An Experimental Evaluation of Mixup Regression Forests

Juan J. Rodríguez[a], Mario Juez-Gil[a], Álvar Arnaiz-González[a], Ludmila I. Kuncheva[b]

[a]*Escuela Politécnica Superior, Universidad de Burgos, 09006 Burgos, SPAIN*
[b]*School of Computer Science and Electronic Engineering, Bangor University, Dean Street, Bangor LL57 1UT, UK*

Abstract

Over the past few decades, the remarkable prediction capabilities of ensemble methods have been used within a wide range of applications. Maximization of base-model ensemble accuracy and diversity are the keys to the heightened performance of these methods. One way to achieve diversity for training the base models is to generate artificial/synthetic instances for their incorporation with the original instances. Recently, the *mixup* method was proposed for improving the classification power of deep neural networks (Zhang et al., 2017). *Mixup* method generates artificial instances by combining pairs of instances and their labels, these new instances are used for training the neural networks promoting its regularization. In this paper, new regression tree ensembles trained with mixup, which we will refer to as Mixup Regression Forest, are presented and tested. The experimental study with 61 datasets showed that the mixup approach improved the results of both Random Forest and Rotation Forest.

Keywords: Mixup, Regression, Random Forest, Rotation Forest

1. Introduction

The idea that motivates this study, in relation to problems that ensemble techniques can solve, is that an increase in base-model diversity will improve ensemble performance, generalization, and robustness. Diversity is a key attribute of an ensemble, without which ensemble methods would not be as successful as they are (Kuncheva & Whitaker, 2003). It

Email addresses: jjrodriguez@ubu.es (Juan J. Rodríguez), mariojg@ubu.es (Mario Juez-Gil), alvarag@ubu.es (Álvar Arnaiz-González), l.i.kuncheva@bangor.ac.uk (Ludmila I. Kuncheva)

can be achieved in several ways: by using different methods for building the classifiers in the ensemble (heterogeneous ensemble), by using methods that build classifiers with random components, and by using different training sets. The focus of this paper rests on the last strategy, in particular, in making new instances that not found in the original set for creating different training sets.

Mixup has recently been proposed by Zhang et al. (2017) for training deep neural networks using combinations of pairs of examples and their labels. Given a training set where each example is (x, y), with an input, x, and a corresponding output, y, then the combined examples $(\tilde{x}, \tilde{y})$ are generated as

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j$$

where (x_i, y_i) and (x_j, y_j) are two examples, drawn at random from the training data, and $\lambda \in [0, 1]$. The values of λ were obtained using the Beta distribution: $\lambda \sim \text{Beta}(\alpha, \alpha)$, with $\alpha \in (0, \infty)$.

Some example mixup data projections can be seen in figures 1 and 2. Figure 1 shows a single input dataset where the input variable and the output variable are represented on the x axis and the y axis, respectively, and the instances are generated with mixup. Figure 2 shows a couple of examples: two two-input datasets and the mixup-generated instances. The output values of the original datasets are in $\{-1, 1\}$ and the output values of the datasets that are generated are in $[-1, 1]$. Figure 3 shows the predictions of a single random tree for the datasets shown in Figure 2.

Mixup differs from other data augmentation approaches, in so far as its outputs are also combined. The combination of the outputs to address regression problems is a straightforward procedure.

As shown in Figure 1, some of the examples generated with mixup are clearly noise. Although it can be detrimental, noise injection has previously been used as a strategy for building successful ensembles (Melville & Mooney, 2005; Frank & Pfahringer, 2006; Martínez-Muñoz & Suárez, 2005; Gónzalez et al., 2017). In mixup forests, the prevalence

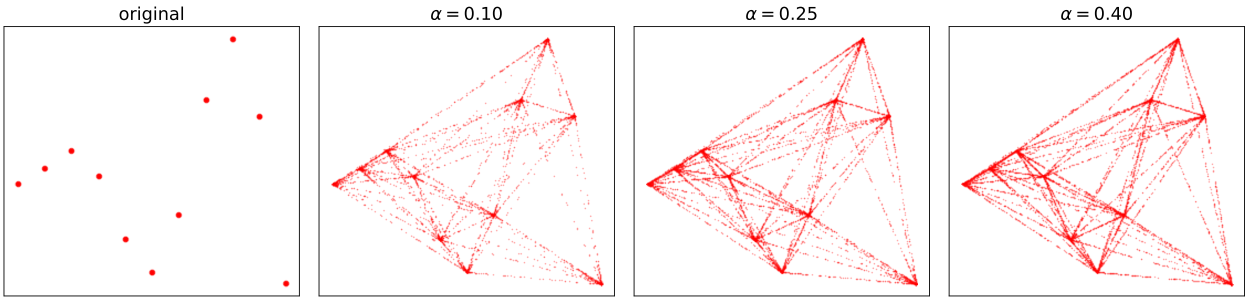


Figure 1: A regression problem dataset with a single input (x axis), and a single continuous output (y axis). Artificial instances are generated with mixup for $\alpha \in \{0.1, 0.25, 0.4\}$.

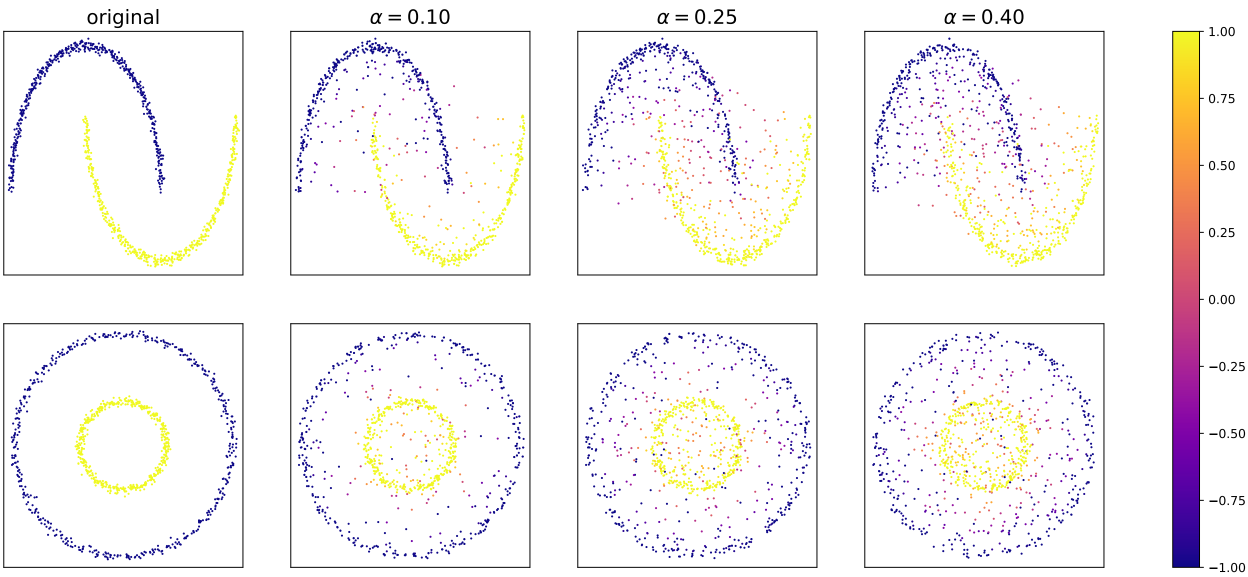


Figure 2: Two two-inputs datasets and the datasets generated with mixup for $\alpha \in \{0.1, 0.25, 0.4\}$. The output variables are shown in yellow and in blue.

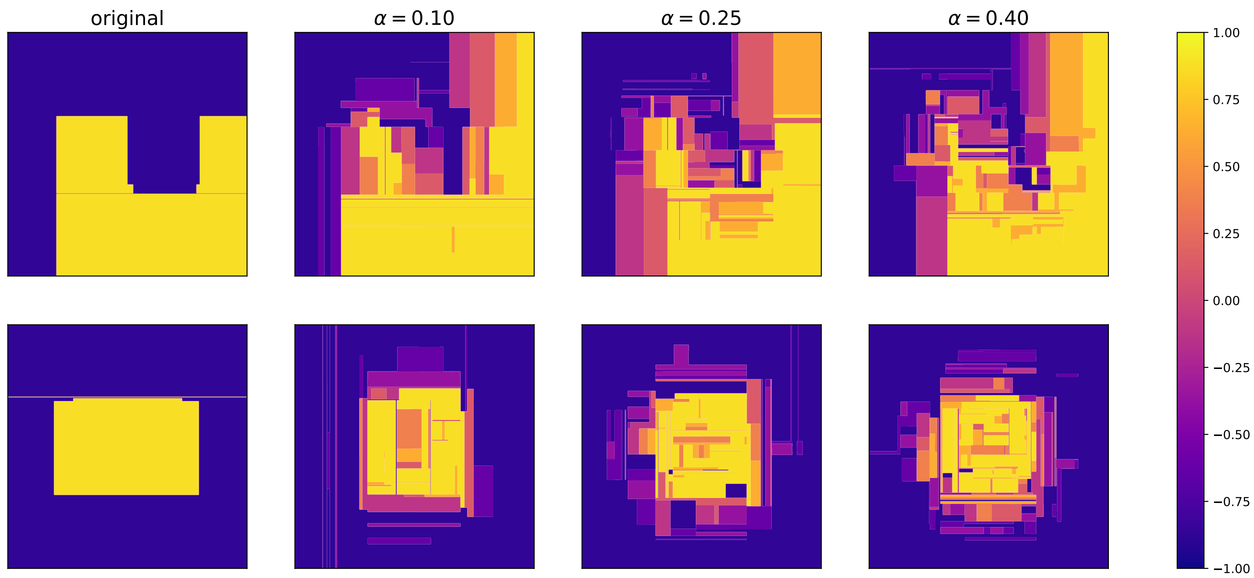


Figure 3: Predictions given by a single random tree trained with the corresponding datasets from Figure 2.

of these noisy examples can be controlled with the α value and the number of artificial examples that are generated.

Ensemble techniques have successfully been applied in various domains over the past few decades. Many works and several literature reviews have been published on both classification (Kuncheva, 2014) and regression (Mendes-Moreira et al., 2012) ensembles. Some illustrative examples of ensemble applications are detailed below.

In industrial environments, ensembles can be used as predictive models with adaptive capabilities, for example, to respond to incidences at processing plants (Soares & Araújo, 2015). Financial forecasting with ensembles has also been a very frequent research topic, among other examples, for the prediction of trading in stocks (Weng et al., 2018) and bankruptcy trends (Chen et al., 2020). It is also of great industrial interest, for example, in the construction industry, where ensembles have been used for the prediction of financial distress (Choi et al., 2018). Many techniques for credit risk assessment have been proposed, based on both statistics and Artificial Intelligence (AI) models; a task in which ensembles have demonstrated good performance (Marqués et al., 2012). In biometrics, improved recognition rates can be achieved using multimodal biometric systems that capture multiple biometric traits, e.g. fingerprint, iris and facial features; multimodal data learning in those fields can be addressed by using ensembles (Ross & Jain, 2003). The advantages and the convenience of ensemble learning to learn from multimodal features have likewise benefited several clinical practices (Tay et al., 2013). The sort of highly robust system required for image recognition tasks, such as facial recognition, can be provided by ensembles, to address the diversity of facial expressions and aging effects (Sirlantzis et al., 2008). Real-life problems, such as spam detection (Geng et al., 2007), translation of DNA sequences (García-Pedrajas et al., 2012), and the detection of credit-card fraud (Panigrahi et al., 2009), are known as imbalanced learning problems that can also be solved using ensemble techniques (Galar et al., 2012). The mixup data augmentation strategy proposed in this paper, might therefore lead to even better ensemble models for the aforementioned applications, as the artificial generation of instances has the potential to improve the performance of almost any ensemble method.

The contribution of this study relates to the novel use of the mixup approach. It demonstrates that artificial examples generated by mixup contribute to improved ensemble performance in regression tasks. Mixup is therefore considered for regression, mainly because of its simplicity: it can be used with all data types and needs no adjustments to the model.

