Data Science Lab 2 Report

Donia Gharbi

Eötvös Loránd University, Faculty of informatics, Budapest, Hungary
qnghnc@inf.elte.hu

Abstract. To overcome the challenges faced with imbalanced datasets various methods can be used. such as Re-sampling which is a frequently used approach for imbalance learning. In the original paper, sample concatenation was used to conquer this issue. The approach proposed is concatenating two samples with the same labels into one sample. To avoid loss of valuable information and aggravate interclass overlap this technique uses the whole dataset which is not the case in most widely used methods. This study, a replication of the originally proposed method is going to be developed

Acknowledgments: This study draws replication from materials by Shi Hongbo, Zhang Ying, Chen Yuwen, Ji Suqin, and Dong Yuanxiang. Contributors in the study Resampling algorithms based on sample concatenation for imbalance learning.

Keywords: imbalanced learning · Re sampling · Concatenation

1 Introduction

An example of a classification issue where the distribution of instances among the recognized classes is biased or distorted is an imbalanced classification problem. When there is one example in the minority class and multiple ones in the majority class, the distribution can vary from slightly skewed to severely imbalanced. Various methods can be used to avoid this problem. Such as re-sampling, a commonly used imbalanced learning method. In most cases for re-sampling a sample from the original dataset is used which might cause a loss of valuable information and aggravate inter-class overlap. The method proposed in the study is Re-sampling algorithms based on sample concatenation for imbalance learning a concatenation approach is proposed in which the whole dataset is used. The contributions of the original study are the following:

– A re-sampling algorithm based on sample concatenation is proposed. It transforms an imbalanced dataset in an original sample space into a concatenated dataset in a new sample space. This will result in the reduction of the overlapping regions between classes having approximately the same sample sizes from different classes in the concatenated dataset.

– An ensemble re-sampling algorithm based on sample concatenation for imbalanced data is proposed: it transforms an original imbalanced dataset into multiple balanced datasets in the new sample space, Thereby reducing the problem of information loss and obtaining a classifier with better generalization ability.

– The classification difficulties of original datasets and concatenated datasets are measured by data complexity, and the effectiveness of the proposed algorithms is verified via experiments on the datasets from the UCI and KEEL dataset repositories.

2 Related work

Sample concatenation connects two samples end to end into a new concatenated sample.Its corresponding concatenated dataset obtained by sample concatenation is different from the original dataset in terms of sample dimensionality, sample size, and data distribution imbalance ance ratio of its corresponding concatenated dataset will be decreased if a suitable concatenating method is adopted, which is what the original paper Resampling algorithms based on sample concatenation for imbalance learning is suggested. The study claims that by using the model Re-sampling algorithms based on sample concatenation could help:

– Reduce the proportion of the inter-class overlapping region to the non-overlapping region of the concatenated datasets.
– Have a clearer classification boundary between two classes in the concatenated datasets.
– decrease the imbalance ratio between classes in the concatenated datasets.

in this paper a replication of the proposed approach will be developed.

3 Model Developement

3.1 Sample concatenation method

The aim of the Sample concatenation method is to decrease the imbalance ratio and inter-class overlapping ratio of datasets by transforming samples from the original sample space into a new new sample space by concatenating samples in pairs with the same class label. In the following section , the sample concatenation for the training data and test data will be explained

Sample concatenation method for training data Considering a two-class imbalanced training dataset, The minority class is labeled as "0" and denoted as **P** and the majority class is labeled as "1" and denoted as **N** . With the dataset being an imbalanced one : $|P|$ is significantly less than $|N|$. The new concatenated dataset will be a result of the concatenation of both the minority class and majority class samples. To get the newly constructed datatset the following mothods will be used

- **For the minority class** : generated by concatenating all pairs of samples in the monority class sample

- **For the majority class**: generated by concatenating each sample in the majority class with a sample subset noted as **set N** selected from the set of N

So each minority sample is concatenated with all minority samples, whereas each majority sample is concatenated with part of the majority samples which is the **set N**. In the following part the methods to select the subset **set N** from **N** and determining the number of its samples is going to be explained.

