Applied Soft Computing 70 (2018) 1146–1153

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An ensemble of decision trees with random vector functional link networks for multi-class classification

Rakesh Katuwal[a], P.N. Suganthan[a,*], Le Zhang[b]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore
[b] Advanced Digital Sciences Center, 1 Fusionopolis Way, 08-10 Connexis North Tower, Singapore 138632, Singapore

A R T I C L E I N F O

Article history:
Received 14 November 2016
Received in revised form 4 September 2017
Accepted 9 September 2017
Available online 18 September 2017

Keywords:
Random forest
Oblique random forest
Neural network
Random vector functional link network (RVFL)
Classification
Ensemble

A B S T R A C T

Ensembles of decision trees and neural networks are popular choices for solving classification and regression problems. In this paper, a new ensemble of classifiers that consists of decision trees and random vector functional link network is proposed for multi-class classification. The random vector functional link network (RVFL) partitions the original training samples into *K* distinct subsets, where *K* is the number of classes in a data set, and a decision tree is induced for each subset. Both univariate and multivariate (oblique) decision trees are used with RVFL. The performance of the proposed method is evaluated on 65 multi-class UCI datasets. The results demonstrate that the classification accuracy of the proposed ensemble method is significantly better than other state-of-the-art classifiers for medium and large sized data sets.

1. Introduction

Ensemble of classifiers [1], also known as multiple classifier systems, is a widely researched and frequently applied approach in machine learning. A multitude of studies corroborate that the combination of many unstable classifiers into one aggregated classifier leads to an improved classification performance compared to a single instance of such unstable classifier [2–5]. The tenets of ensembling technique are perturbation and combination [6] i.e. ensembles are constructed by generating multiple versions of the unstable classifiers by perturbing the training set or injecting some randomness in each classifier and aggregating the outputs of these classifiers in a suitable way.

Decision trees (DT) and neural networks (NN) are two prominent examples of unstable classifiers [6]. The predictive capability of such unstable classifiers greatly vaies even when there is a small perturbation in training set. Hence, ensembles of decision trees and neural networks with various strategies are proposed in [7,5,8,9]. Random Forest (RF) [10] is a paragon of such ensembles. It is a committee of decision trees built using bagging [11] and random subspace [12] methods with eminent performance in classification and regression tasks [13–16]. Random Vector Functional Link (RVFL) network on the other hand, is a randomized variant of Functional Link Neural Network (FLNN) [17]. It derives its architecture from FLNN with enhancement nodes added in between the input and output layers. The weight and bias vectors for the enhancement nodes (congruent to hidden nodes of single-layer feed-forward (SLFN) networks) in RVFL are randomly generated thus, making the learning algorithm less complicated and faster to train than conventional back-prop based SLFN [18,19]. The most appealing trait of RVFL ensemble is its extremely short training time.

In [20], Delgado et al. evaluated 179 classifiers on 121 data sets and posit RF as the best ranked classifier for these data sets. In [21], the authors investigated the performance of a neural network ensemble using RVFL as base classifiers in regression data sets. The authors report impressive performance of the ensemble with significant reduction in training time compared with other ensemble methods like bagging, boosting [22], and RF. The performance of a single RVFL network in time series (TS) forecasting is assessed and compared in [23] with several other methods like RF, support vector regression (SVR), artificial neural network (ANN), seasonal autoregressive integrated moving average (sARIMA). The authors report that the performance of RVFL is second to RF, while being the best among the non-ensemble methods. Similarly, a comprehensive evaluation of RVFL (effect of bias in the output layer, direct links from the input layer to the output layer, scaling of parame-

* Corresponding author.
E-mail addresses: rakeshku001@e.ntu.edu.sg (R. Katuwal), epnsugan@ntu.edu.sg (P.N. Suganthan), zhang.le@adsc.com.sg (L. Zhang).

https://doi.org/10.1016/j.asoc.2017.09.020
1568-4946/© 2017 Elsevier B.V. All rights reserved.

ter randomization, type of activation functions in the hidden layer) on 121 UCI classification data sets is performed in [24]. The performance of RVFL on the classification data sets concords with the conclusions disseminated in [23] with the direct links enhancing the performance of the RVFL. All these works concord that RF and RVFL exhibit a good performance in both classification and regression tasks.

There exists copious researches on decision tree and neural network ensembles [25–29] however, very limited researches have been performed that investigate the performance of the ensembles incorporating both DT and NN. In [30], the authors use decision trees to empirically determine the number of neurons needed in three layers of neural network. [31] extends this idea by mapping stacked RF to convolutional neural networks (CNN) for semantic segmentation. Similarly, in [32], Jerez et al. use decision trees to identify most important variables from breast cancer data set and use those variables as input for their neural network architecture. The work in [33] integrates NN and DT by replacing the final Softmax layer of CNN by DT. In [34], the authors use multi-layer perceptrons as split functions in each node of the trees. However, the works in [30,32] only focus on inducing neural networks via decision trees and are non-ensemble methods, while the rest of the works are only applicable for problems with large training data sets with spatial relationship among the features like computer vision tasks.

This paper investigates the performance of an ensemble of classifiers built with decision trees and neural networks. Since the proposed ensemble is a hybrid ensemble of decision trees and RVFL, we name the new classifiers as RFL and obRFL for the ensemble that consists of RVFL with univariate trees and oblique trees, respectively. In a univariate (axis-parallel or orthogonal) tree, a single feature is used to partition the data at each node while in case of a multivariate (oblique) tree, a combination of features is used. In RFL and obRFL, a decision tree is grown for each subset based on the samples partitioned by RVFL. Each subset consists of training samples with majority of the samples from one class and the rest from other classes. Unlike decision trees that partition the training data into two subsets at each non-leaf node, the RVFL in our proposed method partitions the original data into K distinct subsets at the root node resulting in small tree size built for each problem hereafter. For data sets with two classes, the RVFL in our proposed method partitions the training samples such that the two classes will have the same training set. This results in two identical decision trees, grown with the same training set, in each base classifier. Thus, to avoid duplicating trees (training set) and to obtain disparate trees grown with different training sets in each base classifier, we only consider multi-class classification problems in our work. Our method inherits the intrinsic nature of Random Forest, that means, it can also be implemented in parallel using multi-cores or parallel architectures using CPU or GPU [35–37].