The rest of the paper will be organized as follows. In section 2, a brief literature review of the most relevant works in this field will be presented. In section 3, the experimental setup will be described. Then the results will be presented and analyzed in Section 4. Finally, some concluding remarks and suggestions for future research work will be outlined in section 5.

2. Related works

Diversity between the members of an ensemble means that those ensembles are capable of better predictions than the individual ensemble members. One way to achieve diversity is by introducing artificial examples for training, for example through the mixup approach. Data augmentation with artificial examples has previously been used in many ensemble algorithms, some of which are detailed below.

In DECORATE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) (Melville & Mooney, 2003, 2005), instances are generated based on the distribution of the data. The labels of the new instances are assigned with a probability that is proportional to the inverse of the probability assigned by the current ensemble, because the purpose of the artificial instances is to increase diversity. In Bagging with Input Smearing (Frank & Pfahringer, 2006), the generation of artificial instances add noise to actual instances.

In imbalanced classification problems[1], artificial examples are commonly used for increasing the number of instances of the minority class/es. As with mixup, in SMOTE (Chawla et al., 2002), artificial instances are also obtained by combining pairs of instances. In this case, as both instances in a pair are of the same class, the label of the artificial instances is the same as the instances used to generate them. SMOTE was not originally proposed as an ensemble method and can in fact be used as a pre-processing step before the construction of a model. Nevertheless, it can also be directly used in ensembles, by training each base classifier with a different set of original and artificial instances. SMOTE has been combined with generic ensemble methods giving rise to SMOTEBoost (Chawla et al., 2003) and SMOTEBagging (Wang & Yao, 2009), among others.

There are many other methods for balancing datasets by augmenting the minority classes with artificial instances (Han et al., 2005; He et al., 2008; Menardi & Torelli, 2014; Zhu et al., 2017). Some of these methods, such as SMOTE, have also been adapted to regression problems (Torgo et al., 2013).

Likewise, highly sophisticated approaches exist for augmenting datasets. Most of those have been specifically designed for a given data type, for example, images (Tokozume et al., 2017, 2018; Inoue, 2018; Summers & Dinneen, 2019). Such approaches require training and adjusting a model, in order to generate the artificial instances (Mayo & Frank, 2017; Verma et al., 2018; Guo et al., 2018; Lindenbaum et al., 2018; Beckham et al., 2019).

Here, mixup was chosen as the simplest augmentation method and the significant advantage of its use with regression ensembles of random trees (Mixup Regression Forests) will be demonstrated in the following section.

[1] Imbalanced classification problems are those related to datasets and domains where one class has a much greater number of examples than another (Haixiang et al., 2017).

3. Experimental setting

The purpose of this experiment is to demonstrate the advantage of the mixup augmentation step. Two of the best state-of-the-art ensemble methods (singled out by extensive experimental studies (Random Forest (Breiman, 2001; Fernández-Delgado et al., 2014) and Rotation Forest (Rodríguez et al., 2006; Pardo et al., 2013; Bagnall et al., 2018)) are tested with and without the mixup step over a large collection of datasets. The experimental setup is presented below.

3.1. Datasets

Table 1 shows the main characteristics of the 61 regression datasets used in the experiments. All of them are available in the format used by Weka[2] (Hall et al., 2009). Thirty of the 61 datasets were collected by Luís Torgo[3].

3.2. Methods

The mixup method is used in combination with Random Forest (Breiman, 2001) and Rotation Forest (Rodríguez et al., 2006; Pardo et al., 2013). Both Random and Rotation Forest are used to transform the training dataset. In Random Forest, the dataset is sampled, whereas in Rotation Forest, it is rotated and then sampled. The mixup transformation can be done before or after the two above-mentioned ensemble transformations. Four methods are therefore available:

- MixRandFor: The dataset is augmented with mixup and then sampled.
- RandMixFor: The dataset is sampled and then the sample is augmented with mixup.
- MixRotFor: The dataset is augmented with mixup and then rotated.
- RotMixFor: The dataset is rotated and then augmented with mixup.

[2] `http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html`
[3] `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`

Dataset	Examples	Numeric	Nominal
2d-planes	40768	10	0
abalone	4177	7	1
ailerons	13750	40	0
auto-horse	205	17	8
auto-mpg	398	4	3
auto-price	159	15	0
auto93	93	16	6
bank-32nh	8192	32	0
bank-8FM	8192	8	0
baskball	96	4	0
bodyfat	252	14	0
bolts	40	7	0
breast-tumor	286	1	8
cal-housing	20640	8	0
cholesterol	303	6	7
cleveland	303	6	7
cloud	108	4	2
cpu	209	6	1
cpu-act	8192	21	0
cpu-small	8192	12	0
delta-ailerons	7129	5	0
delta-elevators	9517	6	0
detroit	13	13	0
diabetes-numeric	43	2	0
echo-months	130	6	3
elevators	16599	18	0
elusage	55	1	1
fishcatch	158	5	2
friedman	40768	10	0
fruitfly	125	2	2
gascons	27	4	0

Dataset	Examples	Numeric	Nominal
house-16H	22784	16	0
house-8L	22784	8	0
housing	506	12	1
hungarian	294	6	7
kin8nm	8192	8	0
longley	16	6	0
lowbwt	189	2	7
machine-cpu	209	6	0
mbagrade	61	1	1
meta	528	19	2
mv	40768	7	3
pbc	418	10	8
pharynx	195	1	10
pole	15000	48	0
pollution	60	15	0
puma32H	8192	32	0
puma8NH	8192	8	0
pw-linear	200	10	0
pyrimidines	74	27	0
quake	2178	3	0
schlvote	38	4	1
sensory	576	0	11
servo	167	0	4
sleep	62	7	0
stock	950	9	0
strike	625	5	1
triazines	186	60	0
veteran	137	3	4
vineyard	52	3	0
wisconsin	194	32	0

Table 1: Experimental dataset characteristics.

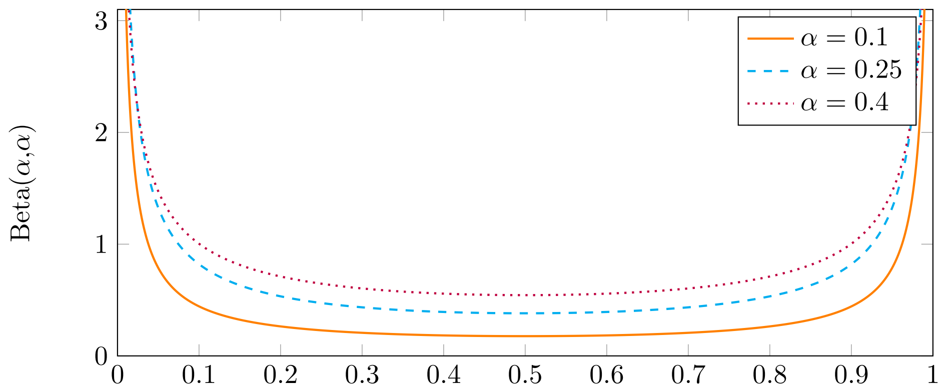


Figure 4: Beta distribution of the α values under consideration.

3.3. Settings

The experiments were performed using Weka (Hall et al., 2009). The default parameter's values of Random Forest and Rotation Forest were used, unless otherwise specified. For Random Forest, the default number of random attributes is $\log_2(m) + 1$ where m is the number of attributes. For Rotation Forest, the default size for each group of attributes is 3. The default method for constructing the trees in Rotation Forest, which only works for classification, is J48. Hence, REPTree, a tree method for regression, was used with no pruning, as ensembles generally work better with unstable models and pruning increases stability.

The results were generated using a 5×2-fold cross validation. The reported values are therefore averaged values from the 10 experiments. Three performance measures were calculated: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and correlation.

The size of each ensemble was set at 100. The number of artificial examples to be generated was set at 50% of the training data size. Three values were applied (0.10, 0.25, and 0.40) for the α values (in the Beta distribution), from the recommended range of $[0.1, 0, 4]$ in (Zhang et al., 2017). Figure 4 plots the Beta distribution for these α values.

One option for using mixup with nominal attributes is to transform them into numeric attributes. For example, one approach is to turn them into numerical values (that introduces an artificial order), and another is to turn them into binary attributes (greatly multiplying the attributes when there are many nominal values per attribute). Nevertheless, the mixing of two nominal value attributes was done in the experiments, by randomly selecting a single one. The probability of selecting the first nominal value is λ.

The number of artificial examples and the α value are hyper-parameters that can potentially improve the results when adjusted for each dataset.

4. Results and discussion

Tables 2, 3, and 4 show the results for RMSE, for MAE, and for correlation, respectively.

Pairwise comparisons. Tables 5 and 6 show the number of datasets for which the column method achieved better results than the row method. As 61 datasets were used in the experiments, a value greater than or equal to 31 will indicate that the column method has better results than the row method. It can be seen that the results are favorable for variants with mixup, especially for RMSE and correlation.

Relative scores. Figure 5 shows the boxplots of the relative scores, comparing the original method (Random or Rotation Forest) with the variants with mixup. The relative score for a given measure is defined as $(b-a)/a$ where a and b represent the performance of the original method and the performance of the variant method, respectively. When the measure is an error (RMSE or MAE), negative values of the score indicate that the variant is better. In contrast, positive values for correlation indicate that the variant is better. Each boxplot was obtained from the relative scores of the 61 datasets. The outliers were not included in the boxplots for the relative scores, as their inclusion would leave the boxes very small, because the relative scores of these few datasets (outliers) are much larger.

The boxplots and the signs of the median values are generally favorable for the variants with mixup. The only exceptions are RandMixFor and RotMixFor with $\alpha \in \{0.25, 0.40\}$ for MAE.

Influence of α. The following approach shows how the α values can affect the performance measures. For a given dataset, method and performance measure, the values of the measure were calculated for $\alpha = 0.1, 0.25, 0.4$ and then scaled to the interval $[0, 1]$. Then, a parabola

Table 2: Results for RMSE. The best result for each dataset is highlighted with a yellow background.

