprehensively compared with several state of the art imbalanced data classification techniques mentioned in table 1 in Appendix Appendix II.

Table 3. Performance of MLP, CSMLP and [MC]s2s-MLP on low imbalanced datasets.

	F_1			F_2		
Dataset	MLP	CSMLP	[MC]s2s-MLP	MLP	CSMLP	[MC]s2s-MLP
Glass1	.55 ± .04	.68 ± .06	*.76 ± .05*	.52 ± .03	.65 ± .06	*.75 ± .05*
Ecoli0vs1	.97 ± .02	.98 ± .02	*1*	.97 ± .02	.98 ± .02	*1*
Pima	.63 ± .03	.66 ± .03	*.69 ± .03*	.56 ± .02	.68 ± .05	*.71 ± .04*
Glass0	.51 ± .02	.66 ± .04	*.68 ± .03*	.48 ± .01	*.76 ± .07*	.75 ± .04
Vehicle3	.58 ± .04	.64 ± .05	*.76 ± .04*	.54 ± .05	.62 ± .05	*.74 ± 0.05*
Haberman	.33 ± .09	.48 ± .07	*.55 ± .05*	.39 ± 0.10	.45 ± .06	*.52 ± .05*
Vehicle0	.91 ± .02	.92 ± .03	*.96 ± .02*	.90 ± .02	.96 ± .03	*.98 ± .03*
Ecoli1	.70 ± .03	.76 ± .03	*.79 ± .03*	.67 ± .03	.82 ± .06	*.86 ± .05*
NewThyroid1	.92 ± .03	.94 ± .03	*.96 ± .03*	.91 ± .03	.95 ± .02	*.96 ± .02*
Yeast3	.60 ± .04	.68 ± .02	*.72 ± .02*	.55 ± .03	.78 ± .05	*.83 ± .05*

Table 4. Performance of MLP, CSMLP and [MC]s2s-MLP on high imbalanced datasets.

	F_1			F_2		
Dataset	MLP	CSMLP	[MC]s2s-MLP	MLP	CSMLP	[MC]s2s-MLP
Vowel0	.92 ± .03	.94 ± .03	*.96 ± .02*	.93 ± .04	.93 ± .03	*.95 ± .03*
Ecoli4	.72 ± .04	.77 ± .04	*.81 ± .04*	.70 ± .05	.76 ± .04	*.79 ± .05*
Shuttle0vs4	.98 ± .02	.99 ± .02	*1*	.97 ± .02	.99 ± .02	*1*
Pageblk	.73 ± .04	.88 ± .03	*.95 ± .03*	.56 ± .03	.89 ± .07	*.96 ± .04*
Abalone9vs18	.51 ± .04	.58 ± .04	*.61 ± .05*	.49 ± .05	.60 ± .05	*.63 ± .05*
Shuttle2vs4	.92 ± .04	.93 ± .04	*.94 ± .03*	.90 ± 0.05	.92 ± .04	*.94 ± .03*
Glass5	.67 ± .05	.76 ± .04	*.88 ± .02*	.58 ± .03	.86 ± .08	*.93 ± .04*
Yeast5	.49 ± .04	.57 ± .03	*.69 ± .02*	.42 ± .02	.73 ± .07	*.79 ± .06*
Yeast6	.21 ± .03	.27 ± .04	*.44 ± .03*	.16 ± .02	.41 ± .06	*.58 ± .06*
Abalone	0 ± 0	.06 ± .01	*.08 ± .01*	0 ± 0	.09 ± .02	*.15 ± .03*

4.3 *Effect of training data size on classification of imbalanced data*

While [MC]s2s-MLP performs well on imbalanced datasets irrespective of the IR, we wanted to verify its performance when there is very limited training (low-resource) data. We evaluate the performance of [MC]s2s-MLP in low-resource conditions on imbalanced datasets. As before, all experimental results are validated using 5-fold cross-validation (75% of data for training, 5% of data for validation and 20% for testing in each fold).

Further, to analyse the effectiveness of [MC]s2s-MLP in low-resource imbalanced scenarios, different proportions of training data, within each fold, are considered to train the system. For this analysis we considered 4 different proportions, namely (1/4), (2/4), (3/4) and (4/4) of the training data to train the classifier. For instance (1/4) means considering only one-fourth (low-resource) of the original training data to train the classifier, and (4/4) means considering the complete training data during training. Note that the [MC]s2s-based representation is obtained separately on the training, validation and test sets. [MC]s2s-based data representation is obtained on training set, and for testing, as in earlier set of experiments, $R = 20$ randomly selected majority class samples from the training set are used as the preselected reference set in each case.

Experimental results in terms of F_1 and F_2 scores are presented in tables 5 and 6, respectively. It can be observed that the [MC]s2s approach outperforms all the state of the art methods across all datasets, including low-resource (observe the rows marked as "Low" in tables 5 and 6) conditions. The performance of [MC]s2s-MLP (in terms of both F_1 and F_2 scores) is much higher compared with other approaches, especially when the training data is limited, even for datasets with high imbalance.

Figures 3 and 4 capture the F_1 performance of all the 10 approaches, including the proposed [MC]s2s-MLP, on Glass5 dataset. The box plot in figure 3 clearly demonstrates the superior performance of [MC]s2s-MLP over other 9 approaches. Observe that the right most box (index 10) in all the four plots (figure 3(a)–(d)) has a higher F_1 compared with all the other techniques (index 1–9). It can further be observed that the quantum of improvement (measured as the distance between the [MC]s2s-MLP and the next best technique) in performance is much better for

Table 5. F_1 values obtained by considering different proportions of training data.