The remainder of the paper is organized as follows. In Section 2, an overview of Random Forest, Oblique Random Forest and Random Vector Functional Link Network is presented followed by our proposed method in Section 3. Section 4 delineates our experimental setup and procedures. In Section 5, the performance of our model is compared with other classifiers. Conclusion is presented in Section 6.

2. Related work

2.1. Random Forest

A Random Forest $h(\mathbf{x}, \Theta)$ is a committee of univariate decision trees grown using an independent and identically distributed (*i.i.d.*) random vector Θ and training set, where $\mathbf{x}$ is an input vector. The random vector Θ is the bootstrapped samples from the training set in case of bagging, or the subset of features (m_{try}) considered at each non-leaf node of the tree in case of random subspace [10]. RF combines these two methods to decrease correlation among the individual trees while increasing the randomization of the individual classifiers for improved generalization. Each tree T_i in the ensemble casts a vote for $\mathbf{x}$. The final classification is based on the majority voting of all the trees T. Thus, RF is simpler to train and tune, and is difficult to overfit [38]. Table 1 depicts the process of creating RF.

Table 1
Random Forest.

```
Algorithm
Training phase:
For $i = 1, \ldots, M$
1. Generate the training set for $T_i$ by drawing $N$ bootstrapped samples
   from the training data, where $N$ is the number of training samples.
2. Grow a tree $T_i$ to the bootstrapped samples repeating the following
   steps at each node of the tree until it is fully grown.
(a) Select $m_{try}$ randomly chosen features from the original feature
    size, and pick the best variable/split-point among them.
(b) Split the node into two nodes.

Classification phase:
To classify a test data, it is pushed down in each tree of the forest. Each
tree votes for a class. The predicted class for that data is the one with
the most votes in the forest.
```

2.2. Oblique Random Forest

An Oblique Random Forest (obRF) is an ensemble of multivariate trees [29]. Several methods like CART-LC [39], OC1 [40], oRF-ridge [29], MPSVM [41] have been used to create oblique splits at each node. However, a clear distinction between these techniques has not been studied assiduously. A drawback of obRF proposed in [29,40] is that these methods are computationally expensive. Since, Multisurface Proximal Support Vector Machine (MPSVM) based oblique decision tree has been extensively evaluated with large number of data sets, it is computationally inexpensive compared with others techniques, and its performance is either better or on par with Random Forest [41], it is used to create oblique trees in our method.

MPSVM was originally proposed in [42] for binary classification using Support Vector Machine (SVM). In [41], the authors use MPSVM with decision trees for both binary and muticlass classification. The notion behind MPSVM technique is to find two clustering hyperplanes H_1 and H_2 for each of the two groups of data D_1 and D_2, such that H_1 (or H_2) is closest to one class of data D_1 (or D_2) and as far as possible from the other class D_2 (or D_1). The clustering hyperplanes are obtained by the eigenvectors corresponding to the smallest eigenvalues of two generalized eigenvalue problems (given by Eqs. (8) and (13) in [42]). The hyperplane used to split the data points in each node is the bisector of H_1 and H_2. In multiclass classification, all classes are divided into two hyper-classes at each node based on the Bhattacharyya distance. For details on oblique trees using MPSVM, readers can refer to [41].

The tree induction process in obRF is similar to that of RF.

2.3. Random Vector Functional Link Network

RVFL is a universal approximator for any continuous function on any compact sets [43]. If $f^*(x)$ is an approximation of $f(x)$ that

Table 2
Proposed method.

```
Algorithm:
Training phase:
For $j = 1, \ldots, M$
1. Generate $T_j$ by drawing $N$ bootstrapped samples from the training data.
2. Construct a RVFL network using $T_j$.
3. For $i = 1, \ldots, N$
(a) For each data $x_i$, select the classes with the highest and the second
    highest scores and store $x_i$ as the training data for the two decision
    trees belonging to those two classes.
4. Grow a decision tree for each partition without pruning.

Testing phase:
In the testing phase, a test data is first passed to RVFL. Similar to the
training phase, two classes with the highest and the second highest
scores are selected based on the output of RVFL. The test data is
pushed down in the decision trees of those classes. The final prediction
is based on the majority votes of the decision trees.
```

maps $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ to a target response $Y = [y_1, y_2, \ldots, y_n]$, a RVFL network can be expressed as:

$$f^*(x) = \sum_j (\beta_j \cdot g(A_j^T \mathbf{x} + b_j)) \tag{1}$$

where A_j is a vector of weights from the input to the jth enhancement node, b_j is the bias and β_j are the weights for the links to the output node. Since the weights and biases are randomly generated, only the output weights β need to be computed. β is computed by solving the following problem:

$$Y_i = d_i^T \beta, \ \ i = 1, 2, \ldots, N \tag{2}$$

where N is the total number of data samples and d is a vector formed by concatenating the original and random features. Solving Eq. (2) directly could result in over-fitting. Two methods could be employed to avoid overfitting: Moore-Penrose Pseudoinverse and L2 norm regularized least square (ridge regression). Since the ridge regression based closed form solution is shown to perform better in [24], we employ ridge regression in our method. The ridge regression solves the following problem:

$$\sum_i (Y_i - d_i^T \beta)^2 + \lambda \|\beta\|^2 \tag{3}$$

The solution of Eq. (3) is $\beta = D(D^T D + \lambda I)^{-1} Y$ where, D and Y are the features and targets matrices, and λ is the regularization parameter.

3. Proposed ensemble method

First we describe the construction of the hybrid ensemble of classifiers, then we discuss the motivation behind the proposed method.

In the training phase, each training sample x_i is passed to RVFL. The output of RVFL is a score for each class in that particular data set. The class with the highest score is the predicted class by RVFL. Since RVFL can misclassify the training data, two classes with the highest and second-highest score are selected as its potential classes in our method. Thus, x_i is used as a training data for the decision trees associated with those two classes/subsets. This procedure is repeated for all training samples, creating a training set for each decision tree. The final model is an ensemble of such base classifiers. In cases, where the true class of x_i is neither the highest nor the second-highest class, the training sample is still placed in its true class also. The algorithm of our proposed method is presented in Table 2 and its visual representation in Fig. 1. A preliminary experiment with the proposed ensemble method with classifier employing kernel method is presented in [44].