	Rand For	MixRand For-0.10	MixRand For-0.25	MixRand For-0.40	RandMix For-0.10	RandMix For-0.25	RandMix For-0.40	Rot For	MixRot For-0.10	MixRot For-0.25	MixRot For-0.40	RotMix For-0.10	RotMix For-0.25	RotMix For-0.40
2dplanes	1.109	1.112	1.112	1.112	1.112	1.112	1.112	1.09	1.043	1.04	1.039	1.043	1.04	1.04
abalone	2.193	2.189	2.187	2.179	2.181	2.179	2.173	2.111	2.114	2.113	2.113	2.115	2.114	2.115
ailerons	1.86e-04	1.86e-04	1.86e-04	1.86e-04	1.85e-04	1.86e-04	1.86e-04	1.73e-04	1.71e-04	1.72e-04	1.72e-04	1.71e-04	1.71e-04	1.72e-04
auto93	6.297	6.298	6.325	6.306	6.353	6.287	6.286	6.015	5.965	5.948	5.903	5.927	5.97	5.921
auto-horse	16.65	16.12	16.3	16.31	16.55	16.49	16.68	14.06	14.22	14.3	14.3	14.1	14.1	13.98
auto-mpg	2.933	2.943	2.906	2.929	2.906	2.921	2.883	2.819	2.81	2.819	2.82	2.813	2.827	2.825
auto-price	2481	2465	2442	2444	2474	2508	2492	2487	2423	2462	2447	2423	2449	2451
bank-32nh	0.08835	0.08845	0.08847	0.08859	0.08866	0.08888	0.08886	0.08542	0.08484	0.0851	0.08506	0.0849	0.08502	0.08507
bank-8FM	0.03268	0.0325	0.03254	0.03256	0.03266	0.03266	0.03273	0.03286	0.03274	0.03281	0.03279	0.03267	0.03265	0.03284
baskball	0.09483	0.09551	0.09505	0.0948	0.09368	0.09463	0.09384	0.09327	0.09367	0.09264	0.09313	0.09296	0.09319	0.09287
bodyfat	2.496	2.417	2.396	2.418	2.517	2.492	2.502	2.133	2.101	2.068	2.08	2.074	2.065	2.102
bolts	13.58	12.89	13.06	13.3	13.17	13.37	13.12	13.82	13.53	13.84	14.04	13.55	13.73	13.8
breast-tumor	10.87	10.98	10.89	10.84	10.79	10.74	10.7	10.53	10.61	10.58	10.52	10.68	10.63	10.64
cal-housing	50325	51196	52012	52442	51624	52436	52746	52983	53254	53875	54056	53150	53608	53927
cholesterol	52.46	52.2	52.48	52.02	52.23	51.97	52.08	51.2	51.34	51.42	51.2	51.55	51.66	51.31
cleveland	0.9146	0.9116	0.9028	0.9014	0.904	0.9087	0.9083	0.8903	0.8877	0.8903	0.8854	0.8891	0.8931	0.8877
cloud	0.5715	0.5643	0.5722	0.5681	0.5764	0.5721	0.5649	0.6	0.5915	0.5873	0.5877	0.5921	0.5913	0.5909
cpu	57.91	54.71	54.65	54.35	57.5	58.79	58.42	62.19	59.04	57.97	59.7	56.69	58.15	59.56
cpu-act	2.562	2.541	2.553	2.551	2.562	2.561	2.566	2.519	2.551	2.552	2.561	2.545	2.564	2.563
cpu-small	2.926	2.873	2.88	2.884	2.887	2.888	2.884	2.928	2.959	2.968	2.966	2.951	2.96	2.961
delta-ailerons	1.69e-04	1.67e-04	1.67e-04	1.67e-04	1.66e-04	1.67e-04	1.67e-04	1.70e-04	1.68e-04	1.68e-04	1.68e-04	1.68e-04	1.68e-04	1.68e-04
delta-elevators	1.46e-03	1.46e-03	1.46e-03	1.46e-03	1.46e-03	1.46e-03	1.45e-03	1.43e-03	1.43e-03	1.43e-03	1.43e-03	1.43e-03	1.43e-03	1.43e-03
detroit	46.99	44.51	45.65	44.77	46.24	46.84	46.62	70.75	52.58	53.31	52.66	50.55	50.15	50.69
diabetes-numeric	0.6433	0.6378	0.6305	0.6266	0.6328	0.6302	0.6297	0.6759	0.6752	0.6723	0.6666	0.6725	0.6655	0.6659
echo-months	12.16	12.04	12.01	12.07	12.11	12.14	12.23	12.03	11.98	12.09	12	12.05	11.97	12
elevators	3.14e-03	3.10e-03	3.10e-03	3.11e-03	3.14e-03	3.13e-03	3.11e-03	2.67e-03	2.63e-03	2.63e-03	2.64e-03	2.63e-03	2.64e-03	2.64e-03
elusage	16.42	16	15.84	15.67	16.14	15.77	15.8	13.41	13.47	13.27	13.36	13.29	13.3	13.16
fishcatch	85.79	85.15	83.25	82.61	86.57	83.75	83.73	95.79	78.7	80.06	80.55	78.96	80.38	79.12
fried	1.377	1.368	1.375	1.377	1.381	1.383	1.388	1.441	1.429	1.446	1.452	1.444	1.455	1.465
fruitfly	19.29	19.35	19.21	19.17	18.89	18.81	18.64	17.58	17.89	17.81	17.77	17.87	17.95	17.76
gascons	10.75	10.35	10.68	10.83	11.09	11.4	11.9	17.02	13.4	13.3	12.66	13.53	12.91	13.09
house-16H	32615	32465	32604	32587	32633	32778	32742	34145	33763	33884	33994	33737	33851	33931
house-8L	29979	29917	29912	29887	29887	29872	29908	30593	30483	30497	30520	30403	30519	30520
housing	3.598	3.559	3.549	3.572	3.535	3.53	3.524	3.721	3.633	3.637	3.643	3.615	3.62	3.646
hungarian	0.3707	0.3712	0.3692	0.3691	0.3676	0.3672	0.3661	0.3627	0.3612	0.3593	0.3588	0.3631	0.3616	0.3623
kin8nm	0.1503	0.1487	0.1492	0.1493	0.1501	0.1502	0.1504	0.1291	0.1279	0.1291	0.1298	0.128	0.1287	0.13
longley	1325	1331	1317	1323	1400	1366	1361	1730	1600	1574	1580	1559	1625	1582
lowbwt	462.9	468.5	466.5	469.2	460.1	468.6	466	457.2	461.7	461.6	460	462.8	458.2	461.6
machine-cpu	65.06	64.39	64.63	65.17	63.81	64.02	65.36	77.62	72.64	72.66	72.8	72.19	73.48	72.97
mbagrade	0.3755	0.3794	0.3761	0.3726	0.3673	0.3649	0.3634	0.3305	0.3403	0.3369	0.3317	0.3401	0.337	0.3327
meta	748.4	747.8	741.1	743.8	744.3	745.5	741.5	725.3	730.6	733.2	731.9	735.7	739.6	737.9
mv	0.2727	0.2719	0.3235	0.3686	0.3133	0.3958	0.4519	0.2231	0.2979	0.3672	0.4067	0.2394	0.2545	0.2626
pbc	920.2	922.6	918.7	918.1	918.7	921.2	920.3	880.5	879.4	880.5	881.3	882.1	879.8	880.7
pharynx	357.4	357.3	358.8	357.2	360.2	360.6	360.7	313.2	311.1	312.5	311.4	311.3	312.3	314
pol	7.315	7.562	7.814	7.947	7.839	8.097	8.27	5.268	5.573	5.996	6.23	5.693	6.106	6.364
pollution	48.92	48.23	48.55	48.72	49.25	49.26	49.27	46.84	47.28	47.6	47.27	46.85	47.29	47.64
puma32H	0.01687	0.01728	0.01743	0.01754	0.01761	0.01808	0.0183	0.01313	0.01359	0.01383	0.01393	0.01364	0.01385	0.01406
puma8NH	3.253	3.257	3.27	3.275	3.266	3.284	3.293	3.278	3.291	3.313	3.319	3.289	3.305	3.316
pw-linear	2.09	2.064	2.07	2.077	2.069	2.108	2.087	1.881	1.883	1.896	1.896	1.896	1.897	1.907
pyrim	0.1004	0.1008	0.1001	0.09966	0.09672	0.09815	0.09838	0.1198	0.1056	0.1048	0.1066	0.1067	0.1072	0.1075
quake	0.1981	0.1985	0.1972	0.1972	0.1967	0.1965	0.1963	0.1904	0.1933	0.1914	0.1909	0.1931	0.1915	0.1908
schlvote	1159316	1178402	1176517	1175270	1162099	1155803	1149687	1123137	1082854	1104934	1091608	1096670	1089156	1091314
sensory	0.7298	0.7321	0.733	0.7299	0.73	0.7327	0.7313	0.7228	0.7234	0.7221	0.7244	0.7217	0.7232	0.7224
servo	0.7638	0.7533	0.7802	0.782	0.7707	0.7813	0.7912	0.7451	0.7746	0.7693	0.7716	0.7407	0.761	0.7426
sleep	3.658	3.695	3.667	3.634	3.614	3.624	3.614	3.439	3.42	3.433	3.431	3.411	3.446	3.426
stock	0.9121	0.8822	0.887	0.8879	0.9033	0.9098	0.9033	0.8486	0.8422	0.8474	0.8506	0.8437	0.8488	0.8549
strike	537.2	544.3	538.7	537	534.7	534.7	534.2	516.2	513.7	510	513.2	511	513	510.6
triazines	0.1355	0.1348	0.1352	0.1346	0.1351	0.1343	0.135	0.1388	0.1363	0.1369	0.1371	0.1361	0.1367	0.1364
veteran	149.8	150.8	149.1	149.1	149.5	148.8	149.9	145.8	145.4	145.3	144.4	148.4	146.3	146.5
vineyard	2.622	2.599	2.607	2.607	2.601	2.607	2.622	3.021	2.906	2.928	2.918	2.935	2.919	2.937
wisconsin	33.26	33.48	33.49	33.25	33.31	33.07	33.15	33.01	32.9	32.88	32.88	32.91	32.91	32.84

Table 3: Results for MAE. The best result for each dataset is highlighted with a yellow background.