Dataset	MLP	MWM	CSM	CMLP	RUSB	EUSB	IIvotes	GSVM	UCML	[MC]s2s-MLP
VEHICLE0 (4/4)	.91 ± .02	.94 ± .03	.93 ± .03	.92 ± .03	.92 ± .02	.91 ± .04	.92 ± .03	.93 ± .03	.91 ± .02	*.96 ± .02*
1/4 (LOW)	.83 ± .03	.84 ± .05	.84 ± .04	.83 ± .05	.83 ± .04	.81 ± .04	.81 ± .06	.87 ± .03	.82 ± .03	*.92 ± .03*
2/4	.87 ± .05	.89 ± .06	.88 ± .04	.91 ± .04	.86 ± .03	.86 ± .03	.84 ± .04	.91 ± .03	.89 ± .03	*.94 ± .02*
3/4	.90 ± .03	.93 ± .03	.91 ± .04	.92 ± .03	.90 ± .02	.89 ± .03	.89 ± .03	.92 ± .03	.90 ± .02	*.95 ± .02*
ECOLI1 (4/4)	.70 ± .03	.74 ± .03	.75 ± .03	.76 ± .03	.74 ± .04	.76 ± .04	.78 ± .04	.78 ± .03	.77 ± .03	*.79 ± .03*
1/4 (LOW)	.63 ± .04	.69 ± .08	.71 ± .07	.73 ± .06	.72 ± .09	.73 ± .07	.72 ± .07	.70 ± .06	.71 ± .07	*.75 ± .06*
2/4	.67 ± .03	.71 ± .09	.72 ± .05	.74 ± .04	.72 ± .06	.73 ± .06	.75 ± .05	.73 ± .04	.73 ± .05	*.77 ± .04*
3/4	.68 ± .03	.73 ± .05	.74 ± .03	.75 ± .03	.73 ± .05	.74 ± .04	.77 ± .04	.75 ± .04	.75 ± .03	*.78 ± .04*
YEAST3 (4/4)	.60 ± .04	.69 ± .03	.65 ± .03	.68 ± .02	.66 ± .03	.67 ± .04	.70 ± .03	.68 ± .03	.66 ± .03	*.72 ± .02*
1/4 (LOW)	.50 ± .05	.62 ± .08	.55 ± .08	.64 ± .06	.59 ± .07	.62 ± .07	.60 ± .08	.63 ± .04	.61 ± .05	*.67 ± .05*
2/4	.54 ± .04	.65 ± .08	.59 ± .07	.66 ± .05	.61 ± .06	.65 ± .05	.65 ± .06	.64 ± .04	.63 ± .03	*.69 ± .03*
3/4	.58 ± .05	.68 ± .05	.63 ± .04	.67 ± .02	.63 ± .03	.66 ± .03	.68 ± .04	.66 ± .03	.65 ± .03	*.70 ± .03*
PAGEBLK(4/4)	.73 ± .04	.88 ± .03	.84 ± .02	.88 ± .03	.86 ± .03	.87 ± .02	.88 ± .05	.89 ± .02	.91 ± .03	*.95 ± .03*
1/4 (LOW)	.28 ± .06	.53 ± .09	.50 ± .09	.39 ± .07	.29 ± .07	.33 ± .06	.52 ± .09	.56 ± .08	.55 ± .07	*.73 ± .06*
2/4	.53 ± .05	.68 ± .10	.63 ± .09	.64 ± .07	.58 ± .08	.66 ± .05	.66 ± .07	.64 ± .05	.68 ± .04	*.81 ± .04*
3/4	.65 ± .05	.79 ± .05	.75 ± .05	.78 ± .05	.75 ± .05	.79 ± .03	.74 ± .05	.78 ± .03	.81 ± .03	*.89 ± .03*
GLASS5 (4/4)	.67 ± .05	.75 ± .03	.72 ± .03	.76 ± .04	.70 ± .03	.76 ± .04	.60 ± .05	.75 ± .03	.81 ± .03	*.88 ± .02*
1/4 (LOW)	0.0 ± 0.0	.18 ± .11	.30 ± .11	.29 ± .10	.12 ± .08	.31 ± .10	.13 ± .10	.39 ± .09	.37 ± .10	*.68 ± .07*
2/4	.30 ± .06	.42 ± .09	.41 ± .08	.60 ± .06	.45 ± .04	.61 ± .06	.35 ± .09	.59 ± .06	.60 ± .09	*.79 ± .06*
3/4	.58 ± .05	.68 ± .05	.66 ± .04	.74 ± .06	.56 ± .04	.68 ± .05	.50 ± .06	.68 ± .04	.74 ± .05	*.86 ± .03*
YEAST5 (4/4)	.49 ± .04	.56 ± .03	.55 ± .03	.57 ± .03	.54 ± .03	.57 ± .03	.54 ± .05	.60 ± .03	.58 ± .02	*.69 ± .02*
1/4 (LOW)	.28 ± .05	.37 ± .07	.40 ± .09	.36 ± .07	.32 ± .10	.34 ± .08	.38 ± .13	.38 ± .04	.37 ± .08	*.45 ± .05*
2/4	.39 ± .05	.43 ± .06	.44 ± .07	.44 ± .03	.38 ± .07	.42 ± .07	.43 ± .09	.47 ± .04	.46 ± .07	*.54 ± .03*
3/4	.46 ± .04	.49 ± .04	.49 ± .04	.52 ± .04	.44 ± .04	.49 ± .04	.48 ± .06	.55 ± .03	.54 ± .04	*.63 ± .03*
YEAST6 (4/4)	.21 ± .03	.25 ± .02	.28 ± .03	.27 ± .04	.28 ± .03	.32 ± .03	.34 ± .03	.30 ± .03	.34 ± .04	*.44 ± .03*
1/4 (LOW)	.13 ± .02	.15 ± .07	.13 ± .10	.19 ± .06	.18 ± .09	.15 ± .09	.17 ± .09	.22 ± .06	.21 ± .05	*.34 ± .05*
2/4	.15 ± .03	.16 ± .05	.20 ± .06	.22 ± .05	.20 ± .04	.18 ± .05	.23 ± .05	.24 ± .04	.26 ± .04	*.38 ± .04*
3/4	.19 ± .03	.20 ± .04	.25 ± .04	.24 ± .05	.25 ± .04	.24 ± .03	.29 ± .03	.27 ± .04	.29 ± .04	*.42 ± .03*
ABALONE(4/4)	0.0 ± 0.0	.05 ± .01	.04 ± .01	.06 ± .01	.04 ± .01	.06 ± .01	.04 ± .03	.06 ± .01	.07 ± .01	*.08 ± .01*
1/4 (LOW)	0.0 ± 0.0	.01 ± .08	0.0 ± 0.0	.01 ± .09	0.0 ± 0.0	.01 ± .09	.01 ± .12	.02 ± .07	.02 ± .08	*.03 ± .08*
2/4	0.0 ± 0.0	.03 ± .02	.01 ± .03	.02 ± .04	.02 ± .02	.04 ± .05	.02 ± .07	.03 ± .06	.03 ± .05	*.04 ± .05*
3/4	0.0 ± 0.0	.04 ± .02	.02 ± .03	.05 ± .03	.03 ± .02	.06 ± .02	.03 ± .04	.05 ± .03	.05 ± .03	*.06 ± .02*