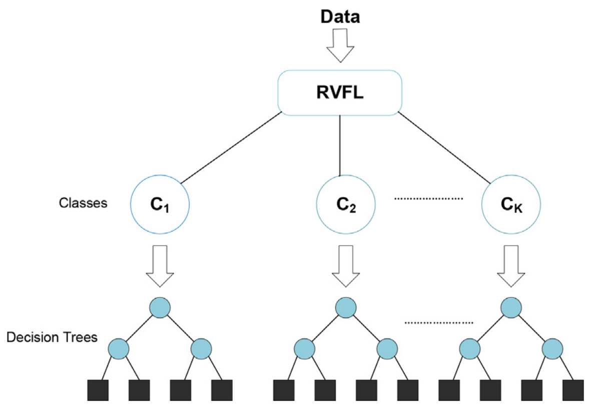


Fig. 1. A base classifier of our proposed ensemble method.

In this paper, unlike the standard decision trees, we apply an efficient neural network (RVFL) to firstly divide the data into 'K' partitions (K: the number of classes in a dataset) where we grow a decision tree thereafter to improve the accuracy. In each subset/data partition, the class distribution of samples is unique i.e. majority of the samples are from one class and the rest from other classes. The samples from the other classes are those that are "hard" to classify by RVFL. Such partitioning is possible by utilizing the output scores given by RVFL. Decision trees are used thereafter because:

- They naturally handle multi-classes. No need to convert them to a series of binary classification problems (like one-vs-one or one-vs-all) as in other popular classifiers such as SVM. Since the samples from other classes are sparse, converting them to such binary problems will deteriorate the performance of the other classifiers however, DT/RF are robust against imbalance problem if such exists [45–47].
- Instead of growing very large trees as done in RF, the size of the trees in our proposed ensemble method is relatively small due to the K-partitions performed by RVFL at the root node. Thus, the tree construction process can be distributed among several workstations or cores without exhausting the resources. This property makes our method ideal for parallelization.
- Normally, binary splits are popular with decision trees. The binary splits result in very deep trees. However, by using multi-way splits as proposed in this paper, we can obtain shallow trees that agrees with Ockham's razor principle of building smaller trees. There are some papers in the literature [48,49] that propose decision trees with multi-way splits, but their technique of splits generation is very complex (based on clustering algorithms, significance tests) and the accuracy of such trees is also generally lower than decision trees with binary splits. But, we show that using multi-way splits at the root node as proposed in this paper can further improve the performance of trees.

4. Experimental setup

The proposed method is tested in 65 multi-class UCI data sets [50]. The number of classes in those data sets ranges from 3 (minimum) to 100 (maximum). We follow the procedures as in [20]. A 4-fold cross validation is developed using the whole data. For those data sets, where the training-testing partition is already available (such as annealing and audiology-std), the classifier is trained on the predefined training set and evaluated on the test set. For such data sets, the experiment is repeated 4 times. The ensemble size M is fixed at 500 [20]. Table 10 in Appendix gives an overview of the data sets used in this paper.

Table 3
Classification accuracies of different methods.

Datasets	RF	RFL	obRF	obRFL	RVFL
Abalone	64.99	64.75	65.37	65.16	**65.73**
Annealing	56	**76.25**	76	76	24
Arrhythmia	**73.45**	70.58	65.04	65.71	68.14
Audiology-std	**84**	69	68	60	64
Balance-scale	88.3	87.5	90.22	**90.87**	**90.87**
Breast-tissue	75	73.08	72.12	**75.96**	64.42
Car	97.16	**97.45**	95.95	97.16	93
Contrac	**54.28**	52.51	51.22	50.68	54.08
Ctg-10classes	86.72	**87.15**	82.77	84.23	79.85
Ctg-3classes	94.59	**94.96**	92.7	92.8	90.77
Chess-krvk	70.48	**75.6**	68.19	70.25	37.07
Conn-bench-vowel	98.48	99.46	99.78	**100**	98.27
Dermatology	**97.8**	**97.8**	96.98	96.7	97.53
Ecoli	88.69	**88.99**	87.5	86.9	87.8
Energy-y1	**95.18**	95.05	92.32	91.93	93.62
Energy-y2	89.71	89.45	89.19	89.58	**90.36**
Flags	**66.15**	64.06	56.25	55.21	52.6
Glass	**74.53**	73.58	73.11	71.23	68.4
Hayes-roth	85.71	**89.29**	85.71	**89.29**	85.71
Heart-cleveland	56.91	57.57	59.54	58.88	**62.5**
Heart-va	**38**	35.5	36	37.5	**38**
Image-segmentation	14.29	**29.54**	14.9	29.32	26.95
Iris	95.95	95.95	97.3	97.3	**98.65**
Led-display	73.7	72.3	72.8	71.6	**75**
Lenses	**87.5**	**87.5**	**87.5**	83.33	83.33
Letter	96	96.36	**96.53**	96.28	79.6
Libras	78.06	83.89	**86.11**	84.44	82.22
Low-res-spect	**92.11**	91.92	91.54	91.73	91.35
Lung cancer	53.13	50	43.75	46.88	**62.5**
Lymphography	84.46	86.49	**87.84**	86.49	84.46
Molec-biol-splice	**95.39**	95.01	89.74	90.37	81.68
Nursery	99.34	**99.43**	98.81	98.96	92.85
OocMerl2F	91.67	92.16	91.96	**92.35**	85.69
OocTris5B	91.78	92	93.42	93.75	**95.06**
Optical	**97.16**	96.7	96.72	97.09	95.6
Page-blocks	97.22	**97.33**	97.04	97.08	95.56
Pendigits	95.31	95.8	96.94	**97.13**	95.63
Pb-MATERIAL	91.35	90.38	**93.27**	92.31	92.31
Pb-REL-L	74.04	73.08	**79.81**	76.92	77.88
Pb-SPAN	65.22	64.13	66.3	66.3	**71.73**
Pb-TYPE	70.19	69.23	69.23	**71.15**	68.27
Plant-margin	**85.5**	83.56	82.82	82.06	77.19
Plant-shape	64.06	67.38	**70.25**	70.19	58.38
Plant-texture	82.19	**82.5**	82.06	81.75	80.75
Post-operative	**72.73**	70.45	70.45	69.32	**72.73**
Primary-tumor	**56.1**	54.57	54.27	53.35	**56.1**
Seeds	94.23	94.23	93.75	94.23	**95.67**
Semeion	**93.97**	93.03	92.78	92.84	89
Soybean	38.56	39.96	63.83	**70.48**	48.4
St-image	97.66	**97.79**	97.53	**97.79**	94.19
St-landstat	90.55	90.66	90.5	**91.1**	85.65
St-shuttle	**99.97**	**99.97**	99.88	99.89	97.26
St-vehicle	74.17	77.37	77.25	77.61	**82.94**
Steel plates	77.73	**79.02**	77.37	78.2	75
Synthetic-control	98.17	98.67	99.33	**99.5**	97.17
Teaching	**62.5**	60.53	**62.5**	61.84	55.26
Thyroid	**98.95**	**98.95**	95.6	95.71	93.93
Vc-3classes	85.06	84.09	85.39	**86.04**	83.77
Waveform	84.7	84.76	86.2	86.06	**86.8**
Waveform-noise	86.56	86.44	86.24	**86.82**	86.24
Wine	98.86	98.86	99.43	99.43	**100**
W-qua-red	68.31	**69**	68.06	68.63	59.63
W-qua-white	68.38	68.44	69.24	**69.77**	55.25
Yeast	62.67	**63.48**	63.14	62.94	60.18
Zoo	**99**	98	**99**	98	97