	Rand For	MixRand For-0.10	MixRand For-0.25	MixRand For-0.40	RandMix For-0.10	RandMix For-0.25	RandMix For-0.40	Rot For	MixRot For-0.10	MixRot For-0.25	MixRot For-0.40	RotMix For-0.10	RotMix For-0.25	RotMix For-0.40
2dplanes	0.8819	0.885	0.8845	0.8845	0.8842	0.8846	0.8846	0.8682	0.8313	0.8291	**0.8278**	0.8313	0.829	0.8287
abalone	1.553	1.549	1.545	1.54	1.54	1.539	1.533	**1.485**	1.489	1.487	1.485	1.487	1.489	1.487
ailerons	1.37e-04	1.37e-04	1.37e-04	1.37e-04	1.37e-04	1.37e-04	1.38e-04	1.28e-04	**1.25e-04**	1.26e-04	1.27e-04	1.26e-04	1.26e-04	1.27e-04
auto93	4.417	4.323	4.333	4.341	4.419	4.366	4.371	3.949	3.917	3.877	3.873	**3.86**	3.933	3.885
auto-horse	9.766	9.292	9.462	9.442	9.65	9.688	9.862	7.46	7.528	7.657	7.607	7.475	7.482	**7.38**
auto-mpg	2.117	2.118	2.098	2.111	2.095	2.115	2.098	2.004	**1.999**	2.009	2.016	2.009	2.011	2.024
auto-price	1552	1551	1541	1541	1560	1582	1561	1574	**1525**	1549	1543	1532	1550	1545
bank-32nh	0.06171	0.06221	0.0628	0.06312	0.06301	0.06379	0.06399	0.05811	**0.05796**	0.05855	0.05882	0.05796	0.05853	0.05887
bank-8FM	0.02353	**0.02346**	0.0235	0.02355	0.02358	0.02366	0.02376	0.02452	0.0245	0.02466	0.02469	0.02446	0.02452	0.02476
baskball	0.07279	0.07399	0.07354	0.07318	**0.07219**	0.07311	0.07251	0.07306	0.07363	0.07244	0.07299	0.07283	0.07312	0.07274
bodyfat	1.752	1.695	1.676	1.692	1.774	1.754	1.751	1.383	1.364	1.336	1.333	1.341	**1.326**	1.362
bolts	10.05	**9.534**	9.734	9.927	9.883	10.04	9.974	10.04	9.93	10.15	10.46	9.797	10.17	10.13
breast-tumor	8.629	8.695	8.638	8.613	8.57	8.522	8.504	8.383	8.437	8.425	**8.357**	8.501	8.468	8.467
cal-housing	**33431**	34315	35085	35529	34796	35579	35942	35978	36258	36991	37249	36159	36764	37149
cholesterol	40.03	39.56	39.82	39.54	39.75	39.66	39.54	39.01	39.02	39.16	**38.97**	39.18	39.37	39.13
cleveland	0.6794	0.6756	0.6735	0.6743	0.677	0.6817	0.6834	0.6463	**0.6449**	0.6474	0.6478	0.6453	0.6493	0.649
cloud	0.3381	0.3314	0.3338	0.3324	0.3356	0.3347	**0.3301**	0.3549	0.35	0.3459	0.3492	0.3493	0.3512	0.3509
cpu	20.25	19.16	19.26	18.86	19.37	19.7	19.69	20.08	17.05	16.87	17.66	**16.38**	17.01	17.6
cpu-act	1.8	1.808	1.826	1.826	1.83	1.835	1.836	**1.765**	1.8	1.812	1.816	1.8	1.818	1.819
cpu-small	2.04	**2.035**	2.047	2.048	2.05	2.05	2.053	2.076	2.112	2.131	2.133	2.108	2.126	2.133
delta-ailerons	1.17e-04	1.16e-04	**1.15e-04**	1.16e-04	**1.15e-04**	**1.15e-04**	1.16e-04	1.17e-04	1.16e-04	1.16e-04	1.16e-04	1.16e-04	1.16e-04	1.16e-04
delta-elevators	1.09e-03	1.09e-03	1.09e-03	1.09e-03	1.09e-03	1.09e-03	1.09e-03	**1.07e-03**	1.07e-03	1.07e-03	1.07e-03	1.07e-03	1.07e-03	1.07e-03
detroit	35.71	**34.8**	35.55	34.85	35.13	35.66	35.76	55.09	40.13	40.63	40	38.04	37.85	38.36
diabetes-numeric	0.5119	0.5078	0.5014	**0.4945**	0.5035	0.5007	0.4996	0.5487	0.5376	0.5333	0.529	0.5369	0.5299	0.528
echo-months	9.625	**9.436**	9.516	9.595	9.675	9.755	9.826	9.565	9.512	9.674	9.591	9.563	9.529	9.547
elevators	2.12e-03	2.09e-03	2.09e-03	2.10e-03	2.12e-03	2.11e-03	2.10e-03	1.84e-03	**1.81e-03**	1.82e-03	1.82e-03	1.81e-03	1.82e-03	1.82e-03
elusage	12.61	12.21	12.1	11.92	12.39	12.2	12.2	9.894	10.04	9.919	9.951	9.952	9.903	**9.817**
fishcatch	53.23	52.17	51.15	51.17	52.92	52	52.08	55.35	47.9	48.8	49.15	**47.74**	48.63	48.27
fried	1.087	**1.08**	1.085	1.086	1.089	1.092	1.096	1.136	1.128	1.142	1.147	1.139	1.149	1.158
fruitfly	14.49	14.53	14.44	14.45	14.18	14.1	13.99	**12.99**	13.33	13.28	13.18	13.32	13.38	13.22
gascons	8.42	**8.032**	8.331	8.48	8.635	8.775	9.181	13.29	10.42	10.25	9.755	10.39	10.04	10.05
house-16H	16388	**16326**	16407	16424	16418	16471	16503	17583	17332	17520	17624	17315	17453	17593
house-8L	15814	15784	15787	15790	**15781**	15807	15850	16417	16263	16349	16409	16233	16354	16408
housing	2.389	2.37	2.377	2.372	2.368	2.379	2.373	2.415	2.371	2.372	2.379	**2.352**	2.361	2.373
hungarian	0.2663	0.267	0.2674	0.2703	0.2684	0.2721	0.274	**0.2507**	0.251	0.254	0.2555	0.251	0.252	0.2549
kin8nm	0.1196	0.1187	0.1194	0.1196	0.1202	0.1205	0.121	0.1022	**0.1015**	0.1028	0.1036	0.1015	0.1025	0.1037
longley	1145	1127	1116	**1106**	1187	1156	1149	1428	1315	1299	1296	1281	1328	1299
lowbwt	363.8	365.6	362.9	365.4	**359.7**	364	361	362.3	362.5	360.5	360.3	363.7	360.4	362.1
machine-cpu	30.76	30.36	30.57	30.81	**30.24**	30.69	30.8	35.33	33.06	33.2	32.88	32.91	33.05	33.13
mbagrade	0.2841	0.2882	0.2861	0.2835	0.2787	0.2755	0.2758	0.2521	0.2544	0.253	**0.2503**	0.2546	0.2536	0.2509
meta	145.2	146	**144.9**	146.6	145.2	146.8	146.4	148.5	147	147.2	148.9	148.1	149.7	149.1
mv	0.18	0.1774	0.2155	0.2444	0.208	0.2622	0.2968	**0.1515**	0.2158	0.2759	0.3061	0.1661	0.1814	0.1896
pbc	717.7	720.2	720.4	720.2	719.3	724.5	722.3	695.2	695.8	697.7	697.3	696.5	**694.3**	698.6
pharynx	278.5	279	281	278.8	282.2	282.5	282.6	234.2	232.2	232.6	**231.8**	232.3	233.3	234.2
pol	4.134	4.391	4.64	4.766	4.632	4.864	5.032	**2.646**	2.994	3.398	3.621	3.068	3.465	3.687
pollution	37.37	37.02	37.22	37.12	37.66	37.65	37.81	**35.39**	36.1	36.09	35.9	35.6	36.01	36.47
puma32H	0.01299	0.01328	0.01342	0.0135	0.01355	0.01392	0.01407	**0.01047**	0.01085	0.01106	0.01114	0.01089	0.01107	0.01124
puma8NH	**2.528**	2.55	2.572	2.587	2.572	2.6	2.618	2.589	2.61	2.642	2.653	2.608	2.634	2.652
pw-linear	1.635	1.612	1.609	1.618	1.617	1.641	1.628	1.449	**1.439**	1.46	1.45	1.452	1.45	1.464
pyrim	0.06158	0.062	0.06161	0.06177	**0.06099**	0.0614	0.06139	0.07847	0.06655	0.06653	0.06726	0.06728	0.06793	0.06794
quake	0.1549	0.155	0.1541	0.1541	0.154	0.1539	0.1537	**0.1494**	0.151	0.15	0.1497	0.1509	0.15	0.1497
schlvote	**639146**	659770	667857	664392	653918	656491	665144	677601	652816	666136	657503	656814	651958	656596
sensory	0.5857	0.5864	0.5849	0.5829	0.5842	0.586	0.5856	0.5794	0.5792	**0.5777**	0.58	0.5793	0.5787	0.5785
servo	0.4635	0.4578	0.4801	0.4866	0.4772	0.4908	0.5022	**0.4327**	0.4642	0.4693	0.4779	0.4454	0.4645	0.4567
sleep	2.937	2.957	2.955	2.941	2.907	2.91	2.906	2.688	2.679	2.682	2.683	2.664	2.699	**2.657**
stock	0.669	0.6516	0.6547	0.657	0.6668	0.6717	0.6699	0.6361	0.6344	0.6371	0.6385	**0.6338**	0.6388	0.6444
strike	**211.9**	214.7	212.9	212.6	212.9	212.3	212.2	245.9	230.4	231.2	237	229.3	235.3	236.7
triazines	0.09521	0.09518	0.09548	**0.09497**	0.09498	0.09503	0.09517	0.09955	0.09742	0.09805	0.09784	0.0971	0.09773	0.09744
veteran	95.34	95.48	94.23	93.92	94.44	93.98	94.81	90.53	**89.82**	90.37	90.27	91.56	91.09	91.97
vineyard	1.95	1.949	1.961	1.961	**1.932**	1.956	1.961	2.311	2.199	2.224	2.216	2.22	2.212	2.219
wisconsin	28.02	28.14	28.27	28.04	28.07	27.88	27.98	27.75	27.68	27.69	**27.66**	27.81	27.76	27.66

Table 4: Results for correlation. The best result for each dataset is highlighted with a yellow background.