low-resource condition (see figure 3(a)) compared with that when all the data (see figure 3(d)) is used for training. This demonstrates that the proposed [MC]s2s-MLP system not only performs well in imbalance data scenario but also it is able to perform much better than other state of the art approaches when the training data is especially small (low-resource scenario).

Figure 4 compares performance of [MC]s2s-MLP on GLASS5 dataset by varying the proportion of training data used for training from 10% to 100% in steps of 10%. It can be observed from the plot that [MC]s2s-MLP is consistently better than other state of the art techniques, irrespective of the amount of training data used to train the system. Even with 23 training samples (22 majority and 1 minority sample, namely, when only 10% of training data is used), [MC]s2s-MLP shows an improved performance compared with all other methods (observe that the F_1 score of [MC]s2s-MLP at 10% is 0.53 compared with the second best score of 0.23 obtained by GSVM), signifying the effectiveness of [MC]s2s-MLP in low-resource imbalanced data scenarios. It is evident that s2s data representation enables [MC]s2s-MLP to perform effectively in low-resource imbalanced data scenario.

The enhanced performance of [MC]s2s-MLP compared with several previous approaches, particularly low-resource scenarios, may be attributed to the s2s data representation. It can be hypothesized that s2s data representation seems to help the classifier to learn not only the class-specific (intra-class) information (as in the case in a traditional data representation) but also the "similarities and differences" (inter-class information) between the two classes. Moreover, the quadratic increment in the number of training samples as in the s2s representation also seems to help the classifier to learn the discriminative inter-class characteristics better.

4.4 *Significance of majority–majority constraint*

In the [MC]s2s-MLP approach, constraining the majority–majority combinations ([MC]s2s) is one of the main component. As shown in (4), the imposition of the majority–

Table 6. F_2 values obtained by considering different proportions of training data.