Bold values indicate the highest accuracy.

Table 4
Average rank values based on classification accuracies of each method. Lower rank reflects better performance.

	RF	RFL	obRF	obRFL	RVFL
Rank	2.79	2.66	3.12	2.79	3.62

4.1. *RVFL configuration*

The necessary and sufficient conditions for building an ensemble of classifiers are the diversity of the base classifiers and the perturbation of training data [3]. To meet these conditions, we employ bagging and randomize RVFL in each base classifier. We use the same parameter setting as in [24,20] where each RVFL randomly picks the activation function and network parameters from the parameter settings listed below:

1. Number of hidden neurons, $\mathcal{N} = 3{:}203$
2. $\lambda(=(1/2)^C)$ in ridge regression, $C = -5{:}14$
3. Activation functions: *radbas*, *sine* and *tribas*
4. Range of the randomization for weights $[-S,+S]$ and bias $[0,S]$, where $S = 2^t$ with $t = -1.5{:}0.5{:}1.5$

The RVFL has direct links from input layer to output layer with bias term in the output neuron.

4.2. *Decision tree configuration*

For both univariate and oblique decision trees, we set the m_{try} parameter (number of features selected at each node during the growing phase of the tree) equal to $\sqrt{n}$ where, n is the total number of features of the data. Similarly, the *minleaf* parameter, which specifies the minimum number of observations per tree leaf, is set to 1 as default [41].

5. Results and discussion

To validate the effectiveness of RFL and obRFL, we compare the classification accuracies obtained by our method with RF, obRF and RVFL. The average accuracies over the data sets by each method are shown in Table 3. For fair comparison, RF and obRF are also built with 500 trees as done in [20]. Our results slightly differ from [20] for RF and [41] for obRF as we use the same configuration for all classifiers i.e. the classifiers are trained with same 500 bags and they all use $\sqrt{n}$ features at each non-leaf node. The RVFL results are tuned for *radbas* activation function and extracted from [24]. The statistical test is based on Friedman test (refer to Appendix for details) as suggested in [51], and we use the rank of each classifier to reflect its performance as in [20]. In this approach, the classifiers are ranked for each data set separately, with the best performing algorithm getting the rank of 1, the second best rank 2. . ., average ranks are assigned in case of ties. Based on the 65 rankings, we summarize the overall ranking of each method in Table 4.

We rank the performances of each method in each data set and take average across all the data sets. By simple calculations we obtain, $\chi^2_F = 12.54$ and $F_F = 3.24$.

With 5 algorithms and 65 data sets, F_F is distributed according to the F-distribution with $5-1=4$ and $(5-1)(65-1)=256$ degrees of freedom. The critical value for $F_{(4,256)}$ for $\alpha = 0.05$ is 2.4, so we reject the null-hypothesis. Based on the Nemenyi test, the critical difference here is $\mathrm{CD} = q_\alpha\sqrt{(k(k+1))/(6N)} = 2.728 * \sqrt{5*6/(6*65)} \simeq 0.75$. Based on the average rank values of each method and CD obtained from Nemenyi test, we can conclude that RF, RFL, and obRFL are significantly better than RVFL. Even though obRF is better than RVFL, the statistical test fails to find any significant difference between them.

5.1. *Comparison of RF, obRF, RFL and obRFL*

Based on the results so far, although it is difficult to find any significant difference, RFL works slightly better than the other methods. RFL is better than RF in 30 of 65 data sets, worse than

Table 5
Average rank values of 4 methods in data sets with $N > 500$. Lower rank signifies better performance.

	RF	RFL	obRF	obRFL
Rank	2.55	2.09	3.01	2.33

RF in 28 data sets, and equal in 7 sets. When calculating for each data set the difference between the accuracies of RFL and RF, the sum of positive differences (RFL is better) is 68.73, while the negative ones (RF is better) sum 48.82. Similarly, obRFL is better than obRF in 36 data sets, worse than obRF in 25 data sets, and equal in 4 sets. Calculating the difference between the accuracies of oRFL and obRF, the sum of positive differences (obRFL is better) is 49.3, while the negative ones (obRF is better) sum 31.27.

Since RF performs slightly better than obRF in multi-class data sets, similar pattern is observed in RFL and obRFL. The average accuracies of RFL and obRFL over all the data sets are 80.63% and 80.62% respectively. RFL has the highest average accuracy among all the 4 methods followed by obRFL.