set N *selection method* In the method developed in the original paper, a weighted random sampling method to select a subset **set N** of the majority class. In this method, The stronger the class representation the greater the weight value and vice versa.The probability of that each sample is selected is determined by its weight , so that the ones that are more representative will most likely be selected. There is also a slight chance that some unrepresentative or particular samples will be chosen. To calculate the weight for each instance in the majority class the following steps need to be followed:

1. Find its k nearest neighbors
2. Calculate the weight of the instance in the following manner : if the neighboring point belongs to the majority class we assign the weight 1 to it and if it belongs to the minority class we assign 0 to it.After calculating the weight for each neighboring point individually, we calculate their average as the wight for the instance itself
3. Select the **set N** representative points from the majority class based on the weight

The samples with larger weight values have a greater chance of becoming members of **set N**

set N *size definition method* The sample size of **set N** has a significant effect on the imbalance ratio of the concatenated dataset.

The equation has two major parts:

- **n1:** Both of the two classes in a concatenated dataset are expected to have the same sample size it will be represented in as follows:
  - N : Number of instances in the majority class

$$n_1 = \frac{|\mathcal{P}| * |\mathcal{P}|}{|\mathcal{N}|}$$

Fig. 1. n1 Equation

- P : Number of instances in the minority class

– **n2:** in order to retain more information in the majority class, the sample size should be determined by means of methods in statistics. Specifically, the number of samples chosen from the majority class N should be:

$$n_2 = \frac{|\mathcal{N}| Z^2_{\alpha/2} \sigma^2}{|\mathcal{N}| \varepsilon^2 + Z^2_{\alpha/2} \sigma^2}$$

Fig. 2. n2 Equation

- N : Number of instances in the majority class
- δ : is the standard deviation of the majority samples
- ϵ : is the acceptable tolerance error that can be adjusted as required
- Z^2 α /2 : is the critical value of the Z test at the significance level α

The aim of this study is not necessary to make different classes in a dataset have exactly the same sample size , It is enough to generate an approximately balanced concatenated dataset. we can restrict the imbalance ratio of a concatenated dataset to be smaller than a certain threshold M, for example, M=1.5. Specifically, suppose [1, M] is the range of Imbalance ratio, and Pr = n2/n1 is the ratio of the sample size.

The sample size can be then conducted based on the following cases :

– if $1 \leq \text{Pr} \leq \text{M}$: the size should be equal to the result of equation n2
– if $\text{Pr} > \text{M}$: the size should be equal to the result of equation n1/*M
– if $\text{Pr} < 1$: the size should be equal to the result of equation n1

Sample concatenation method for test data After concatenating the training samples, we need to transform the training set into the sample space of the concatenated training data to predict its class label by using the classifier. However,we cannot use the same sample concatenation method for training samples cannot be used for test samples because of we don't have another sample to concatenate the test sample with; hence, a sample concatenation method for test samples needs to be designed, which consists of simply concatenating the same instance with itself to fit the same format as the training set.

Re-sampling algorithms based on sample concatenation As previously mentioned, In the original study there were two main algorithms for re-sampling the data and it works as follows:

- The training dataset is split into minority subset P and majority subset N
- Calculate the weight for each instance in the majority class N
- Calculate the number of samples in Set N
- Generate the concatenated set for the majority class: each instance from the majority class N with all the points from set N.
- Generate the concatenated set for the Minority class: each instance from the minority class P with all the points from the minority class P.
- Return the concatenated dataset that contains both the newly generated minority and majority class.