Dataset	MLP	MWM	CSM	CMLP	RUSB	EUSB	IIvotes	GSVM	UCML	[MC]s2s-MLP
Vehicle0 (4/4)	.90 ± .02	.96 ± .03	.96 ± .04	.96 ± .03	.95 ± .04	.94 ± .05	.95 ± .05	.95 ± .03	.93 ± .03	*.98 ± .03*
1/4 (Low)	.85 ± .04	.87 ± .05	.87 ± .04	.89 ± .03	.89 ± .03	.87 ± .04	.86 ± .06	.90 ± .03	.88 ± .03	*.94 ± .03*
2/4	.88 ± .05	.90 ± .06	.89 ± .04	.93 ± .04	.92 ± .03	.90 ± .03	.89 ± .04	.92 ± .03	.91 ± .03	*.95 ± .02*
3/4	.89 ± .02	.94 ± .03	.92 ± .04	.95 ± .03	.94 ± .02	.92 ± .03	.92 ± .03	.94 ± .03	.92 ± .02	*.97 ± .02*
Ecoli1 (4/4)	.67 ± .03	.78 ± .05	.81 ± .06	.82 ± .06	.80 ± .05	.79 ± .05	.80 ± .05	.85 ± .05	.83 ± .05	*.86 ± .05*
1/4 (Low)	.60 ± .07	.72 ± .08	.73 ± .07	.76 ± .05	.75 ± .07	.74 ± .06	.74 ± .07	.77 ± .06	.73 ± .07	*.79 ± .06*
2/4	.64 ± .08	.74 ± .09	.75 ± .05	.79 ± .04	.76 ± .06	.75 ± .06	.76 ± .05	.79 ± .04	.74 ± .05	*.82 ± .04*
3/4	.65 ± .08	.75 ± .05	.77 ± .03	.80 ± .03	.78 ± .05	.78 ± .04	.79 ± .04	.82 ± .04	.78 ± .03	*.83 ± .04*
Yeast3 (4/4)	.55 ± .03	.81 ± .07	.74 ± .06	.78 ± .05	.75 ± .06	.76 ± .05	.77 ± .06	.79 ± .07	.78 ± .06	*.83 ± .05*
1/4 (Low)	.46 ± .08	.64 ± .09	.57 ± .10	.69 ± .09	.65 ± .08	.64 ± .08	.59 ± .09	.69 ± .08	.67 ± .08	*.76 ± .08*
2/4	.49 ± .04	.69 ± .08	.63 ± .07	.72 ± .08	.69 ± .07	.68 ± .10	.67 ± .08	.72 ± .07	.71 ± .07	*.78 ± .05*
3/4	.53 ± .05	.76 ± .06	.69 ± .09	.76 ± .06	.72 ± .06	.73 ± .07	.73 ± .08	.75 ± .08	.74 ± .06	*.80 ± .04*
Pageblk(4/4)	.56 ± .03	.88 ± .06	.85 ± .06	.89 ± .07	.87 ± .06	.89 ± .06	.89 ± .06	.91 ± .05	.92 ± .04	*.96 ± .04*
1/4(Low)	.30 ± .12	.59 ± .11	.58 ± .11	.60 ± .08	.49 ± .13	.48 ± .14	.58 ± .07	.58 ± .09	.61 ± .06	*.78 ± .05*
2/4	.41 ± .06	.72 ± .10	.70 ± .09	.75 ± .07	.68 ± .10	.76 ± .07	.70 ± .07	.71 ± .07	.77 ± .05	*.88 ± .05*
3/4	.50 ± .05	.83 ± .08	.80 ± .07	.84 ± .08	.79 ± .07	.85 ± .05	.81 ± .05	.82 ± .06	.86 ± .05	*.92 ± .04*
Glass5 (4/4)	.58 ± .04	.81 ± .05	.79 ± .05	.86 ± .08	.77 ± .06	.83 ± .06	.70 ± .06	.83 ± .06	.88 ± .07	*.93 ± .04*
1/4 (Low)	0.0 ± 0.0	.26 ± .19	.39 ± .17	.40 ± .18	.21 ± .21	.42 ± .15	.17 ± .16	.47 ± .17	.44 ± .14	*.71 ± .11*
2/4	.26 ± .05	.51 ± .10	.49 ± .11	.72 ± .08	.57 ± .11	.70 ± .09	.49 ± .12	.66 ± .08	.66 ± .07	*.83 ± .06*
3/4	.45 ± .05	.73 ± .05	.71 ± .06	.79 ± .06	.69 ± .07	.78 ± .06	.61 ± .08	.75 ± .06	.79 ± .05	*.90 ± .04*
Yeast5 (4/4)	.42 ± .03	.68 ± .06	.63 ± .05	.73 ± .06	.60 ± .05	.67 ± .06	.62 ± .07	.69 ± .05	.76 ± .06	*.79 ± .06*
1/4 (Low)	.21 ± .08	.48 ± .10	.46 ± .12	.49 ± .06	.43 ± .14	.40 ± .16	.42 ± .15	.46 ± .05	.49 ± .08	*.67 ± .06*
2/4	.27 ± .05	.55 ± .08	.50 ± .09	.55 ± .07	.47 ± .10	.49 ± .11	.51 ± .11	.51 ± .09	.57 ± .07	*.71 ± .07*
3/4	.35 ± .04	.61 ± .08	.57 ± .10	.63 ± .09	.54 ± .08	.58 ± .09	.57 ± .08	.62 ± .08	.66 ± .07	*.74 ± .06*
Yeast6 (4/4)	.16 ± .02	.39 ± .07	.37 ± .06	.41 ± .06	.38 ± .05	.46 ± .07	.43 ± .06	.46 ± .05	.48 ± .05	*.58 ± .06*
1/4 (Low)	.08 ± .12	.22 ± .12	.18 ± .14	.26 ± .10	.23 ± .13	.19 ± .10	.22 ± .09	.29 ± .11	.28 ± .10	*.46 ± .10*
2/4	.11 ± .08	.25 ± .13	.26 ± .11	.31 ± .08	.29 ± .09	.26 ± .12	.27 ± .10	.35 ± .09	.34 ± .08	*.49 ± .08*
3/4	.14 ± .05	.30 ± .10	.32 ± .09	.37 ± .06	.34 ± .08	.32 ± .09	.37 ± .09	.40 ± .08	.41 ± .07	*.54 ± .08*
Abalone(4/4)	0.0 ± 0.0	.10 ± .02	.07 ± .01	.09 ± .02	.09 ± .01	.10 ± .02	.07 ± .04	.11 ± .03	.12 ± .02	*.15 ± .03*
1/4 (Low)	0.0 ± 0.0	.02 ± .12	0.0 ± 0.0	.02 ± .13	0.0 ± 0.0	.02 ± .17	.02 ± .14	.03 ± .08	.03 ± .06	*.05 ± .07*
2/4	0.0 ± 0.0	.05 ± .09	.02 ± .10	.04 ± .09	.03 ± .09	.06 ± .11	.03 ± .11	.05 ± .07	.06 ± .06	*.09 ± .05*
3/4	0.0 ± 0.0	.07 ± .05	.05 ± .04	.07 ± .04	.05 ± .03	.09 ± .08	.04 ± .08	.08 ± .05	.10 ± .04	*.13 ± .05*