From the results obtained in Table 3, it is worth noticing that RFL and obRFL generally perform worse in data sets with sample size $N < 500$. Out of 65 data sets, there are 28 such data sets. RF performs better than RFL in 16 data sets out of 28, worse in 7 data sets, and equal in 5. Similarly, obRF is better than obRFL in 15 data sets, worse in 10, and equal in 3. 57.14% (16 out of 28) of RF's overall wins over RFL emanate from such data sets while for obRF, it is 60%. On further investigating the small size data sets, we notice that RFL is better than or comparable to RF only in balanced (number of samples from all the classes are equal) or slightly unbalanced (disparity between the number of samples from different classes) data sets. On highly unbalanced data sets like Arrhythima (Minority class % = 0.44), Glass (Minority class % = 4.2), etc., RF generally outperforms RFL. However, on some unbalanced data sets like Ecoli (Minority class % = 0.59) and Lymphography (Minority class % = 1), RFL still wins over RF since there are very few samples from the minority classes in the test data. This analysis is congruent with the performance of obRF and obRFL in those data sets. For unbalanced data sets with $N > 500$, we observe no such effect. Considering only those 37 data sets with $N > 500$ (both balanced and unbalanced), we present the overall ranking of each method in Table 5.

On repeating the statistical test for new rank values, we obtain $\chi_F^2 = 7.98$ and $F_F = 2.79$. With 4 algorithms and 37 data sets, F_F is distributed according to the F-distribution with $4-1=3$ and $(4-1)(37-1)=108$ degrees of freedom. The critical value for $F_{(3,108)}$ for $\alpha = 0.05$ is 2.68, so we reject the null-hypothesis. Based on the Nemenyi test, the critical difference here is CD $= q_\alpha\sqrt{(k(k+1))/(6N)} = 2.569 * \sqrt{4*5/(6*37)} \simeq 0.77$. Since 2.09 and 3.01 are the only pair of average rank whose difference is larger than 0.77, we can conclude that the RFL is significantly better than obRF for data sets with $N > 500$. Although the difference is not significant, obRFL also works slightly better than RF. The average accuracies of RF, obRF, RFL and obRFL over these data sets are 85.38%, 85.6%, 86.22%, and 85.83% respectively.

Furthermore, we use sign test to compare each pair of algorithms. Out of N data sets, if the number of wins is at least $N/2 + 1.96\sqrt{(N)}/2$, the algorithm is significantly better with $p < 0.05$. The tied matches are split evenly between the two classifiers; if there is an odd number of them, we ignore one. The results are summarized in Table 6.

From Table 6, we can observe that, under the sign test, RFL is significantly better than RF and obRF. obRFL is also statistically better than obRF. However, it is important to note that these observations are based on the data sets with $N > 500$. Our proposed method (RFL and obRFL) performs better than RF and obRF for those data sets (both balanced and unbalanced) with $N > 500$. RFL and obRFL also works better than or comparable to RF and obRF with small ($N < 500$) and balanced or slightly unbalanced data sets. The performance of our proposed method is poor when the data sets are small and highly unbalanced. A reason for this is, for small and highly unbalanced data sets, the accuracy of RVFL is generally low. RVFL in our proposed method determines the corresponding decision trees through which a test data needs to be pushed down. A fairly accurate RVFL predicts the first and the second classes with more confidence. Hence, the data is likely to be correctly classified by the two decision trees associated with those two classes. For $N < 500$ and highly unbalanced data sets, RVFL usually gives mediocre performance and the test data is pushed down in different decision trees rather than the ones belonging to its true class. Hence, such test data is likely to be misclassified. To corroborate this statement, we present the testing accuracy of RVFL in 6 highly unbalanced data sets (3 data sets with $N > 500$ and the other 3 with $N < 500$) in Table 7. The accuracy here is the average accuracy of an ensemble of RVFL (500 RVFLs) calculated based on the prediction made by RVFL of a sample's true class belonging to either the highest or the second highest class. That means, a correct decision is made by RVFL if the test sample's predicted class (either the one with the highest or second-highest scores) matches its true class. This is represented by RVFL (Top 2) in the table. Similarly, we also present the accuracy of RVFL (here, the test sample's predicted class is the one with the highest score) and RF in these data sets for comparison purposes. For detailed results, please refer to Table 3 of the manuscript.

Table 6
Significance of the difference between each pairs of algorithms.

Method	Significance
(RF, RFL)(13,24)	Yes
(RF, obRF)(23,14)	No
(RF, obRFL)(17,19)	No
(RFL, obRF)(25,12)	Yes
(RFL, obRFL)(21,15)	No
(obRF, obRFL)(10,26)	Yes

The numbers in the bracket represent the number of times each method wins.

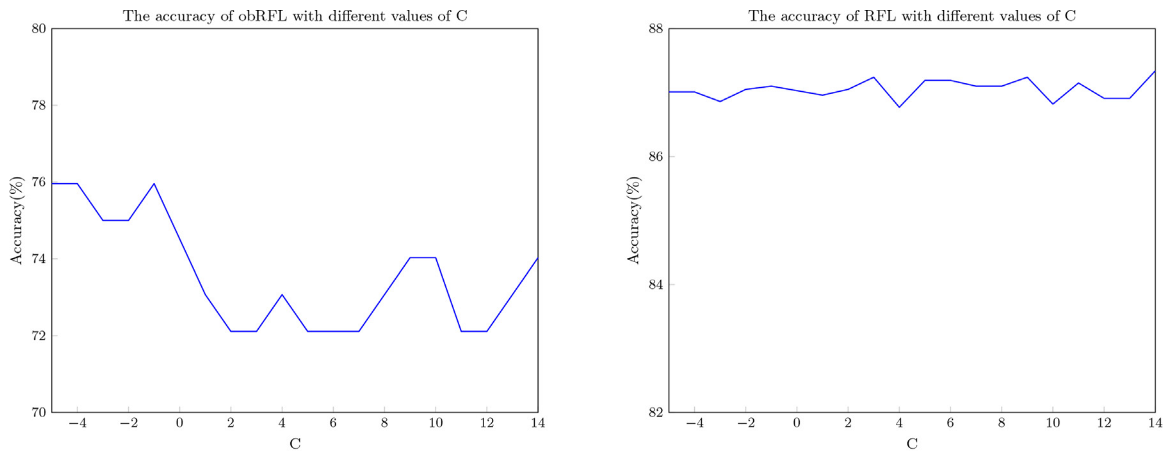
Table 7
Average testing accuracy of different methods.