Algorithm 1 Resampling algorithm based on sample concatenation

Input: a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, n, \boldsymbol{x}_i \in \mathcal{R}^d, y_i \in \{+1, -1\}\}$
Output: a concatenated dataset
Procedure:

Let $\mathcal{P} = \{(\boldsymbol{x}_i, y_i) | (\boldsymbol{x}_i, y_i) \in \mathcal{D}, y_i = +1\}$, $\mathcal{N} = \{(\boldsymbol{x}_i, y_i) | (\boldsymbol{x}_i, y_i) \in \mathcal{D}, y_i = -1\}$
Let $\mathcal{P}_C = \emptyset$, $\mathcal{N}_C = \emptyset$
Calculate the weight of each sample in $\mathcal{N}$
Calculate the number of samples in $\boldsymbol{Set}_{\mathcal{N}}$
Obtain the majority subset $\boldsymbol{Set}_{\mathcal{N}}$ by random sampling from the weighted majority samples
for $i = 1$ to $|\mathcal{N}|$ **do**
  for $j = 1$ to $|\boldsymbol{Set}_{\mathcal{N}}|$ **do**
    Concatenate $\boldsymbol{x}_i$ in $\mathcal{N}$ with $\boldsymbol{x}_j$ in $\boldsymbol{Set}_{\mathcal{N}}$ to get a new sample $(\widehat{\boldsymbol{x}_i \boldsymbol{x}_j}, -1)$
    Add $(\widehat{\boldsymbol{x}_i \boldsymbol{x}_j}, -1)$ into $\mathcal{N}_C$
  end for
end for
for $i = 1$ to $|\mathcal{P}|$ **do**
  for $j = 1$ to $|\mathcal{P}|$ **do**
    Concatenate two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in $\mathcal{P}$ to get a new sample $(\widehat{\boldsymbol{x}_i \boldsymbol{x}_j}, +1)$
    Add $(\widehat{\boldsymbol{x}_i \boldsymbol{x}_j}, +1)$ into $\mathcal{P}_C$
  end for
end for
Return a concatenated dataset $\mathcal{D}_C = \mathcal{N}_C \bigcup \mathcal{P}_C$

Fig. 3. Re-sampling algorithms based on sample concatenation

An ensemble re-sampling algorithm based on sample concatenation

An ensemble re-sampling algorithm based on sample concatenation Ensemble learning refers to a group of base learners, or models, which work collectively to achieve a better final prediction. A single model, also known as a base or weak learner, may not perform well individually due to high variance or high bias. However, when weak learners are aggregated, they can form a strong learner, as their combination reduces bias or variance, yielding better model performance.

In this function to make sure that our classifier is not biased and to achieve a classifier with stronger generalization performance the ensemble method was suggested: The original dataset was split into T subsets. The classifier is trained by T-1 subsets and tested by 1. We make sure to iterate all the sub-datasets as training sets and save the results on average to test the performance.

4 Experiments and result analysis

4.1 Experimental data

To verify the effectiveness of the proposed algorithms, In the original paper 28 datasets were selected from the UCI machine learning repository. Frome which some of them have two classes, while others have multiple classes. As the focus is on two-class classification problems, multi-class datasets are converted into two-class imbalanced datasets by merging some classes. To test the performance further 18 other datasets were added from the same source which gives us the result of 46 datasets. More details about the chosen datasets used in the experiments are portrayed in the table in the following figure:

	Dataset	original count	original Majority	original Minority	Imbalance ratio original
0	abalone9-18.csv	584	548	36	15.222222
1	Breast.csv	455	282	173	1.630058
2	ecoli-0-1_vs_2-3-5.csv	196	176	19	9.263158
3	ecoli-0-1_vs_5.csv	192	177	15	11.800000
4	ecoli-0-1-4-7_vs_5-6.csv	265	249	16	15.562500
5	ecoli-0-2-3-4_vs_5.csv	161	146	15	9.733333
6	ecoli-0-4-6_vs_5.csv	162	147	15	9.800000
7	ecoli-0-6-7_vs_5.csv	176	157	19	8.263158
8	ecoli2.csv	268	227	41	5.536585
9	ecoli3.csv	268	240	28	8.571429
10	glass0123vs456.csv	171	130	41	3.170732
11	glass0.csv	171	121	50	2.420000
12	glass1.csv	171	104	67	1.552239
13	glass6.csv	171	148	23	6.434783
14	haberman.csv	244	182	62	2.935484
15	iris1.csv	120	79	41	1.926829
16	leaf.csv	272	215	57	3.771930
17	new-thyroid1.csv	172	142	30	4.733333
18	new-thyroid2.csv	172	144	28	5.142857
19	parkinsons.csv	156	122	34	3.588235
20	seeds.csv	168	111	57	1.947368
21	spect.csv	213	173	40	4.325000
22	wpbc.csv	455	282	173	1.630058
23	yeast-1_vs_7.csv	367	342	25	13.680000
24	yeast-2_vs_4.csv	411	370	41	9.024390
25	abalone11--17.csv	436	391	45	8.688889
26	abalone4--8.csv	500	456	44	10.363636
27	abalone5--10.csv	599	511	88	5.806818
28	BreastCancerWisconsin.csv	455	282	173	1.630058
29	ecoli1.csv	268	207	61	3.393443
30	eligibility-loan.csv	491	331	160	2.068750
31	iris0.csv	120	79	41	1.926829
32	lung-cancer.csv	800	496	304	1.631579
33	Maternal-Risk-Imvsh.csv	811	589	222	2.653153
34	page-blocks1vs2345.csv	4378	3907	471	8.295117
35	page-blocks2vs4.csv	333	264	69	3.826087
36	page-blocks3vs5.csv	114	90	24	3.750000
37	pima-indians-diabetes.csv	614	410	204	2.009804
38	pima.csv	614	410	204	2.009804
39	wheat1.csv	168	117	51	2.294118
40	wilt.csv	3471	3408	63	54.095238
41	winequality-red-3456vs78.csv	1279	1103	176	6.267045
42	winequality-red-34vs56.csv	1105	1057	48	22.020833
43	wisconsin.csv	546	354	192	1.843750
44	yeast1.csv	1187	841	346	2.430636
45	yeast3.csv	1187	1060	127	8.346457

Fig. 4. Datasets used in the experiments

To use the ensemble method all of the datasets were split into k=10 sub-datasets from which in each iteration 9 were considered a training set and 1 a test set.

4.2 Experiments

Generated data comparison the first experiment was to generate the concatenated dataset using the Re-sampling algorithm. There is a significant improvement in the imbalance ratio when we compare the original datasets' results and the concatenated ones. In most cases (44) the new imbalance ratio is less than M which is threshold defined for the imbalance ratio that was set at 1.5. The following table showcases the a comparison between the characteristics of the original dataset and one of the iterations from the concatenated one:

	Dataset	original count	original Majority	original Minority	Imbalance ratio original	representitive points	con count	con Majority	con Minority	Imbalance ratio con
0	abalone9-18.csv	584	548	36	15.222222	3	2940	1644	1296	1.268519
1	Breast.csv	455	282	173	1.630058	107	60103	30174	29929	1.008186
2	ecoli-0-1_vs_2-3-5.csv	195	176	19	9.263158	3	889	528	361	1.462604
3	ecoli-0-1_vs_5.csv	192	177	15	11.800000	2	579	354	225	1.573333
4	ecoli-0-1-4-7_vs_5-6.csv	265	249	16	15.562500	2	754	498	256	1.945312
5	ecoli-0-2-3-4_vs_5.csv	161	146	15	9.733333	2	517	292	225	1.297778
6	ecoli-0-4-6_vs_5.csv	162	147	15	9.800000	2	519	294	225	1.306667
7	ecoli-0-6-7_vs_5.csv	176	157	19	8.263158	3	832	471	361	1.304709
8	ecoli2.csv	268	227	41	5.536585	8	3497	1816	1681	1.080309
9	ecoli3.csv	268	240	28	8.571429	4	1744	960	784	1.224490
10	glass0123vs456.csv	171	130	41	3.170732	13	3371	1690	1681	1.005354
11	glass0.csv	171	121	50	2.420000	21	5041	2541	2500	1.016400
12	glass1.csv	171	104	67	1.552239	44	9065	4576	4489	1.019381
13	glass6.csv	171	148	23	6.434783	4	1121	592	529	1.119093
14	haberman.csv	244	182	62	2.935484	22	7848	4004	3844	1.041623
15	iris1.csv	120	79	41	1.926829	22	3419	1738	1681	1.033908
16	leaf.csv	272	215	57	3.771930	16	6689	3440	3249	1.058787
17	new-thyroid1.csv	172	142	30	4.733333	7	1894	994	900	1.104444
18	new-thyroid2.csv	172	144	28	5.142857	6	1648	864	784	1.102041
19	parkinsons.csv	156	122	34	3.588235	10	2376	1220	1156	1.055363
20	seeds.csv	168	111	57	1.947368	30	6579	3330	3249	1.024931
21	spect.csv	213	173	40	4.325000	10	3330	1730	1600	1.081250
22	wpbc.csv	455	282	173	1.630058	107	60103	30174	29929	1.008186
23	yeast-1_vs_7.csv	367	342	25	13.680000	2	1309	684	625	1.094400
24	yeast-2_vs_4.csv	411	370	41	9.024390	5	3531	1850	1681	1.100535
25	abalone11-17.csv	436	391	45	8.688889	6	4371	2346	2025	1.158519
26	abalone4-8.csv	500	456	44	10.363636	5	4216	2280	1936	1.177686
27	abalone5-10.csv	599	511	88	5.806818	16	15920	8176	7744	1.055785
28	BreastCancerWisconsin.csv	455	282	173	1.630058	107	60103	30174	29929	1.008186
29	ecoli1.csv	268	207	61	3.393443	18	7447	3726	3721	1.001344
30	eligibility-loan.csv	491	331	160	2.068750	78	51418	25818	25600	1.008516
31	iris0.csv	120	79	41	1.926829	22	3419	1738	1681	1.033908
32	lung-cancer.csv	800	496	304	1.631579	187	185168	92752	92416	1.003636
33	Maternal-Risk-Imvsh.csv	811	589	222	2.653153	84	98760	49476	49284	1.003896
34	page-blocks1vs2345.csv	4378	3907	471	8.295117	57	444540	222699	221841	1.003868
35	page-blocks2vs4.csv	333	264	69	3.826087	19	9777	5016	4761	1.053560
36	page-blocks3vs5.csv	114	90	24	3.750000	7	1206	630	576	1.093750
37	pima-indians-diabetes.csv	614	410	204	2.009804	102	83436	41820	41616	1.004902
38	pima.csv	614	410	204	2.009804	102	83436	41820	41616	1.004902
39	wheat1.csv	168	117	51	2.294118	23	5292	2691	2601	1.034602
40	wilt.csv	3471	3408	63	54.095238	2	10785	6816	3969	1.717309
41	winequality-red-3456vs78.csv	1279	1103	176	6.267045	29	62963	31967	30976	1.032038
42	winequality-red-34vs56.csv	1105	1057	48	22.020833	3	5475	3171	2304	1.376302
43	wisconsin.csv	546	354	192	1.843750	105	74034	37170	36864	1.008301
44	yeast1.csv	1187	841	346	2.430636	143	239979	120263	119716	1.004569
45	yeast3.csv	1187	1060	127	8.346457	16	33089	16960	16129	1.051522

Fig. 5. Characteristics of the original dataset and concatenated dataset comparison

Classification results comparison To test the concatinated dataset , classification was performed on the original dataset and the concatenated one.The models used in these experiments are Linear Discriminant Analysis and Random Forest Classifier. and to measure the performance of both of them the following scores were used:

- Precision : quantifies the number of positive class predictions that actually belong to the positive class.
- Recall : quantifies the number of positive class predictions made out of all positive examples in the dataset.
- F-Measure : provides a single score that balances both the concerns of precision and recall in one number
- Accuracy : Calculates the percentage of labels that our model successfully predicted is represented by accuracy.
- AUC : Represents the probability that a random positive example is positioned to the right of a random negative example.

Also to compare how well the model performs in general for each dataset we checked for each score mentioned above which one performed better. The better performing dataset gives either the concatenated sum +1 or the generated one +1. and based on the results we can compare the overall performance.