majority constraint helps in decreasing the class imbalance. Here, we discuss the significance of this constraint on the performance of [MC]s2s-MLP approach. The proposed approach *without* considering majority–majority constraint ($M_+ = N_+$) is referred to as s2s-MLP, while the earlier [MC]s2s-MLP method considering majority–majority constraint ($M_+ = N_-$) is represented as [MC]s2s-MLP in table 7. Note that ΔF_1 and ΔF_2 in table 7 are computed as

$$\Delta F_1 = F_1([\text{MC}]\text{s2s} - MLP) - F_1(\text{s2s} - MLP),$$
$$\Delta F_2 = F_2([\text{MC}]\text{s2s} - MLP) - F_2(\text{s2s} - MLP),$$

respectively. Table 7 provides the performance obtained for s2s-MLP and [MC]s2s-MLP on 10 datasets. It can be observed that the majority–majority constraint improves the performance of the [MC]s2s-MLP approach by a significant margin (note that ΔF_1 and ΔF_2 are positive across all datasets as shown in table 7). This shows the importance of the proposed majority–majority constraint (which enables reducing the class IR (4)) on [MC]s2s-MLP for addressing the class imbalance problem.

4.5 *Computational complexity*

The quadratic increase in the number of training samples generated by the proposed s2s data representation results in an increase in the number of input and output layer units of the MLP , compared with the traditional MLP (see figure 1). The sheer increase in the size of the architecture would have an impact on the computational time and complexity due the data representation in our s2s approach. We analyse the computational aspects in this section.

As one can expect, the computational time taken to train the s2s-MLP for a single epoch is higher compared with the single epoch time taken to train a traditional MLP or a CSMLP. However, as observed in figure 5 the [MC]s2s-MLP-based system converges much faster and to a better local minima compared with MLP and CSMLP. Figure 5 shows the loss (binary cross-entropy) obtained on the train and test sets, of Yeast6 dataset, at every epoch for MLP-, CSMLP- and [MC]s2s-MLP-based systems. It can be observed from figure 5 that [MC]s2s-MLP requires lower number of epochs (23 epochs) to reach the minima of the training loss (loss = 0.017) compared with MLP

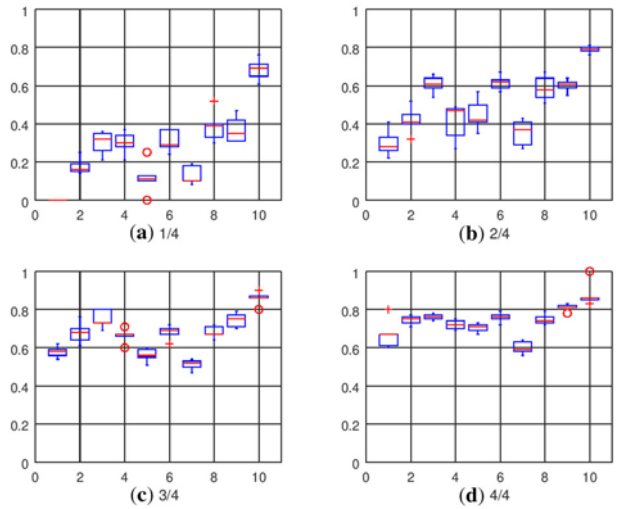

Figure 3. Box plot of F_1 score for the dataset GLASS5: (a) 1/4 (low-resource), (b) 2/4, (c) 3/4 and (d) 4/4 (see table 5). (1) MLP, (2) MWM, (3) CSM, (4) CMLP, (5) RUSB, (6) EUSB, (7) IIvotes (8) GSVM, (9) UCML and (10) [MC]s2s-MLP.

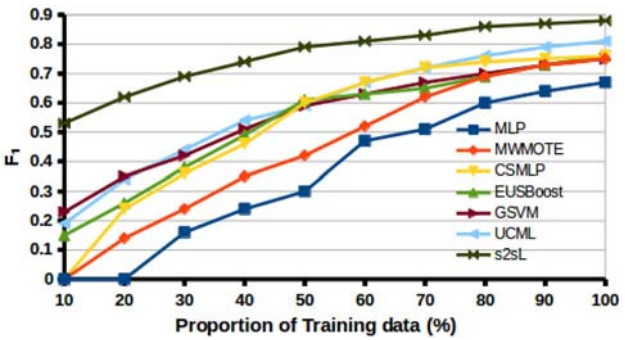

Figure 4. Performance (F_1 values) of [MC]s2s-MLP compared with other methods on GLASS5 dataset by varying the proportion of training data from 10% to 100%.