	Rand For	MixRand For-0.10	MixRand For-0.25	MixRand For-0.40	RandMix For-0.10	RandMix For-0.25	RandMix For-0.40	Rot For	MixRot For-0.10	MixRot For-0.25	MixRot For-0.40	RotMix For-0.10	RotMix For-0.25	RotMix For-0.40
2dplanes	0.9677	0.9675	0.9675	0.9675	0.9675	0.9675	0.9675	0.9688	0.9714	0.9716	0.9717	0.9714	0.9716	0.9716
abalone	0.7342	0.7353	0.7358	0.7377	0.7374	0.7378	0.7395	0.7566	0.7555	0.7558	0.7560	0.7553	0.7558	0.7555
ailerons	0.8937	0.8945	0.8946	0.8944	0.8949	0.8948	0.8940	0.9077	0.9105	0.9096	0.9090	0.9104	0.9098	0.9093
auto93	0.8073	0.8102	0.8061	0.8098	0.8036	0.8118	0.8123	0.8044	0.8091	0.8107	0.8150	0.8130	0.8101	0.8147
auto-horse	0.9253	0.9285	0.9268	0.9268	0.9264	0.9276	0.9255	0.9411	0.9400	0.9399	0.9402	0.9414	0.9415	0.9428
auto-mpg	0.9281	0.9277	0.9297	0.9289	0.9298	0.9294	0.9316	0.9341	0.9346	0.9343	0.9343	0.9343	0.9337	0.9338
auto-price	0.9134	0.9132	0.9168	0.9162	0.9149	0.9119	0.9133	0.9126	0.9170	0.9144	0.9160	0.9173	0.9165	0.9157
bank-32nh	0.7079	0.7081	0.7096	0.7100	0.7090	0.7094	0.7101	0.7209	0.7254	0.7242	0.7250	0.7252	0.7245	0.7248
bank-8FM	0.9772	0.9775	0.9776	0.9777	0.9775	0.9776	0.9777	0.9776	0.9779	0.9780	0.9782	0.9780	0.9782	0.9781
baskball	0.5190	0.5110	0.5149	0.5192	0.5309	0.5231	0.5335	0.5306	0.5286	0.5397	0.5367	0.5368	0.5354	0.5377
bodyfat	0.9619	0.9641	0.9655	0.9647	0.9623	0.9627	0.9630	0.9744	0.9758	0.9766	0.9762	0.9762	0.9766	0.9764
bolts	0.8821	0.9004	0.8965	0.8917	0.8978	0.8942	0.9055	0.8840	0.8884	0.8820	0.8796	0.8932	0.8876	0.8863
breast-tumor	0.1622	0.1568	0.1622	0.1659	0.1690	0.1708	0.1748	0.2034	0.1986	0.1970	0.2044	0.1916	0.1996	0.1924
cal-housing	0.9011	0.8978	0.8947	0.8930	0.8963	0.8932	0.8921	0.8903	0.8892	0.8868	0.8862	0.8896	0.8880	0.8869
cholesterol	0.1551	0.1633	0.1518	0.1718	0.1605	0.1738	0.1591	0.2126	0.2058	0.2007	0.2108	0.2009	0.1987	0.2114
cleveland	0.6798	0.6801	0.6900	0.6922	0.6916	0.6860	0.6904	0.6977	0.6993	0.6980	0.7021	0.6982	0.6952	0.7003
cloud	0.8693	0.8730	0.8701	0.8712	0.8675	0.8717	0.8742	0.8528	0.8611	0.8608	0.8608	0.8578	0.8585	0.8588
cpu	0.9508	0.9536	0.9538	0.9554	0.9513	0.9484	0.9495	0.9327	0.9341	0.9386	0.9339	0.9384	0.9367	0.9358
cpu-act	0.9903	0.9905	0.9905	0.9905	0.9904	0.9905	0.9904	0.9907	0.9906	0.9907	0.9906	0.9907	0.9906	0.9906
cpu-small	0.9873	0.9878	0.9877	0.9877	0.9877	0.9877	0.9877	0.9873	0.9872	0.9872	0.9872	0.9873	0.9873	0.9873
delta-ailerons	0.8310	0.8357	0.8353	0.8348	0.8366	0.8361	0.8350	0.8313	0.8355	0.8353	0.8351	0.8355	0.8349	0.8349
delta-elevators	0.7886	0.7892	0.7895	0.7898	0.7897	0.7906	0.7909	0.7994	0.7978	0.7985	0.7985	0.7979	0.7983	0.7987
detroit	0.8533	0.8663	0.8542	0.8731	0.8661	0.8613	0.8594	0.2396	0.8758	0.8805	0.9049	0.8954	0.9126	0.8977
diabetes-numeric	0.5527	0.5793	0.5840	0.5933	0.5660	0.5640	0.5760	0.4812	0.4740	0.4744	0.4954	0.4752	0.4911	0.4926
echo-months	0.6511	0.6583	0.6625	0.6567	0.6533	0.6551	0.6467	0.6620	0.6635	0.6563	0.6633	0.6587	0.6628	0.6625
elevators	0.8907	0.8937	0.8939	0.8939	0.8915	0.8925	0.8939	0.9243	0.9265	0.9267	0.9263	0.9263	0.9261	0.9264
elusage	0.7646	0.7799	0.7856	0.7902	0.7732	0.7875	0.7862	0.8576	0.8566	0.8585	0.8569	0.8594	0.8592	0.8623
fishcatch	0.9719	0.9729	0.9742	0.9748	0.9710	0.9730	0.9730	0.9610	0.9765	0.9757	0.9753	0.9765	0.9752	0.9764
fried	0.9631	0.9638	0.9638	0.9639	0.9635	0.9638	0.9637	0.9622	0.9633	0.9628	0.9628	0.9626	0.9624	0.9621
fruitfly	-0.1610	-0.1420	-0.1435	-0.1382	-0.1570	-0.1432	-0.1429	-0.1355	-0.1271	-0.1327	-0.1359	-0.1275	-0.1455	-0.1324
gascons	0.9772	0.9792	0.9782	0.9783	0.9759	0.9762	0.9744	0.9196	0.9542	0.9544	0.9633	0.9546	0.9605	0.9606
house-16H	0.7922	0.7947	0.7930	0.7936	0.7929	0.7913	0.7921	0.7803	0.7830	0.7821	0.7812	0.7835	0.7829	0.7822
house-8L	0.8241	0.8250	0.8250	0.8254	0.8255	0.8256	0.8251	0.8176	0.8185	0.8185	0.8183	0.8196	0.8183	0.8183
housing	0.9247	0.9264	0.9266	0.9256	0.9282	0.9282	0.9283	0.9178	0.9214	0.9213	0.9210	0.9222	0.9221	0.9210
hungarian	0.6391	0.6374	0.6424	0.6421	0.6460	0.6464	0.6505	0.6595	0.6626	0.6668	0.6678	0.6582	0.6613	0.6601
kin8nm	0.8336	0.8393	0.8391	0.8398	0.8373	0.8384	0.8386	0.8891	0.8925	0.8915	0.8907	0.8922	0.8924	0.8905
longley	0.9549	0.9511	0.9534	0.9525	0.9497	0.9533	0.9545	0.9133	0.9222	0.9246	0.9220	0.9270	0.9183	0.9262
lowbwt	0.7755	0.7700	0.7721	0.7691	0.7790	0.7708	0.7758	0.7840	0.7779	0.7793	0.7808	0.7773	0.7821	0.7789
machine-cpu	0.9291	0.9296	0.9290	0.9280	0.9341	0.9322	0.9294	0.8951	0.9048	0.9071	0.9052	0.9092	0.9049	0.9049
mbagrade	0.1402	0.1373	0.1327	0.1383	0.1544	0.1573	0.1599	0.3457	0.2740	0.2799	0.3075	0.2756	0.2907	0.3133
meta	0.3405	0.3443	0.3550	0.3473	0.3522	0.3423	0.3552	0.2595	0.2795	0.2880	0.2828	0.2782	0.2745	0.2658
mv	0.9997	0.9997	0.9997	0.9996	0.9997	0.9995	0.9994	0.9998	0.9997	0.9996	0.9995	0.9998	0.9998	0.9998
pbc	0.5686	0.5610	0.5668	0.5681	0.5676	0.5643	0.5663	0.6097	0.6116	0.6104	0.6096	0.6073	0.6109	0.6106
pharynx	0.6351	0.6214	0.6209	0.6283	0.6282	0.6220	0.6338	0.6690	0.6745	0.6760	0.6752	0.6737	0.6725	0.6681
pol	0.9860	0.9854	0.9846	0.9842	0.9844	0.9836	0.9831	0.9924	0.9918	0.9909	0.9903	0.9915	0.9905	0.9899
pollution	0.6756	0.6862	0.6786	0.6790	0.6682	0.6671	0.6691	0.7036	0.6937	0.6853	0.7003	0.7018	0.6966	0.6883
puma32H	0.8947	0.8940	0.8944	0.8959	0.8924	0.8904	0.8880	0.9338	0.9333	0.9346	0.9352	0.9325	0.9345	0.9340
puma8NH	0.8161	0.8163	0.8160	0.8162	0.8164	0.8159	0.8161	0.8162	0.8156	0.8149	0.8149	0.8159	0.8156	0.8153
pw-linear	0.9013	0.9045	0.9029	0.9033	0.9051	0.9036	0.9037	0.9186	0.9194	0.9195	0.9197	0.9178	0.9190	0.9181
pyrim	0.6091	0.6017	0.6106	0.6123	0.6356	0.6194	0.6219	0.3970	0.5646	0.5837	0.5616	0.5686	0.5480	0.5494
quake	0.1188	0.1167	0.1208	0.1162	0.1213	0.1192	0.1181	0.1050	0.1097	0.1154	0.1156	0.1089	0.1143	0.1169
schlvote	0.4697	0.4801	0.4780	0.5011	0.4950	0.4965	0.5284	0.4001	0.4742	0.4675	0.4802	0.4437	0.4797	0.4958
sensory	0.4709	0.4655	0.4642	0.4714	0.4687	0.4638	0.4669	0.4908	0.4880	0.4906	0.4850	0.4937	0.4895	0.4917
servo	0.8759	0.8812	0.8749	0.8740	0.8759	0.8765	0.8760	0.8864	0.8785	0.8831	0.8849	0.8885	0.8817	0.8899
sleep	0.6236	0.6043	0.6185	0.6167	0.6399	0.6374	0.6389	0.6678	0.6721	0.6709	0.6708	0.6732	0.6681	0.6731
stock	0.9903	0.9911	0.9911	0.9911	0.9907	0.9908	0.9910	0.9918	0.9921	0.9920	0.9920	0.9920	0.9920	0.9919
strike	0.3985	0.3801	0.3879	0.3882	0.3974	0.3933	0.3917	0.4268	0.3686	0.3847	0.3683	0.3852	0.3734	0.3884
triazines	0.5109	0.5190	0.5139	0.5221	0.5144	0.5229	0.5135	0.4924	0.5093	0.5061	0.5037	0.5148	0.5058	0.5106
veteran	0.3744	0.3586	0.3779	0.3786	0.3749	0.3791	0.3578	0.3865	0.4030	0.4028	0.4053	0.3782	0.3956	0.3961
vineyard	0.8111	0.8135	0.8136	0.8144	0.8181	0.8176	0.8178	0.7449	0.7758	0.7751	0.7799	0.7743	0.7788	0.7747
wisconsin	0.3056	0.2885	0.2879	0.2983	0.2961	0.3129	0.3078	0.3323	0.3333	0.3384	0.3389	0.3384	0.3349	0.3407
MEAN	0.7244	0.7253	0.7261	0.7279	0.7276	0.7277	0.7286	0.7182	0.7335	0.7342	0.7355	0.7338	0.7341	0.7354

Table 5: Comparisons of Random Forest variants. Each cell shows the number of datasets where the column method is better than the row method.

(a) RMSE

	RandFor	MixRandFor-0.10	MixRandFor-0.25	MixRandFor-0.40	RandMixFor-0.10	RandMixFor-0.25	RandMixFor-0.40	Total
RandFor		36	39	45	39	39	32	230
MixRandFor-0.10	24		29	30	27	28	28	166
MixRandFor-0.25	21	30		31	29	25	25	161
MixRandFor-0.40	15	29	28		25	25	20	142
RandMixFor-0.10	22	34	31	36		25	27	175
RandMixFor-0.25	21	31	34	34	36		33	189
RandMixFor-0.40	28	31	34	39	34	26		192
Total	131	191	195	215	190	168	165	

(b) MAE

	RandFor	MixRandFor-0.10	MixRandFor-0.25	MixRandFor-0.40	RandMixFor-0.10	RandMixFor-0.25	RandMixFor-0.40	Total
RandFor		35	36	33	33	27	28	192
MixRandFor-0.10	25		25	28	26	22	22	148
MixRandFor-0.25	24	34		25	29	17	19	148
MixRandFor-0.40	27	31	35		28	16	18	155
RandMixFor-0.10	27	34	30	31		19	18	159
RandMixFor-0.25	33	38	42	44	40		27	224
RandMixFor-0.40	33	38	42	42	43	33		231
Total	169	210	210	203	199	134	132	

(c) Correlation

	RandFor	MixRandFor-0.10	MixRandFor-0.25	MixRandFor-0.40	RandMixFor-0.10	RandMixFor-0.25	RandMixFor-0.40	Total
RandFor		40	40	46	44	44	43	257
MixRandFor-0.10	21		33	39	31	33	33	190
MixRandFor-0.25	21	28		40	34	31	35	189
MixRandFor-0.40	15	22	21		24	30	30	142
RandMixFor-0.10	17	30	27	36		30	31	171
RandMixFor-0.25	17	28	30	31	31		37	174
RandMixFor-0.40	18	28	26	31	30	24		157
Total	109	176	177	223	194	192	209	

Table 6: Comparisons of Rotation Forest variants. Each cell shows the number of datasets where the column method is better than the row method.

(a) RMSE

	RotFor	MixRotFor-0.10	MixRotFor-0.25	MixRotFor-0.40	RotMixFor-0.10	RotMixFor-0.25	RotMixFor-0.40	Total
RotFor		38	35	36	37	34	36	216
MixRotFor-0.10	23		24	24	28	22	22	143
MixRotFor-0.25	26	36		28	36	26	26	178
MixRotFor-0.40	25	36	30		37	30	26	184
RotMixFor-0.10	24	30	24	23		20	24	145
RotMixFor-0.25	27	37	33	29	39		24	189
RotMixFor-0.40	25	38	33	33	36	35		200
Total	150	215	179	173	213	167	158	