Classification using Linear Discriminant Analysis In the first experiment Linear Discriminant Analysis classification model was used. To compare the performance in both cases the previously mentioned scores were used : Precision , Recall , F-Measure and Accuracy and AUC. In most cases the methods developed has better results than the original datasets. A sample of the results obtained can be shown in the following figure:

	Dataset	Version	Accuracy score %	F1 Score %	precision score %	recall Score %	AUC Score %	tn	fp	fn	tp
42	spect.csv	original	49.120	51.770	51.700	59.260	49.120	1	9	2	15
43	spect.csv	concatinated	94.331	91.595	94.299	90.923	94.331	5.5	0.0	2.4	18.6
44	wpbc.csv	original	96.670	98.230	98.290	98.250	96.670	42	0	1	14
45	wpbc.csv	concatinated	100.000	100.000	100.000	100.000	100.000	35.7	0.0	0.0	21.2
46	yeast-1_vs_7.csv	original	50.000	90.330	87.580	93.480	50.000	43	0	3	0
47	yeast-1_vs_7.csv	concatinated	80.540	78.612	94.009	72.320	80.540	30.5	12.4	0.3	2.7
48	yeast-2_vs_4.csv	original	60.000	89.900	92.910	92.310	60.000	47	0	4	1
49	yeast-2_vs_4.csv	concatinated	85.705	78.577	93.954	74.214	85.705	33.1	13.2	0.0	5.1
50	abalone11–17.csv	original	56.290	85.340	84.160	87.270	56.290	47	2	5	1
51	abalone11–17.csv	concatinated	86.846	81.785	92.765	77.793	86.846	36.7	12.0	0.1	5.7
52	abalone4–8.csv	original	99.150	98.490	98.730	98.410	99.150	4	0	1	58
53	abalone4–8.csv	concatinated	85.114	78.279	93.442	72.921	85.114	5.7	0.0	16.9	39.9
54	abalone5–10.csv	original	93.730	94.810	95.160	94.670	93.730	12	1	3	59
55	abalone5–10.csv	concatinated	97.447	96.514	97.380	96.248	97.447	11.4	0.1	2.7	60.7
56	BreastCancerWisconsin.csv	original	96.670	98.230	98.290	98.250	96.670	42	0	1	14
57	BreastCancerWisconsin.csv	concatinated	100.000	100.000	100.000	100.000	100.000	35.7	0.0	0.0	21.2
58	ecoli1.csv	original	94.440	97.000	97.170	97.060	94.440	25	0	1	8
59	ecoli1.csv	concatinated	97.437	96.856	97.540	96.720	97.437	24.9	1.0	0.1	7.6
60	eligibility-loan.csv	original	72.830	82.500	83.420	83.870	72.830	8	8	2	44
61	eligibility-loan.csv	concatinated	76.579	79.613	81.036	79.157	76.579	14.8	4.4	8.4	33.6

Fig. 6. Sample of LDA classification results using the original and the concatenated datasets

To compare the overall performance the following table showcases for each score how many datasets performed better for each of the concatenated and the original ones, we can see that based on the AUC ,Accuracy and Precision scores there are more cases in which the concatenated datasets perform better.

	Measure	Original	Concatinated
0	Accuracy	10	36
1	F1	25	21
2	Precision	14	32
3	Recall	26	20
4	AUC	10	36

Fig. 7. Scores performance comparison Using LDA classification

Classification using Random Forest Classifier In the second experiment Random Forest Classifier was used. The same table to compare the overall per-