Table 7. F_1 values for s2s-MLP and [MC]s2s-MLP.

	s2s-MLP		[MC]s2s-MLP			
Dataset	F_1	F_2	F_1	F_2	ΔF_1	ΔF_2
PIMA	0.67	0.68	*0.69*	*0.71*	0.02	0.01
GLASS0	0.66	0.72	*0.68*	*0.75*	0.02	0.03
VEHICLE0	0.93	0.94	*0.96*	*0.98*	0.03	0.04
ECOLI1	0.76	0.82	*0.79*	*0.86*	0.03	0.04
YEAST3	0.67	0.77	*0.72*	*0.83*	0.05	0.06
PAGEBLK	0.87	0.89	*0.95*	*0.96*	0.08	0.07
GLASS5	0.80	0.82	*0.88*	*0.93*	0.08	0.11
YEAST5	0.57	0.65	*0.69*	*0.79*	0.12	0.14
YEAST6	0.35	0.46	*0.44*	*0.58*	0.09	0.12
ABALONE	.06	0.11	*.08*	*0.15*	0.02	0.04

(which required 132 epochs to reach the minima of the training loss = 0.062) and CSMLP (which required 124 epochs to reach the minima of the training loss of 0.056). Further, the average training times (in seconds) for convergence (using i5-3210M 3.1GHz CPU with 4-GB RAM configuration machine) on YEAST6 for different techniques are 98.7 ([MC]s2s-MLP), 38.5 (MLP), 43.7 (CSMLP), 213.8 (CSM), 95.3 (GSVM) and 146.4 (EUSB). While s2s is computationally more expensive than MLP and CSMLP, it is comparable or better compared with other state of the art techniques in terms of training time. Note that, as seen in earlier sections, the performance of [MC]s2s-MLP is consistently better than that of all the state of the art approaches.

5. Summary and conclusions

In this paper we proposed a novel approach of data representation or reorganization, called s2s, to address the class imbalance problem in low-resource scenarios. Our [MC]s2s-MLP approach increases the number of training sample instances quadratically by simultaneously considering two samples to train the classifier (MLP). We showed, through extensive experimentation, that this quadratic increase in the number of training sample instances helps the model to learn better because of (i) larger number of training samples generated and (ii) allowing for the model to learn inter- and intra-class similarities and differences. This s2s data representation is advantageous even in low-resource class imbalance conditions because of its ability to control the class balance (as shown in Appendix I). Also, the increased data size does not affect the computational complexity of the model because of the model's ability to converge faster and to a better local minimum (figure 5). Further, to test the trained model, multiple instances of the same test sample are generated by combining the test sample with a set of preselected reference samples; this is another novel aspect that has been described in the paper. In [MC]s2s-MLP approach, a single base classifier is sufficient to obtain majority-voting-based decision on these test sample instances. Experiments conducted on several benchmark datasets, with different degrees of imbalance,

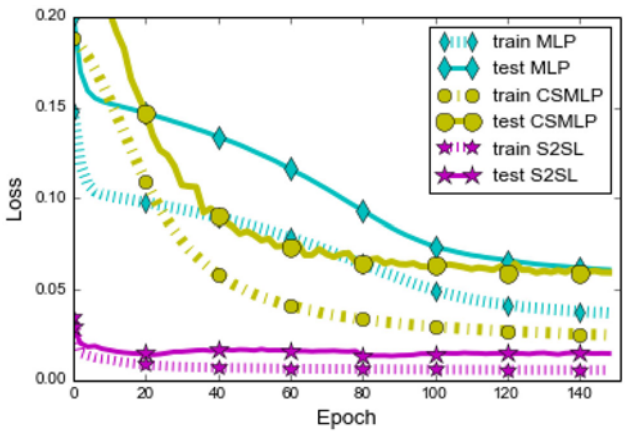

Figure 5. Train and test loss of MLP, CSMLP and s2s MLP.

illustrate the significance of the proposed [MC]s2s-MLP approach over existing state of the art approaches. Experiments demonstrate that the proposed approach works significantly better even in low-resource conditions.

Appendix I.

We know from earlier sections that

$$\begin{aligned}[\text{MC}]\text{s2s}_{N_+} &= (N_+ \times M_+) + (N_+ \times N_-) + (N_- \times N_+) \\ &= (N_+ \times N_-) + (N_+ \times N_-) + (N_- \times N_+) \\ &\quad \text{because} \quad M_+ = N_- \\ &= 3(N_+ \times N_-)\end{aligned}$$

and

$$\begin{aligned}[\text{MC}]\text{s2s}_{N_-} &= (N_- \times N_-) + (N_+ \times N_-) + (N_- \times N_+) \\ &= 2(N_+ \times N_-) + (N_- \times N_-), \\ \frac{[\text{MC}]\text{s2s}_{N_+}}{[\text{MC}]\text{s2s}_{N_-}} &= \frac{3(N_+ \times N_-)}{2(N_+ \times N_-) + (N_- \times N_-)}.\end{aligned}$$

Dividing the numerator and denominator by $(N_+ \times N_-)$ we get

$$\begin{aligned}IR' &= \frac{[\text{MC}]\text{s2s}_{N_+}}{[\text{MC}]\text{s2s}_{N_-}} = \frac{3}{2\frac{(N_+ \times N_-)}{(N_+ \times N_-)} + \frac{(N_- \times N_-)}{(N_+ \times N_-)}} \\ &= \frac{3}{2 + \frac{(N_-)}{(N_+)}} \\ &= \frac{3}{2 + \frac{1}{\left(\frac{N_+}{N_-}\right)}}.\end{aligned} \tag{A1}$$

$$IR' = \frac{[\text{MC}]\text{s2s}_{N_+}}{[\text{MC}]\text{s2s}_{N_-}} = \frac{3\left(\frac{N_+}{N_-}\right)}{\left(1 + 2\left(\frac{N_+}{N_-}\right)\right)}.$$