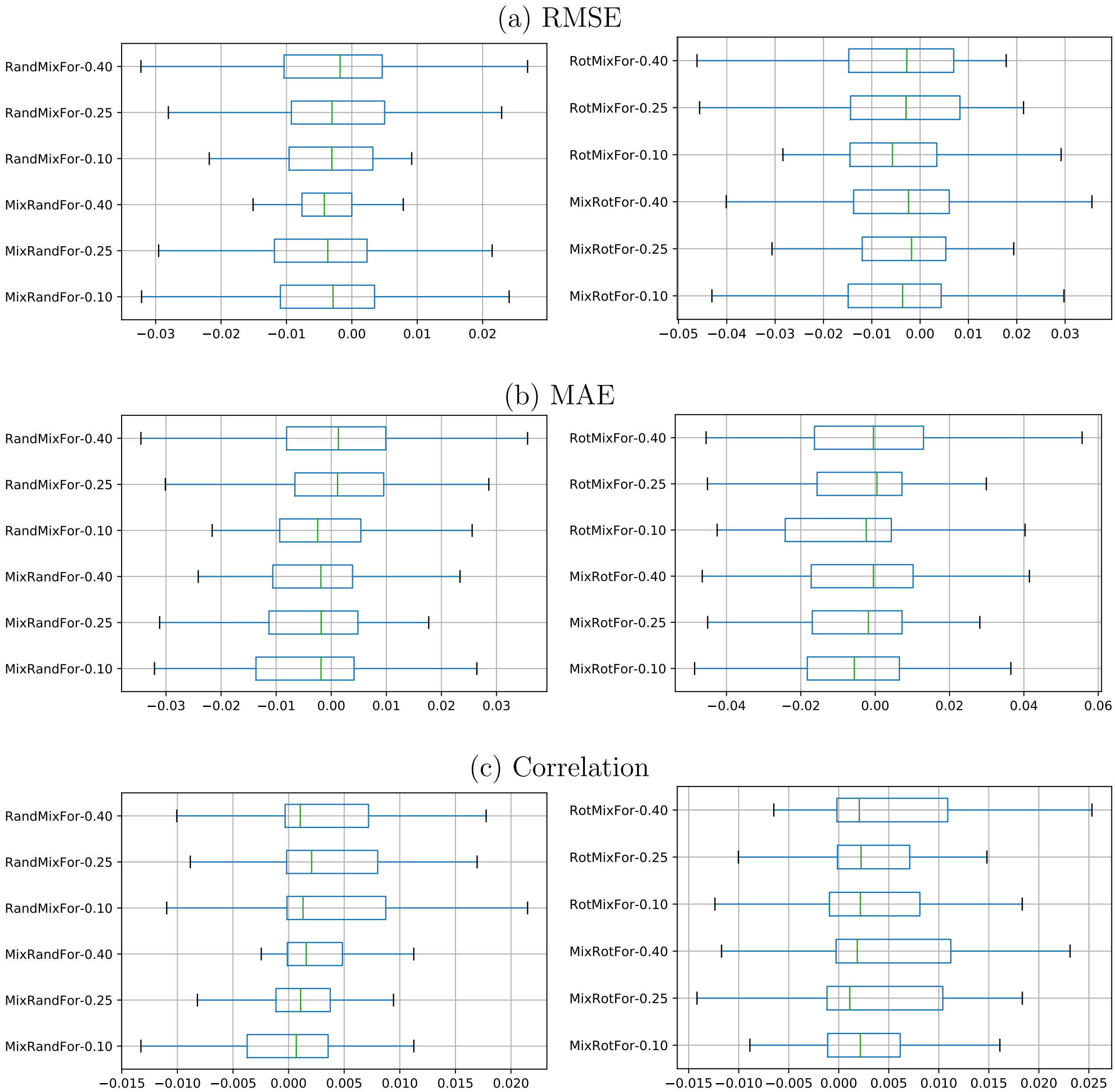
(b) MAE

	RotFor	MixRotFor-0.10	MixRotFor-0.25	MixRotFor-0.40	RotMixFor-0.10	RotMixFor-0.25	RotMixFor-0.40	Total
RotFor		39	31	31	35	29	32	197
MixRotFor-0.10	22		20	22	34	20	21	139
MixRotFor-0.25	30	40		27	38	29	22	186
MixRotFor-0.40	30	38	32		39	32	23	194
RotMixFor-0.10	26	25	21	21		19	20	132
RotMixFor-0.25	32	40	30	28	40		24	194
RotMixFor-0.40	29	39	38	36	40	35		217
Total	169	221	172	165	226	164	142	

(c) Correlation

	RotFor	MixRotFor-0.10	MixRotFor-0.25	MixRotFor-0.40	RotMixFor-0.10	RotMixFor-0.25	RotMixFor-0.40	Total
RotFor		40	41	41	40	42	43	247
MixRotFor-0.10	21		30	31	33	30	29	174
MixRotFor-0.25	20	31		31	35	25	34	176
MixRotFor-0.40	20	30	30		31	21	29	161
RotMixFor-0.10	21	27	26	29		23	32	158
RotMixFor-0.25	19	31	35	40	38		32	195
RotMixFor-0.40	18	32	27	32	29	29		167
Total	119	191	189	204	206	170	199	

Figure 5: Boxplots of relative performances. The start and end of the box are the first and third quartiles, the band inside the box is the median. Outliers are not shown.



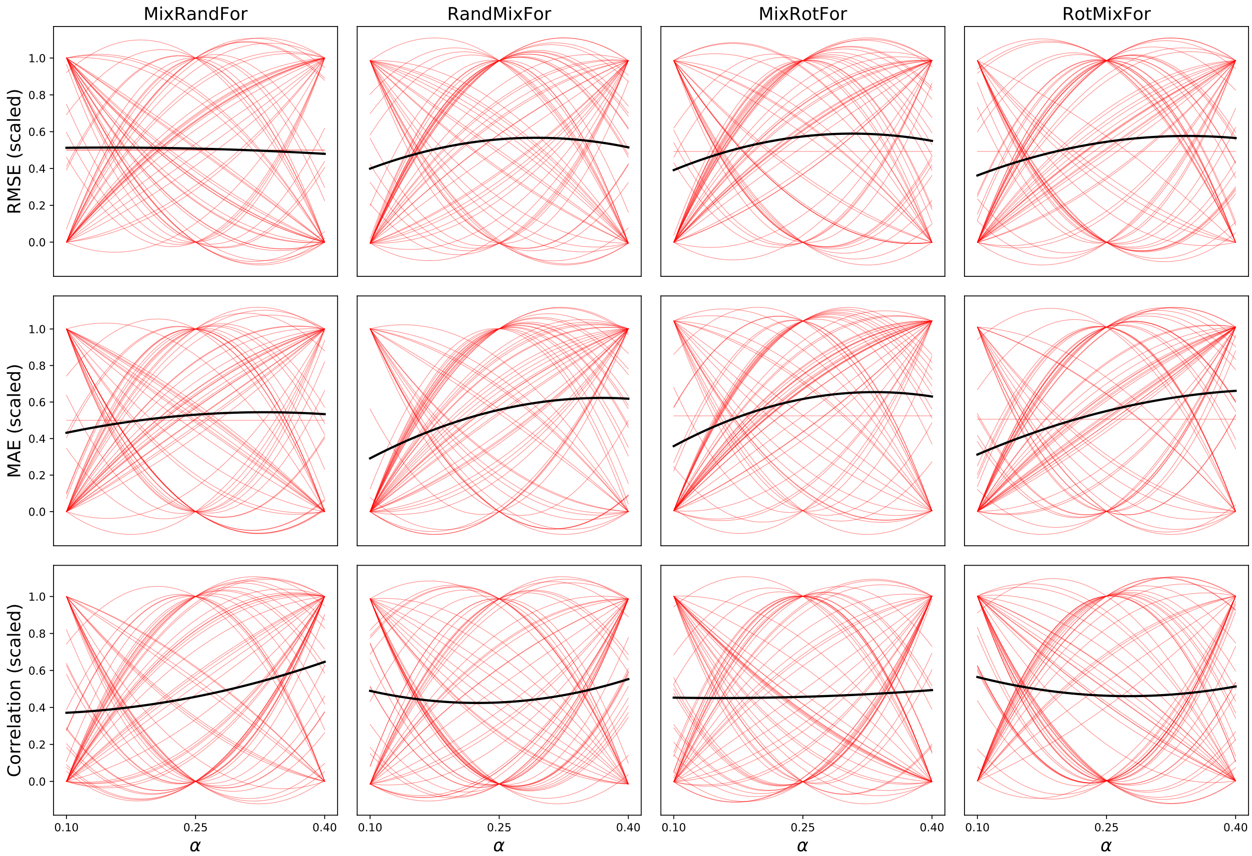


Figure 6: Scaled measures as a function of α. Each red parabola corresponds to a single dataset; the black parabola plots the average values.

was fitted to the three points. Figure 6 shows these parabolas, and a final parabola (shown in black) obtained by averaging the scaled values across all the datasets. There is no consistent pattern of the parabolas for the individual datasets, indicating that the optimal value of α depends on the dataset.

Average ranks. Figure 7 shows the average ranks for Random Forest and its variants with mixup. The best method is assigned rank 1, the second is assigned rank 2, and so on. The worst method is assigned rank 7, as we are comparing 7 alternatives for each ensemble method (the original ensemble, MixXXX for three values of α, and XXXMix for three values of α.) With the aim of evaluating whether some variants are significantly better than the starting method (without mixup), the Bonferroni-Dunn test was performed over the

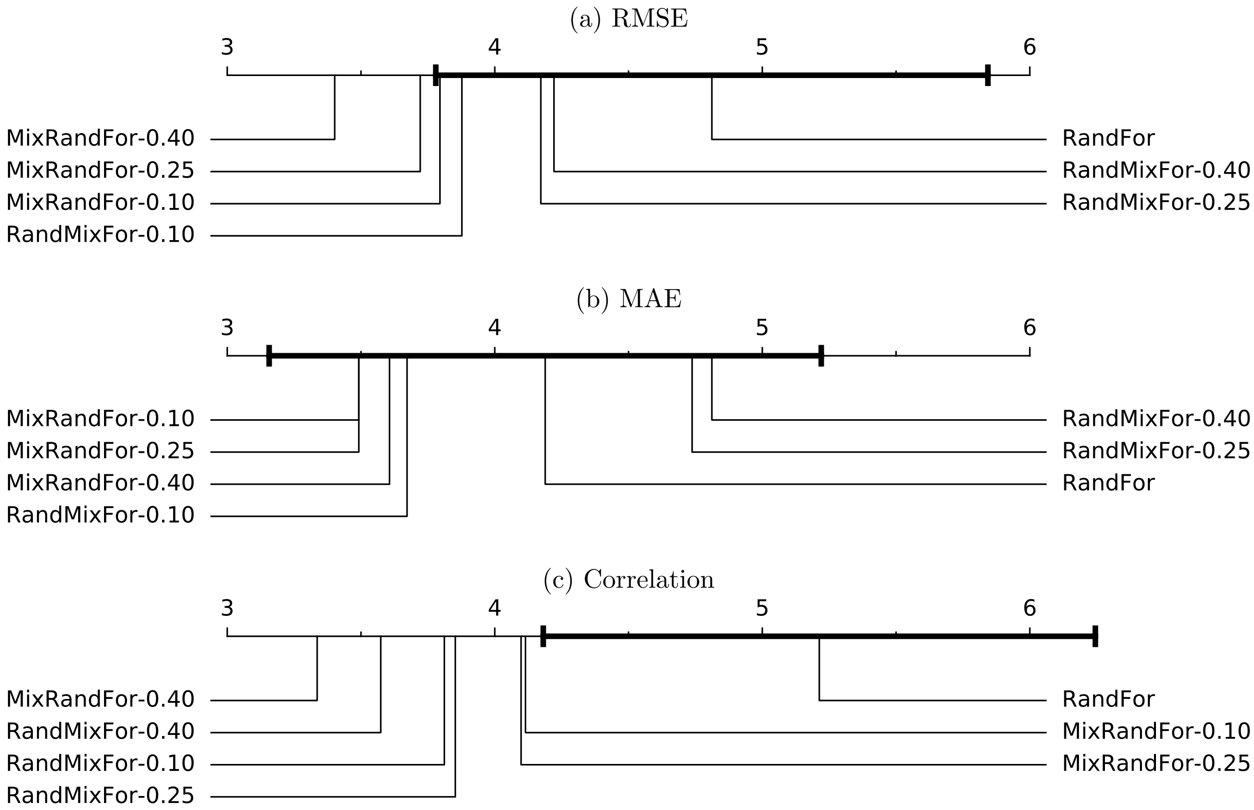


Figure 7: Comparison of Random Forest against variants with Mixup, with the Bonferroni-Dunn test. The marked interval spans the critical value and is centered at the mean rank for Random Forest. Variants with ranks outside the marked interval are significantly different ($p < 0.05$) than Random Forest.

ranks (Demšar, 2006) using Random or Rotation Forest as the control classifier. Random Forest without mixup had the worst average rank for RMSE and correlation. The advantage of mixup for MAE was less clear, as two variants with mixup were worse.

Figure 8 shows the average ranks for Rotation Forest and its mixup variants. In the same way as Random Forest, Rotation Forest without mixup shows the worst average rank for RMSE and correlation. The three variants with mixup were worse for MAE, while the other three were better.