	Datasets	RVFL	RVFL (Top 2)	RF
($N < 500$)	Arrhythmia	68.14	68.73	73.45
	Flags	52.6	52.8	66.15
	Glass	68.4	77.2	74.53
($N > 500$)	Car	93	97.17	97.16
	Ctg-10classes	79.85	86.74	86.72
	Nursery	92.85	97.57	99.34

From Table 7, we can see that the performance of RF is better than RVFL and comparable to or also better than RVFL (Top 2) in almost all the data sets. Thus, by using DT after RVFL we are trying to improve the accuracy further. However, RVFL gives mediocre performance for small ($N < 500$) and unbalanced data sets even when selecting Top 2 classes. Because of the RVFL's inferior performance in such small and highly unbalanced data sets, the overall accuracy of the proposed ensemble method in such data sets is also poor. However, for larger data sets, the testing accuracy of RVFL is fair and hence, the performance of the ensemble method is also significantly better than RF and obRF.

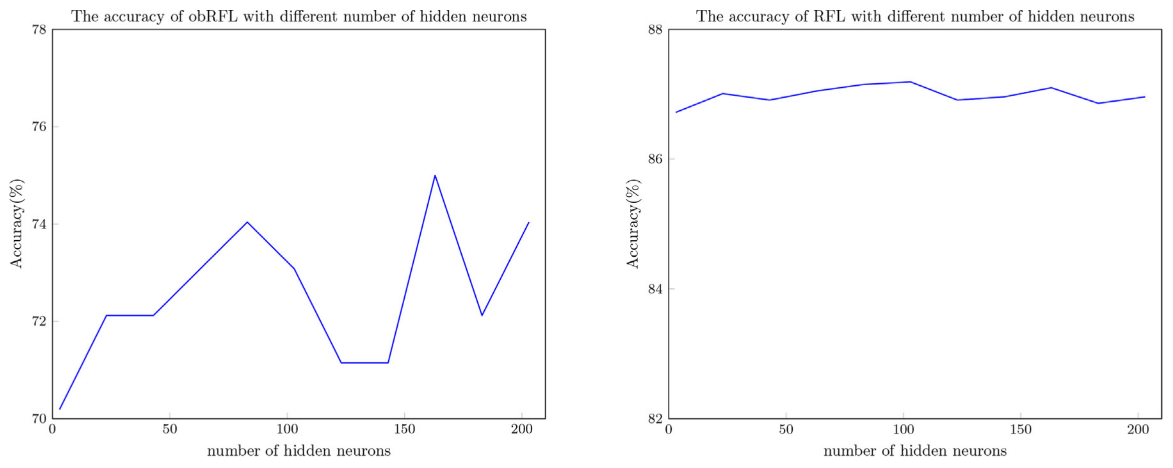
5.2. *On the effect of randomizing lambda and the number of hidden neurons*

λ and the number of hidden neurons ($\mathcal{N}$) are two important parameters of RVFL. Generally, these parameters are tuned for each data set. However, in our method, we choose the values of



(a) obRFL-Breast tissue (b) RFL-Ctg 10classes

Fig. 2. Accuracy of RFL and obRFL with different values of *C*.



(a) obRFL-Breast tissue (b) RFL-Ctg 10classes

Fig. 3. Accuracy of RFL and obRFL with different number of hidden neurons.

Table 8
Average number of nodes in a base classifier of each ensemble.

Datasets	RF	RFL	obRF	obRFL
Abalone	533.01	373.13	713.8	500.38
Annealing	44.54	20.818	78.51	35.28
Arrhythmia	52.94	7.8	92.33	12.56
Audiology-std	28.52	2.96	40.52	3.96
Balance-scale	60.02	44.85	60.21	44.79
Breast-tissue	19.18	4.89	19.18	6.32
Car	81.8	47.68	84.13	50.51
Contrac	229.35	166.44	148.55	120.57
Ctg-10classes	190.28	41.46	329.42	68.14
Ctg-3classes	148.56	61.69	148.56	102.95
Chess-krvk	1942.68	346.95	2801.54	435.04
Conn-bench-vowel	78.72	16.21	114.63	22.39
Dermatology	21.49	9.23	27.62	11.52
Ecoli	29.67	8.16	42.33	10.93

Table 9
Average training time (in seconds) for each ensemble method.

Datasets	RF	RFL	obRF	obRFL
Abalone	1472.49	2370	382.51	820.45
Annealing	114.52	263.02	62.69	101.23
Arrhythmia	433.11	530.02	96.48	263.51
Audiology-std	32.33	41.2	45.71	51.34
Balance-scale	37.2	75.59	43.4	105.37
Breast-tissue	12.78	34.4	15.06	34.11
Car	68.03	120.52	59.09	140.05
Contrac	212.73	351.94	86.47	319.46
Ctg-10classes	630.36	927.31	221.63	816.66
Ctg-3classes	95.91	640.76	86.47	346.21
Chess-krvk	2290.29	5559.42	2936.18	5940
Conn-bench-vowel	273.09	310.68	86.92	314.06
Dermatology	35.91	64.61	19.15	60.11
Ecoli	52.57	71.89	27.71	50.52

λ (or C) and $\mathcal{N}$ randomly from suitable ranges. Although this may result in non-optimal RVFL models, it may offer some advantages in generating diverse base classifiers fast. This may lead to a better performance by reducing variance of the overall system. The objective of our proposed method is to obtain diverse RVFL models in each base classifier which in turn results in diverse data partitions hence, diverse decision trees, and thus varying these parameters in each base classifier fits our objective.

To demonstrate the effect of λ (or C) and $\mathcal{N}$ in the accuracy of the ensemble method, we plot a graph of accuracy vs C and accuracy vs $\mathcal{N}$ in two different data sets as shown in Figs. 2 and 3 respectively. From the plots, we can see that a careful selection of these param-

Table 10
Details of the data sets used in this report.