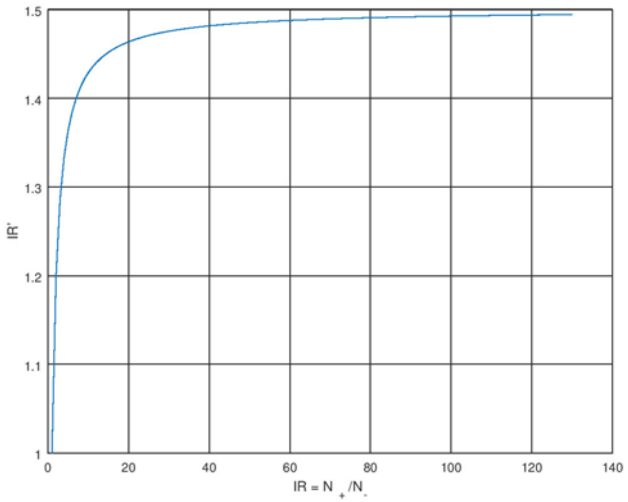

Figure 6. $\frac{[\text{MC}]\text{s2s}_{N_+}}{[\text{MC}]\text{s2s}_{N_-}}$ (y-axis) versus $\left(\frac{N_+}{N_-}\right)$ (x-axis). Plot of (A1).

As seen in figure 6, the [MC]s2s helps in reducing the class imbalance.

Appendix II.

The performance of the proposed [MC]s2s approach is compared to the state of the art, namely (a) data-level approaches (MWMOTE (MWM) [25], CSMOTE (CSM) [28]), (b) algorithm-level approaches (EUSBoost (EUSB) [26]) and (c) hybrid approaches (IIvotes [23], GSVM [24], UCML [27]). The results are provided in tables 8 and 9.

Clearly the proposed method outperforms the state of the art techniques, especially when the imbalance ratio is very high (table 9). Even in case of low imbalance datasets, the performance of the proposed method is as good as or better than that of all other methods. Only in case of GLASS0, IIvotes technique betters the proposed [MC]s2s-MLP on the F_2 score.

Table 8. Comparing [MC]s2s-MLP with other techniques (low imbalance).

Dataset		MWM	CSM	EUSB	IIvotes	GSVM	UCML	[MC]s2s-MLP
PIMA	F_1	.68 ± .03	.65 ± .02	.64 ± .03	.68 ± .03	.67 ± .03	.65 ± .03	*.69 ± .03*
	F_2	*.71 ± .04*	.66 ± .03	.67 ± .03	.69 ± .04	.70 ± .04	.67 ± .03	*.71 ± .04*
GLASS0	F_1	*.68 ± .03*	.64 ± .04	.65 ± .03	*.68 ± .04*	.67 ± .03	.65 ± .03	*.68 ± .03*
	F_2	.74 ± .04	.73 ± .05	.72 ± .05	*.76 ± .04*	.75 ± .06	.69 ± .03	.75 ± .04
VEHICLE0	F_1	.94 ± .03	.93 ± .03	.91 ± .04	.92 ± .03	.93 ± .03	.91 ± .02	*.96 ± .02*
	F_2	.96 ± .03	.96 ± .04	.94 ± .05	.95 ± .05	.95 ± .03	.93 ± .03	*.98 ± .03*
ECOLI1	F_1	.74 ± .03	.75 ± .03	.76 ± .04	.78 ± .04	.78 ± .03	.77 ± .03	*.79 ± .03*
	F_2	.78 ± .05	.81 ± .06	.79 ± .05	.80 ± .05	.85 ± .05	.83 ± .05	*.86 ± .05*
YEAST3	F_1	.69 ± .03	.65 ± .03	.67 ± .04	.70 ± .03	.68 ± .03	.66 ± .03	*.72 ± .02*
	F_2	.81 ± .07	.74 ± .06	.76 ± .05	.77 ± .06	.79 ± .07	.78 ± .06	*.83 ± .05*

Table 9. Comparison of [MC]s2s-MLP with other techniques (high imbalance): F_1 and F_2 with standard deviation.

Dataset		MWM	CSM	EUSB	IIvotes	GSVM	UCML	[MC]s2s-MLP
PAGEBLK	F_1	.88 ± .03	.84 ± .02	.87 ± .02	.88 ± .05	.89 ± .02	.91 ± .03	*.95 ± .03*
	F_2	.88 ± .06	.85 ± .06	.89 ± .06	.89 ± .06	.91 ± .05	.92 ± .04	*.96 ± .04*
GLASS5	F_1	.75 ± .03	.72 ± .03	.76 ± .04	.60 ± .05	.75 ± .03	.81 ± .03	*.88 ± .02*
	F_2	.81 ± .05	.79 ± .05	.83 ± .06	.70 ± .06	.83 ± .06	.88 ± .07	*.93 ± .04*
YEAST5	F_1	.56 ± .03	.55 ± .03	.57 ± .03	.54 ± .05	.60 ± .03	.58 ± .02	*.69 ± .02*
	F_2	.68 ± .06	.63 ± .05	.67 ± .06	.62 ± .07	.69 ± .05	.76 ± .06	*.79 ± .06*
YEAST6	F_1	.25 ± .02	.28 ± .03	.32 ± .03	.34 ± .03	.30 ± .03	.34 ± .04	*.44 ± .03*
	F_2	.39 ± .07	.37 ± .06	.46 ± .07	.43 ± .06	.46 ± .05	.48 ± .05	*.58 ± .06*
ABALONE	F_1	.05 ± .01	.04 ± .01	.06 ± .01	.04 ± .03	.06 ± .01	.07 ± .01	*.08 ± .01*
	F_2	.10 ± .02	.07 ± .01	.10 ± .02	.07 ± .04	.11 ± .03	.12 ± .02	*.15 ± .03*