Table 7 shows the average ranks for Random Forest, Rotation Forest, and their variants with mixup. Instead of having two independent ranks, one for Random Forest and the other for Rotation Forest, as with the two previous Figures (7 and 8), these tables show the ranks

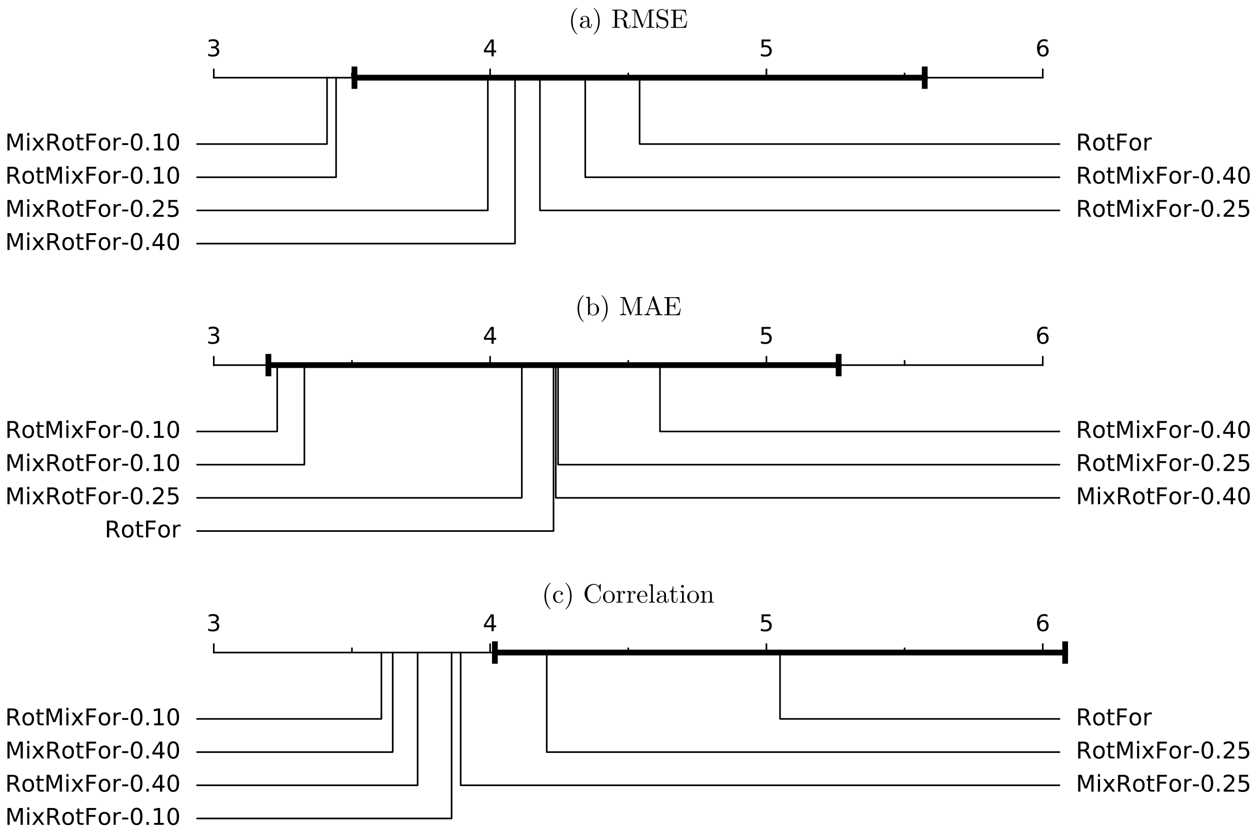


Figure 8: Comparison of Rotation Forest against variants with Mixup, with the Bonferroni-Dunn test. The marked interval spans the critical value and is centered at the mean rank for Rotation Forest. Variants with ranks outside the marked interval are significantly different ($p < 0.05$) from Rotation Forest.

Table 7: Average ranks.

RMSE		MAE		Correlation	
Method	Rank	Method	Rank	Method	Rank
RotMixFor-0.10	5.655738	RotMixFor-0.10	5.581967	RotMixFor-0.10	6.016393
MixRotFor-0.10	5.786885	MixRotFor-0.10	5.827869	RotMixFor-0.40	6.114754
MixRotFor-0.25	6.467213	RotMixFor-0.25	6.762295	MixRotFor-0.40	6.188525
RotMixFor-0.25	6.508197	MixRotFor-0.25	6.811475	MixRotFor-0.10	6.418033
MixRotFor-0.40	6.598361	MixRotFor-0.40	6.950820	MixRotFor-0.25	6.549180
RotMixFor-0.40	6.770492	RotFor	7.106557	RotMixFor-0.25	6.795082
RotFor	7.016393	RotMixFor-0.40	7.163934	MixRandFor-0.40	7.778689
MixRandFor-0.40	7.893443	MixRandFor-0.10	7.754098	RotFor	7.852459
MixRandFor-0.25	8.196721	MixRandFor-0.25	7.852459	RandMixFor-0.40	7.983607
MixRandFor-0.10	8.221311	RandMixFor-0.10	7.885246	RandMixFor-0.10	8.155738
RandMixFor-0.10	8.418033	MixRandFor-0.40	8.098361	RandMixFor-0.25	8.213115
RandMixFor-0.25	8.893443	RandFor	8.540984	MixRandFor-0.10	8.491803
RandMixFor-0.40	9.040984	RandMixFor-0.25	9.311475	MixRandFor-0.25	8.491803
RandFor	9.532787	RandMixFor-0.40	9.352459	RandFor	9.950820

for all the methods together. With regard to RMSE, all the Rotation Forest variants are above all the Random Forest variants. Moreover, the two original methods (without mixup) are the last methods in their respective sets. Likewise, with regard to MAE, the Rotation Forest variants are above all the Random Forest variants, although there are a few variants with mixup below the method without mixup. The methods without mixup for correlation are below all the other methods in their set, although there is some overlap between the two sets, because RandMixFor-0.40 is above RotFor.

Figures 9 and 10 show boxplots for the ranks of the different datasets. Both the Random Forest and the Rotation Forest variants are independently depicted in Figure 9, so the rank values range from 1 to 7. The Random Forest and the Rotation Forest variants are jointly depicted in Figure 10, so the rank values range from 1 to 14. These figures support the idea that the use of mixup variants is advisable.

Overall, Rotation Forest shows better performance compared to Random Forest, and mixup offers an advantage for both ensemble methods, which has been empirically demonstrated in our experiment.

Figure 9: Boxplots for the ranks. The boxplots to the left refer to the Random Forest variants and those to the right refer to the Rotation Forest variants.

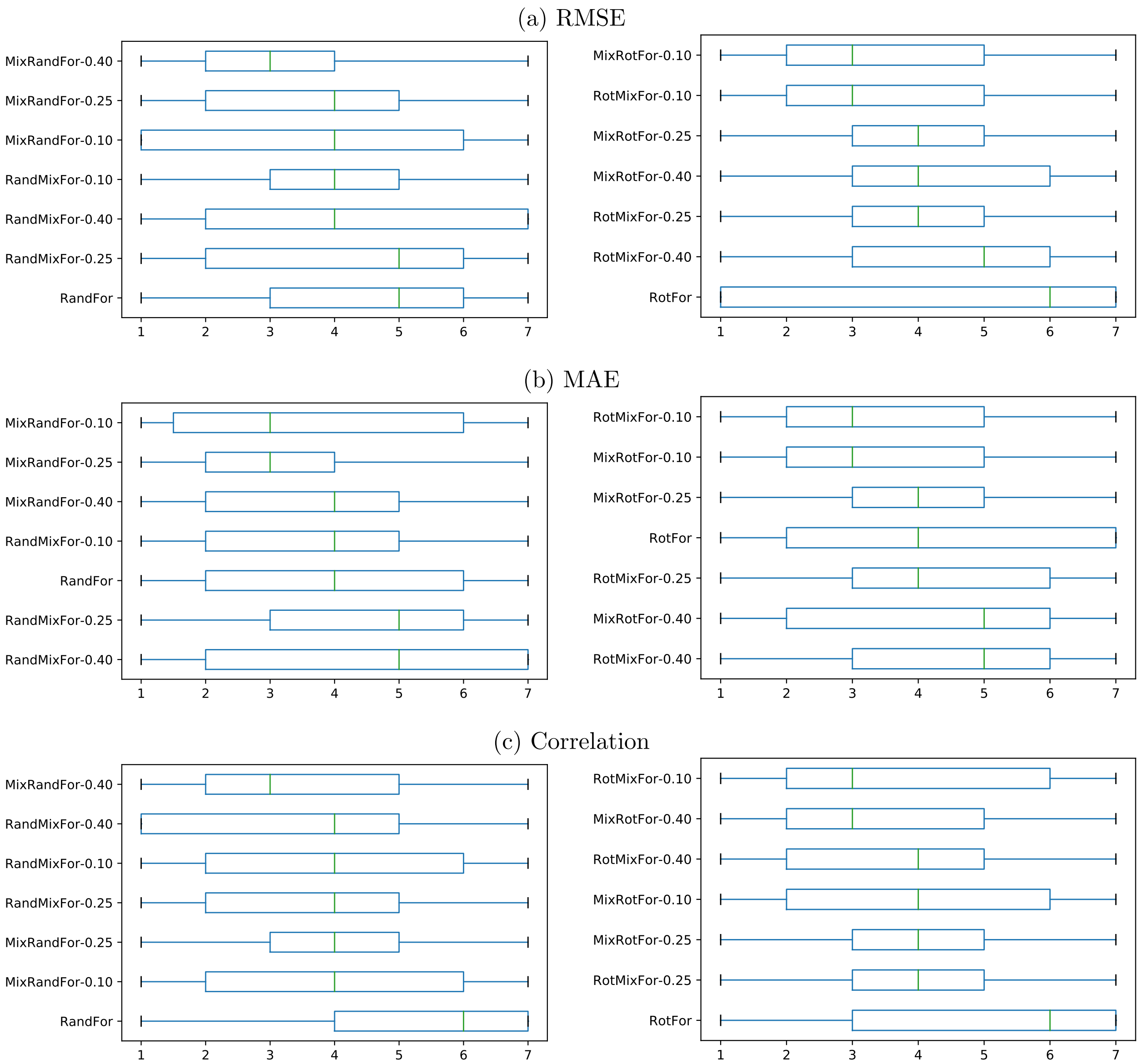
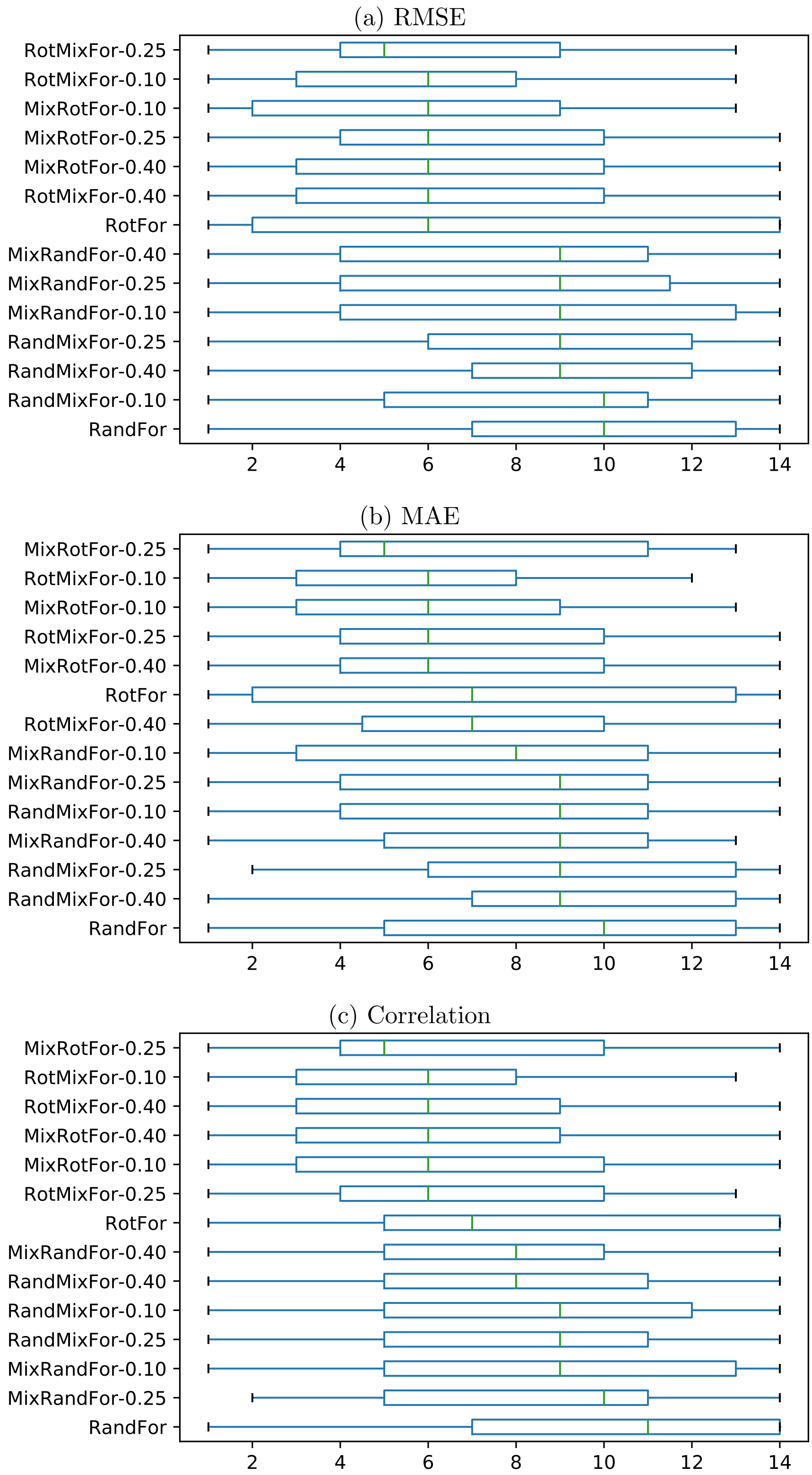


Figure 10: Boxplots for the ranks. The ranks are obtained using both Random and Rotation Forests variants.