	Dataset	Version	Accuracy score %	F1 Score %	precision score %	recall Score %	AUC Score %	tn	fp	fn	tp
0	abalone9-18.csv	original	50.000	95.960	94.670	97.300	50.000	72	0	2	0
1	abalone9-18.csv	concatinated	82.130	72.145	95.696	66.298	82.130	44.3	24.6	0.0	4.2
2	Breast.csv	original	90.950	92.980	92.980	92.980	90.950	13	2	2	40
3	Breast.csv	concatinated	100.000	100.000	100.000	100.000	100.000	21.2	0.0	0.0	35.7
4	ecoli-0-1_vs_2-3-5.csv	original	50.000	94.040	92.160	96.000	50.000	24	0	1	0
5	ecoli-0-1_vs_2-3-5.csv	concatinated	98.636	97.953	99.027	97.500	98.636	21.4	0.6	0.0	2.4
6	ecoli-0-1_vs_5.csv	original	97.730	96.200	97.220	95.830	97.730	21	1	0	2
7	ecoli-0-1_vs_5.csv	concatinated	98.408	97.544	98.666	97.084	98.408	21.3	0.7	0.0	2.0
8	ecoli-0-1-4-7_vs_5-6.csv	original	85.710	93.730	94.520	94.120	85.710	27	0	2	5
9	ecoli-0-1-4-7_vs_5-6.csv	concatinated	95.343	92.537	98.657	91.287	95.343	27.9	2.8	0.0	2.5
10	ecoli-0-2-3-4_vs_5.csv	original	100.000	100.000	100.000	100.000	100.000	18	0	0	3
11	ecoli-0-2-3-4_vs_5.csv	concatinated	99.166	98.680	99.167	98.500	99.166	17.9	0.3	0.0	2.0
12	ecoli-0-4-6_vs_5.csv	original	100.000	100.000	100.000	100.000	100.000	18	0	0	3
13	ecoli-0-4-6_vs_5.csv	concatinated	99.444	99.086	99.334	99.000	99.444	18.1	0.2	0.0	2.0
14	ecoli-0-6-7_vs_5.csv	original	100.000	100.000	100.000	100.000	100.000	21	0	0	1
15	ecoli-0-6-7_vs_5.csv	concatinated	98.250	97.435	99.015	96.818	98.250	19.3	0.7	0.0	2.0
16	ecoli2.csv	original	83.330	93.630	94.510	94.120	83.330	28	0	2	4
17	ecoli2.csv	concatinated	99.292	98.658	99.069	98.797	99.292	28.0	0.4	0.0	5.2
18	ecoli3.csv	original	96.880	95.000	97.060	94.120	96.880	30	2	0	2
19	ecoli3.csv	concatinated	97.333	95.912	97.780	95.179	97.333	28.5	1.6	0.0	3.5

Fig. 8. Sample of LDA classification results using the original and the concatenated datasets

formance was generated for the Random Forest Classifier, we can see that based on that based on all the scores : Precision , Recall , F-Measure and Accuracy and AUC there are more cases in which the concatenated datasets perform better.

	Measure	Original	Concatinated
0	Accuracy	10	36
1	F1	16	30
2	Precision	11	35
3	Recall	17	29
4	AUC	10	36

Fig. 9. Scores performance comparison Using LDA classification

When we compare the results for both classifiers based on the scores the Random Forest Classifier has a better performance than the Linear Discriminant Analysis model

5 Conclusion

In this study , a replication of from materials by Shi Hongbo, Zhang Ying, Chen Yuwen, Ji Suqin, and Dong Yuanxiang. Contributors in the study Resampling algorithms based on sample concatenation for imbalance learning which contains three main parts: The development of a re-sampling algorithm based on sample concatenation , An ensemble re-sampling algorithm based on sample concatenation for imbalanced data and testing the model developed. Due to lack of resources for this paper It was a little challenging to create the project from scratch especially that some parts were a little vague and there was nothing to compare your code to due the unavailability of the source code, nevertheless the model developed has shown a significantly better performance than the original datasets in most cases. In the future further exploration of the methods can be studied to improve the performance of imbalanced learning. Also making the developed model work on Multi-class datasets and not just binary-class ones could help analysing the performance.

References

1. esampling algorithms based on sample concatenation for imbalance learning by Shi Hongbo, Zhang Ying, Chen Yuwen, Ji Suqin, and Dong Yuanxiang : https://www.sciencedirect.com/science/article/abs/pii/S0950705122002659