Datasets	Patterns	Features	Classes
Abalone	4177	8	3
Annealing	798	38	6
Arrhythmia	452	262	13
Audiology-std	226	59	18
Balance-scale	625	4	3
Breast-tissue	106	9	6
Car	1728	6	4
Ctg-10classes	2126	21	10
Ctg-3classes	2126	21	3
Contrac	1473	9	3
Chess-krvk	28056	6	18
Conn-bench-vowel	528	11	11
Dermatology	366	34	6
Ecoli	336	7	8
Energy-y1	768	8	3
Energy-y2	768	8	3
Flags	194	28	8
Glass	214	9	6
Hayes-roth	132	3	3
Heart-cleveland	303	13	5
Heart-va	200	12	5
Image-segmentation	210	19	7
Iris	150	4	3
Led-display	1000	7	10
Lenses	24	4	3
Letter	20000	16	26
Libras	360	90	15
Low-res-spect	531	100	9
Lung cancer	32	56	3
Lymphography	148	18	4
Molec-biol-splice	3190	60	3
Nursery	12960	8	5
OocMerl2F	1022	25	3
OocTris5B	912	32	3
Optical	3823	62	10
Page-blocks	5473	10	5
Pendigits	7494	16	10
Pb-MATERIAL	106	4	3
Pb-REL-L	103	4	3
Pb-SPAN	92	4	3
Pb-TYPE	105	4	6
Plant-margin	1600	64	100
Plant-shape	1600	64	100
Plant-texture	1600	64	100
Post-operative	90	8	3
Primary-tumor	330	17	15
Seeds	210	7	3
Semeion	1593	256	10
Soybean	307	35	18
St-image	2310	18	7
St-landstat	4435	36	6
St-shuttle	43500	9	7
St-vehicle	846	18	4
Steel plates	1941	27	7
Synthetic-control	600	60	6
Teaching	151	5	3
Thyroid	3772	21	3
Vc-3classes	310	6	3
Waveform	5000	21	3
Waveform-noise	5000	40	3
Wine	179	13	3
W-qua-red	1599	11	6
W-qua-white	4898	11	7
Yeast	1484	8	10
Zoo	101	16	7

eters is required to obtain reasonable performance or performance close to the ones obtained when randomizing them for different RVFL models. However, as stated earlier, ensemble methods require diversity and such diversity can easily be obtained by randomizing the values of λ and $\mathcal{N}$ which results in diverse RVFL models. On the other hand, tuning RVFL for best values would increase the computational time and complexity without any significant improvement in the performance.

5.3. Computational complexity

The objective here is to investigate the complexity introduced by the hybridization instead of comparing the computational complexities associated with RF and obRF. In Table 8, we summarize the average number of nodes in a decision tree in each case. First 14 data sets (see Table 3) are selected for this purpose. However, similar results pertain to other data sets too. Intuitively, we can state that the decision trees in RFL and obRFL are relatively smaller compared to those in RF and obRF because the trees in RFL and obRFL are grown with reduced data set. One may argue that the smaller decision trees are unstable and hence, may lead to increased bias. However, the nature of construction of trees (multiple branches at the root node) in our proposed method is able to alleviate this problem. In Table 9, we report the training time for each ensemble. However, the code for the proposed ensemble method is not optimized for speed. From the table, we can infer that our proposed method gives an improved result compared to their standard counterparts in expense of additional computational time. The trees here are grown sequentially, however with the use of multi-core and parallel/distributed architectures in CPU and GPU, the training time would decrease significantly.

6. Conclusion

We proposed a new ensemble of classifiers based on decision trees and random vector functional link networks. Since our method evolves from RF and RVFL, we named our proposed method as RFL (RVFL with univariate trees) and obRFL (RVFL with oblique trees). In each base classifier of RFL and obRFL, decision trees were induced based on the samples partitioned by RVFL. We evaluated the performance of our proposed methods on 65 multi-class data sets from UCI repository. As reported in Section 5, RFL and obRFL perform better than other methods (RF, obRF, RVFL) for data sets with relatively high number of samples ($N > 500$). Overall, RFL is better than the rest of the methods followed by obRFL. RFL and obRFL also achieve higher average accuracies than RF, obRF and RVFL.

Appendix A.

Friedman test.

Let r_i^j be the rank of the jth of the k algorithms on the ith of the N data sets. The Friedman test compares the average ranks of the algorithms, $R_j = \sum_i r_i^j$. The null hypothesis states that all the algorithms are equivalent and so their ranks R_j should be equal. When N and k are large enough, the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right],$$

is distributed according to χ_F^2 with k-1 degrees of freedom under the null hypothesis. In this case, χ_F^2 is undesirably conservative. A better statistics is

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2},$$

which is distributed according to F-distribution with k-1 and (k-1)(N-1) degrees of freedom. If the null-hypothesis is rejected (performance of the classifiers are statistically significant), the Nemenyi post-hoc test [52] can be used to check whether the performance of two among k classifiers is significantly different. The performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}},$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$. α is the significance level and is equal to 0.05 in this paper.