References

[1] Dubey R, Zhou J, Wang Y, Thompson P M, Ye J and Alzheimer's Disease Neuroimaging Initiative 2014 Analysis of sampling techniques for imbalanced data: an $n = 648$ ADNI study. *NeuroImage* 87: 220–241

[2] Horton P and Nakai K 1996 A probabilistic classification system for predicting the cellular localization sites of proteins. In: *Proceedings of ISMB*, vol. 4, pp. 109–115

[3] Liu Y H and ChenY T 2005 Total margin based adaptive fuzzy support vector machines for multiview face recognition. In: *Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1704–1711

[4] Bermejo P, Gámez J A and Puerta J M 2011 Improving the performance of naive Bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. *Expert Systems with Applications* 38(3): 2072–2080

[5] Kopparapu S K 2014 *Non-linguistic analysis of call center conversations*. In: *Springer Briefs in Electrical and Computer Engineering*. Springer International Publishing

[6] Chakraborty R, Pandharipande M and Kopparapu S K 2017 *Analyzing emotion in spontaneous speech*. Springer

[7] Chawla N V, Japkowicz N and Kotcz A 2004 Editorial. Special issue on learning from imbalanced data sets. *SIGKDD Explorer Newsletter* 6(01): 1–6

[8] Sun Y, Wong A K C and Kamel M S 2009 Classification of imbalanced data: a review. *International Journal of Pattern Recognition and Artificial Intelligence* 23(04): 687–719

[9] Kotsiantis S, Kanellopoulos D, Pintelas P *et al* 2006 Handling imbalanced datasets: a review. *GESTS International Transactions on Computer Science and Engineering* 30(1): 25–36

[10] He H and Garcia E A 2009 Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21(9): 1263–1284

[11] Hu J, Yang H, King I, Lyu M R and So A M C 2015 Kernelized online imbalanced learning with fixed budgets. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15. AAAI Press, pp. 2666–2672

[12] Chawla N V, Bowyer K W, Hall L O and Philip Kegelmeyer W 2002 SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16: 321–357

[13] Liu X Y, Wu J and Zhou Z H 2009 Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics* 39(2): 539–550

[14] Polikar R 2006 Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine* 6(3): 21–45

[15] Seiffert C, Khoshgoftaar T M, Van Hulse J and Napolitano A 2010 RUSBoost: a hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans* 40(1): 185–197

[16] Castro C L and Braga A P 2013 Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems* 24(6): 888–899

[17] Zhou Z H and Liu X Y 2006 Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* 18(1): 63–77

[18] Dumpala S H, Chakraborty R and Kopparapu S K 2018 A novel data representation for effective learning in class imbalanced scenarios. In: *Proceedings of IJCAI*, pp. 2100–2106

[19] Dumpala S H, Chakraborty R and Kopparapu S K 2017 A novel approach for effective learning in low resourced scenarios. In: *Proceedings of the Machine Learning for Audio Signal Processing Workshop*, NIPS

[20] Abd Elrahman S M and Abraham A 2013 A review of class imbalance problem. *Journal of Network and Innovative Computing* 1(2013): 332–340

[21] Ali A, Shamsuddin S M and Ralescu A L 2015 Classification with class imbalance problem: a review. *International Journal of Advances in Soft Computing and its Applications* 7(3): 176–204

[22] Ting K M 2002 An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering* 14(3): 659–665

[23] Błaszczyński J, Deckert M, Stefanowski J and Wilk S 2010 Integrating selective pre-processing of imbalanced data with Ivotes ensemble. In: *Proceedings of the International Conference on Rough Sets and Current Trends in Computing*. Springer, pp. 148–157

[24] Tang Y, Zhang Y Q, Chawla N V and Krasser S 2009 SVMS modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics* 39(1): 281–288

[25] Barua S, Islam M M, Yao X and Murase K 2014 MWMOTE—majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering* 26(2): 405–425

[26] Galar M, Fernández A, Barrenechea E and Herrera F 2013 EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition* 46(12): 3460–3471

[27] Wu F, Jing X Y, Shan S, Zuo W and Yang J Y 2017 Multiset feature learning for highly imbalanced data classification. In: *Proceedings of AAAI*, pp. 1583–1589

[28] Nanni L, Fantozzi C and Lazzarini N 2015 Coupling different methods for overcoming the class imbalance problem. *Neurocomputing* 158: 48–61

[29] Guo Y, Greiner R and Schuurmans D 2005 Learning coordination classifiers. In: *Proceedings of IJCAI*, pp. 714–721

[30] Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L and Herrera F 2011 Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing* 17

[31] Fernández A, García S, del Jesus M J and Herrera F 2008 A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems* 159(18): 2378–2398

[32] Maratea A, Petrosino A and Manzo M 2014 Adjusted *F*-measure and kernel scaling for imbalanced data learning. *Information Sciences* 257: 331–341