Limitations. The scope of this study is nevertheless limited. The two parameters of the method, the α value for the Beta distribution, and the number of synthetic examples that are generated were not adjusted for each dataset. Only three values of α were considered and the number of synthetic examples was arbitrarily fixed at 50%. Ensemble size is another parameter that can affect the results and that can interact with the previous parameters. Moreover, the default parameter's values for Random Forest and Rotation Forest were used with no previous adjustment for the study.

The mixup approach has been applied to only two ensemble methods, Random Forest and Rotation Forest, although it could be applied to other methods. For instance, another very successful ensemble method, although not commonly used for regression, is boosting (Solomatine & Shrestha, 2004). The mixup approach can also be used with ensembles by combining other regression methods rather than classification trees. The usefulness of the mixup approach for regression ensembles with other ensembles and base methods is as yet unproven.

The mixup method was the only method considered for generating artificial instances. Other methods for generating artificial instances might be better suited for a given dataset.

5. Conclusions and future research

The mixup strategy has been previously used for regularizing deep neural networks, although this method can also be used for increasing diversity in ensembles. In this paper, we have shown that the performance of regression forest methods can be improved by using the mixup strategy, which introduces artificial instances in the datasets used for training each regression tree. The advantages of the mixup method have been experimentally shown for both Random Forest and Rotation Forest over a broad set of 61 datasets. Our experimental results favored the Rotation Forest and its improved variants.

Some limitations of the study can be approached in future works. The mixup method has one parameter, α. We found no clear pattern of influence for the three experimental values (0.1, 0.25, and 0.4). Adjusting α for each dataset and varying the number of generated artificial instances can both potentially improve the results.

Mixup forest can be applied to other ensemble methods, such as boosting variants. It can also be used with ensembles formed by other regression models instead of trees.

A future research line is the adaptation of the mixup method for classification datasets. As mentioned earlier, the use of mixup for regression is straightforward, because the output value is continuous. Nevertheless, the application of this method to classification requires a previous decision on the best way of combining different nominal classes. The method could also be useful in problems with several outputs, such as muti-label classification and multi-target regression.

The distribution of the instances can make the mixup strategy counterproductive, because it may add noise in a localized region of the space. With this in mind, further research on the convexity of the space could help clarify the advisability of applying mixup. Moreover, more advanced data augmentation techniques that take into account the manifold of the actual instances would be interesting to explore (Guo et al., 2018; Verma et al., 2018).

Recently, imbalance for regression has been studied (Torgo et al., 2013). The evaluation of whether mixup can be used to work with imbalanced datasets is also a promising area for future research.

Acknowledgments

This work was supported through project TIN2015-67534-P (MINECO/FEDER, UE) of the *Ministerio de Economía y Competitividad* of the Spanish Government, project BU085P17 (JCyL/FEDER, UE) of the *Junta de Castilla y León* (both projects co-financed through European Union FEDER funds), and by the *Consejería de Educación* of the *Junta de Castilla y León* and the European Social Fund through a pre-doctoral grant (EDU/1100/2017). The second author is grateful for a Mobility Grant from the *Universidad de Burgos*. The third author is grateful for a Mobility Grant (CAS19/00100) from the *Ministerio de Ciencia, Innovación y Universidades* of the Spanish Government. The authors gratefully acknowledge the support of NVIDIA Corporation and its donation of the TITAN Xp GPUs that facilitated this research.

References

Bagnall, A., Bostrom, A., Cawley, G., Flynn, M., Large, J., & Lines, J. (2018). Is rotation forest the best classifier for problems with continuous features? *arXiv preprint arXiv:1809.06705*, .

Beckham, C., Honari, S., Lamb, A., Verma, V., Ghadiri, F., Devon Hjelm, R., & Pal, C. (2019). Adversarial mixup resynthesizers. *arXiv e-prints*, .

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chawla, N., Lazarevic, A., Hall, L., & Bowyer, K. (2003). SMOTEBoost: Improving prediction of the minority class in boosting. In *7th European Conference on Principles and Practice of Knowledge Discovery in Databases(PKDD 2003)* (pp. 107–119).

Chen, Z., Chen, W., & Shi, Y. (2020). Ensemble learning with label proportions for bankruptcy prediction. *Expert Systems with Applications*, *146*, 113155. doi:10.1016/j.eswa.2019.113155.

Choi, H., Son, H., & Kim, C. (2018). Predicting financial distress of contractors in the construction industry using ensemble learning. *Expert Systems with Applications*, *110*, 1 – 10. doi:10.1016/j.eswa.2018.05.026.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, *7*, 1–30.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, *15*, 3133–3181. URL: http://jmlr.org/papers/v15/delgado14a.html.

Frank, E., & Pfahringer, B. (2006). Improving on bagging with input smearing. In W.-K. Ng, M. Kitsuregawa, J. Li, & K. Chang (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 97–106). Berlin, Heidelberg: Springer Berlin Heidelberg.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *42*, 463–484. doi:10.1109/TSMCC.2011.2161285.

García-Pedrajas, N., Pérez-Rodríguez, J., García-Pedrajas, M., Ortiz-Boyer, D., & Fyfe, C. (2012). Class imbalance methods for translation initiation site recognition in DNA sequences. *Knowledge-Based Systems*, *25*, 22–34. doi:10.1016/j.knosys.2011.05.002.

Geng, G. G., Wang, C. H., Li, Q. D., Xu, L., & Jin, X. B. (2007). Boosting the performance of web spam detection with ensemble under-sampling classification. In *Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007* (pp. 583–587). volume 4. doi:10.1109/FSKD.

2007.207.

Gónzalez, S., García, S., Lázaro, M., Figueiras-Vidal, A. R., & Herrera, F. (2017). Class switching according to nearest enemy distance for learning from highly imbalanced data-sets. *Pattern Recognition*, *70*, 12–24.

Guo, H., Mao, Y., & Zhang, R. (2018). Mixup as locally linear out-of-manifold regularization. *CoRR*, *abs/1809.02499*. URL: http://arxiv.org/abs/1809.02499.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220 – 239. doi:10.1016/j.eswa.2016.12.035.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, *11*, 10–18. doi:10.1145/1656274.1656278.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning. In D.-S. Huang, X.-P. Zhang, & G.-B. Huang (Eds.), *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I* (pp. 878–887). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/11538059_91.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (pp. 1322–1328). doi:10.1109/IJCNN.2008.4633969.

Inoue, H. (2018). Data augmentation by pairing samples for images classification. *CoRR*, *abs/1801.02929*. URL: http://arxiv.org/abs/1801.02929.

Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, *51*, 181–207. doi:10.1023/A:1022859003006.

Lindenbaum, O., Stanley, J., Wolf, G., & Krishnaswamy, S. (2018). Geometry based data generation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 31* (pp. 1400–1411). Curran Associates, Inc. URL: http://papers.nips.cc/paper/7414-geometry-based-data-generation.pdf.

Marqués, A., García, V., & Sánchez, J. (2012). Two-level classifier ensembles for credit risk assessment. *Expert Systems with Applications*, *39*, 10916 – 10922. doi:10.1016/j.eswa.2012.03.033.

Martínez-Muñoz, G., & Suárez, A. (2005). Switching class labels to generate classification ensembles. *Pattern Recognition*, *38*, 1483–1494.

Mayo, M., & Frank, E. (2017). Improving naive bayes for regression with optimised artificial surrogate data. *CoRR*, *abs/1707.04943*. URL: http://arxiv.org/abs/1707.04943.

Melville, P., & Mooney, R. J. (2003). Constructing diverse classifier ensembles using artificial training examples. In *IJCAI* (pp. 505–510). volume 3.

Melville, P., & Mooney, R. J. (2005). Creating diversity in ensembles using artificial data. *Information Fusion*, *6*, 99–111.

Menardi, G., & Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, *28*, 92–122. doi:10.1007/s10618-012-0295-5.

Mendes-Moreira, J., Soares, C., Jorge, A. M., & Sousa, J. F. D. (2012). Ensemble approaches for regression: A survey. *ACM computing surveys*, *45*, 10.

Panigrahi, S., Kundu, A., Sural, S., & Majumdar, A. K. (2009). Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning. *Information Fusion*, *10*, 354–363. doi:10.1016/j.inffus.2008.04.001.

Pardo, C., Diez-Pastor, J. F., García-Osorio, C., & Rodríguez, J. J. (2013). Rotation forests for regression. *Applied Mathematics and Computation*, *219*, 9914–9924.

Rodríguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*, 1619–1630. URL: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2006.211.

Ross, A., & Jain, A. (2003). Information fusion in biometrics. *Pattern Recognition Letters*, *24*, 2115 – 2125. doi:10.1016/S0167-8655(03)00079-5.

Sirlantzis, K., Hoque, S., & Fairhurst, M. (2008). Diversity in multiple classifier ensembles based on binary feature quantisation with application to face recognition. *Applied Soft Computing*, *8*, 437 – 445. doi:10.1016/j.asoc.2005.08.002.

Soares, S. G., & Araújo, R. (2015). A dynamic and on-line ensemble regression for changing environments. *Expert Systems with Applications*, *42*, 2935 – 2948. doi:10.1016/j.eswa.2014.11.053.

Solomatine, D. P., & Shrestha, D. L. (2004). Adaboost.RT: a boosting algorithm for regression problems. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)* (pp. 1163–1168 vol.2). volume 2. doi:10.1109/IJCNN.2004.1380102.

Summers, C., & Dinneen, M. J. (2019). Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1262–1270). IEEE.

Tay, W.-L., Chui, C.-K., Ong, S.-H., & Ng, A. C.-M. (2013). Ensemble-based regression analysis of multimodal medical data for osteopenia diagnosis. *Expert Systems with Applications*, *40*, 811 – 819. doi:10.1016/j.eswa.2012.08.031.

Tokozume, Y., Ushiku, Y., & Harada, T. (2017). Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282*, .

Tokozume, Y., Ushiku, Y., & Harada, T. (2018). Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5486–5494).

Torgo, L., Ribeiro, R. P., Pfahringer, B., & Branco, P. (2013). Smote for regression. In L. Correia, L. P. Reis, & J. Cascalho (Eds.), *Progress in Artificial Intelligence* (pp. 378–389). Berlin, Heidelberg: Springer Berlin Heidelberg.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., Lopez-Paz, D., & Bengio, Y. (2018). Manifold mixup: Better representations by interpolating hidden states. *arXiv preprint arXiv:1806.05236*, .

Wang, S., & Yao, X. (2009). Diversity analysis on imbalanced data sets by using ensemble models. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on* (pp. 324–331). IEEE.

Weng, B., Lu, L., Wang, X., Megahed, F. M., & Martinez, W. (2018). Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications*, *112*, 258 – 273. doi:10.1016/j.eswa.2018.06.016.

Zhang, H., Cissé, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *CoRR*, *abs/1710.09412*. URL: http://arxiv.org/abs/1710.09412v2.

Zhu, T., Lin, Y., & Liu, Y. (2017). Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognition*, *72*, 327 – 340. URL: http://www.sciencedirect.com/science/article/pii/S0031320317302947. doi:10.1016/j.patcog.2017.07.024.