References

[1] Y. Ren, L. Zhang, P. Suganthan, Ensemble classification and regression – recent developments applications and future directions, IEEE Comput. Intell. Mag. 11 (1) (2016) 41–53 (review article).
[2] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1) (2010) 1–39, http://dx.doi.org/10.1007/s10462-009-9124-7.
[3] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15.
[4] G. Valentini, F. Masulli, Ensembles of learning machines, in: Italian Workshop on Neural Nets, Springer, 2002, pp. 3–20.
[5] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, J. Artif. Intell. Res. 11 (1999) 169–198.
[6] L. Breiman, Bias, variance, and arcing classifiers, Tech. Rep. 460, Statistics Department, University of California, Berkeley, CA, USA.
[7] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging boosting and randomization, Mach. Learn. 40 (2) (2000) 139–157.
[8] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (1) (2002) 239–263.
[9] D.W. Opitz, J.W. Shavlik, Generating accurate and diverse members of a neural-network ensemble, in: Advances in Neural Information Processing Systems, 1996, pp. 535–541.
[10] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[11] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[12] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
[13] A. Liaw, M. Wiener, Classification and regression by randomforest, R News 2 (3) (2002) 18–22.
[14] R. Díaz-Uriarte, S.A. De Andres, Gene selection and classification of microarray data using random forest, BMC Bioinform. 7 (1) (2006) 1.
[15] P. Jiang, H. Wu, W. Wang, W. Ma, X. Sun, Z. Lu, MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features, Nucleic Acids Res. 35 (Suppl. 2) (2007) W339–W344.
[16] V. Svetnik, A. Liaw, C. Tong, J.C. Culberson, R.P. Sheridan, B.P. Feuston, Random forest: a classification and regression tool for compound classification and QSAR modeling, J. Chem. Inf. Comput. Sci. 43 (6) (2003) 1947–1958.
[17] S. Dehuri, S.-B. Cho, A comprehensive survey on functional link neural networks and an adaptive PSO-BP learning for CFLNN, Neural Comput. Appl. 19 (2) (2010) 187–205.
[18] Y.-H. Pao, S.M. Phillips, D.J. Sobajic, Neural-net computing and the intelligent control of systems, Int. J. Control 56 (2) (1992) 263–289.
[19] L. Zhang, P.N. Suganthan, A survey of randomized algorithms for training neural networks, Inf. Sci. 364 (2016) 146–155.
[20] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems, J. Mach. Learn. Res. 15 (1) (2014) 3133–3181.
[21] M. Alhamdoosh, D. Wang, Fast decorrelated neural network ensembles with random weights, Inf. Sci. 264 (2014) 104–117.
[22] Y. Freund, Boosting a weak learning algorithm by majority, in: COLT, vol. 90, 1990, pp. 202–216.
[23] Y. Ren, P.N. Suganthan, N. Srikanth, G. Amaratunga, Random vector functional link network for short-term electricity load demand forecasting, Inf. Sci. 367 (2016) 1078–1093.
[24] L. Zhang, P.N. Suganthan, A comprehensive evaluation of random vector functional link networks, Inf. Sci. 367 (2016) 1094–1105.
[25] L. Zhang, P.N. Suganthan, Visual tracking with convolutional random vector functional link network, IEEE Trans. Cybern. PP (99) (2016) 1–11, http://dx.doi.org/10.1109/TCYB.2016.2588526.
[26] L. Zhang, Y. Ren, P.N. Suganthan, Instance based random forest with rotated feature space, in: 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), IEEE, 2013, pp. 31–35.
[27] L. Zhang, Y. Ren, P.N. Suganthan, Towards generating random forests via extremely randomized trees, in: 2014 International Joint Conference on Neural Networks (IJCNN), IEEE, 2014, pp. 2645–2652.
[28] L. Zhang, P.N. Suganthan, Random forests with ensemble of feature spaces, Pattern Recognit. 47 (10) (2014) 3429–3437.
[29] B.H. Menze, B.M. Kelm, D.N. Splitthoff, U. Koethe, F.A. Hamprecht, On oblique random forests, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 453–469.
[30] I.K. Sethi, Entropy nets: from decision trees to neural networks, Proc. IEEE 78 (10) (1990) 1605–1613.
[31] D.L. Richmond, D. Kainmueller, M.Y. Yang, E.W. Myers, C. Rother, Relating cascaded random forests to deep convolutional neural networks for semantic segmentation, arXiv:1507.07583.
[32] J.M. Jerez-Aragonés, J.A. Gómez-Ruiz, G. Ramos-Jiménez, J. Mu noz-Pérez, E. Alba-Conejo, A combined neural network and decision trees model for prognosis of breast cancer relapse, Artif. Intell. Med. 27 (1) (2003) 45–63.
[33] P. Kontschieder, M. Fiterau, A. Criminisi, S. Rota Bulo, Deep neural decision forests, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1467–1475.
[34] S. Rota Bulo, P. Kontschieder, Neural decision forests for semantic image labelling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 81–88.
[35] L. Mitchell, T.M. Sloan, M. Mewissen, P. Ghazal, T. Forster, M. Piotrowski, A.S. Trew, A parallel random forest classifier for R, in: Proceedings of the Second International Workshop on Emerging Computational Methods for the Life Sciences, ACM, 2011, pp. 1–6.
[36] T. Sharp, Implementing decision trees and forests on a GPU, in: European Conference on Computer Vision, Springer, 2008, pp. 595–608.
[37] H. Grahn, N. Lavesson, M.H. Lapajne, D. Slat, CudaRF: a CUDA-based implementation of random forests, in: 2011 9th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA), IEEE, 2011, pp. 95–101.
[38] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning Springer Series in Statistics, vol. 1, Springer, Berlin, 2001.
[39] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, CRC Press, 1984.
[40] S.K. Murthy, S. Kasif, S. Salzberg, R. Beigel, OC1: a randomized algorithm for building oblique decision trees, in: Proceedings of AAAI, vol. 93, 1993, pp. 322–327.
[41] L. Zhang, P.N. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine, IEEE Trans. Cybern. 45 (10) (2015) 2165–2176.
[42] O.L. Mangasarian, E.W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, IEEE Trans. Pattern Anal. Mach. Intell. 28 (1) (2006) 69–74.
[43] B. Igelnik, Y.-H. Pao, Stochastic choice of basis functions in adaptive function approximation and the functional-link net, IEEE Trans. Neural Netw. 6 (6) (1995) 1320–1329.
[44] K. Rakesh, P. Suganthan, An ensemble of kernel ridge regression for multi-class classification, Procedia Comput. Sci. 108 (2017) 375–383, http://dx.doi.org/10.1016/j.procs.2017.05.109.
[45] T.M. Khoshgoftaar, M. Golawala, J. Van Hulse, An empirical study of learning from imbalanced data using random forest, in: 19th IEEE International Conference on Tools with Artificial Intelligence, 2007. ICTAI 2007, vol. 2, IEEE, 2007, pp. 310–317.
[46] M. Khalilia, S. Chakraborty, M. Popescu, Predicting disease risks from highly imbalanced data using random forest, BMC Med. Inform. Decis. Mak. 11 (1) (2011) 51.
[47] I. Brown, C. Mues, An experimental comparison of classification algorithms for imbalanced credit scoring data sets, Expert Syst. Appl. 39 (3) (2012) 3446–3453.
[48] F. Berzal, J.-C. Cubero, N. Marın, D. Sánchez, Building multi-way decision trees with numerical attributes, Inf. Sci. 165 (1) (2004) 73–90.
[49] D. Biggs, B. De Ville, E. Suen, A method of choosing multiway partitions for classification and decision trees, J. Appl. Stat. 18 (1) (1991) 49–62.
[50] M. Lichman, UCI machine learning repository, 2013 http://archive.ics.uci.edu/ml.
[51] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[52] P. Nemenyi, Distribution-free multiple comparisons, in: Biometrics, vol. 18, International Biometric SOC 1441 I ST, NW, Washington, DC, 1962, p. 263.