Intelligence-Based Medicine 3-4 (2020) 100023

Contents lists available at ScienceDirect

Intelligence-Based Medicine

journal homepage: www.journals.elsevier.com/intelligence-based-medicine

Outlier-SMOTE: A refined oversampling technique for improved detection of COVID-19

Venkata Pavan Kumar Turlapati [a,*], Manas Ranjan Prusty [b,c]

[a] School of Computing, SRM Institute of Science and Technology, Kattankulathur, 603203, India
[b] Centre for Cyber Physical Systems, Vellore Institute of Technology, Chennai, 600127, India
[c] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600127, India

ARTICLE INFO

Keywords:
Imbalanced dataset
SMOTE
COVID-19
Over-sampling
Classification

ABSTRACT

Almost every dataset these days continually faces the predicament of class imbalance. It is difficult to train classifiers on these types of data as they become biased towards a set of classes, hence leading to reduction in classifier performance. This setback is often tackled by the use of various over-sampling or under-sampling algorithms. But, the method which stood out of all the numerous algorithms was the Synthetic Minority Oversampling Technique (SMOTE). SMOTE generates synthetic samples of the minority class by oversampling each data-point by considering linear combinations of existing minority class neighbors. Each minority data sample generates an equal number of synthetic data. As the world is suffering from the plight of COVID-19 pandemic, the authors applied the idea to help boost the classifying performance whilst detecting this deadly virus. This paper presents a modified version of SMOTE known as Outlier-SMOTE wherein each data-point is oversampled with respect to its distance from other data-points. The data-point which is farther than the other data-points is given greater importance and is oversampled more than its counterparts. Outlier-SMOTE reduces the chances of overlapping of minority data samples which often occurs in the traditional SMOTE algorithm. This method is tested on five benchmark datasets and is eventually tested on a COVID-19 dataset. F-measure, Recall and Precision are used as principle metrics to evaluate the performance of the classifier as is the case for any class imbalanced data set. The proposed algorithm performs considerably better than the traditional SMOTE algorithm for the considered datasets.

1. Introduction

As of July 22nd, 2020, over 15 million cases of COVID-19 were detected over the world. Schools postponed their examinations, offices were closed, many employees were laid-off, and a plethora of labourers were stranded. There may be just a countable number of people whom this pandemic has not affected since its onset. Also called the Novel Coronavirus, it has caused an average of 144 deaths per 1 million people. Although the death rate is on the lower side, the number of people hospitalized proliferated day by day. Hospitals could rarely accommodate so many people, and also faced a shortage of equipment. The authors salute the heroic efforts by the doctors [1] and the government to overcome this abysmal situation.

To combat this situation, the researchers present their contribution in improving the classification accuracy while dealing with highly imbalanced COVID-19 datasets. Since only 9% of the people tested positive for coronavirus, every dataset having a clinical history of patients is bound to be imbalanced. An imbalanced dataset is the dataset where elements of one class heavily outnumber the elements of the other [2]. While dealing with imbalanced datasets, this paper has considered two-class imbalances to ensure clarity [3,4]. These imbalanced datasets heavily affect the performance of the classifier [5]. The class imbalance problems have been noticed in a myriad of fields such as fraud detection [6], medicine [7], bioinformatics [8], intrusion detection [9], financial management [10], and event identification in nuclear plants [11] being a few of them. Therefore, the authors propose an improved oversampling method which will create new and near-accurate synthetic data based upon the existing data. This method is an adaptation of the Synthetic Minority Oversampling Technique (SMOTE) [12]. SMOTE creates synthetic samples of the minority class by calculating the euclidean distance between any two randomly chosen k-nearest neighbors [13] and introducing new synthetic samples along the line joining the two minority samples. Every

* Corresponding author.
E-mail address: pk1842000@gmail.com (V.P.K. Turlapati).

https://doi.org/10.1016/j.ibmed.2020.100023
Received 27 July 2020; Received in revised form 10 November 2020; Accepted 16 November 2020

data point is oversampled a certain amount, to produce more synthetic samples. For instance, 300% oversampling means every minority sample is oversampled thrice, meaning three synthetic samples are generated for every data point. The proposed algorithm in this paper called the Outlier-SMOTE states that every minority sample need not be oversampled by the same amount. Each data-point should be oversampled based upon its position in the space. Outlier-SMOTE gives more importance to the outlying/remotely placed samples, and oversamples them more as compared to its counterparts. This makes the job of the classifier easy to notice the miniscule features and classify them correctly. Since the motive is to apply the algorithm to a medical domain (example: COVID-19), the main goal should be to reduce the false negatives, which means outputting the result 'safe' for the person who is affected. The aftermath could be disastrous if something of the sort happens.

1.1. COVID-19 and Outlier-SMOTE

The detection of COVID-19 involves undergoing a 'nasopharyngeal swab', a 'throat swab' and a check through RT-PCR. A real-time polymerase chain reaction (real-time PCR), is a laboratory technique of molecular biology based on the polymerase chain reaction. Other than its function in detection of COVID-19, RT-PCR has been used to measure viral load with HIV and may also be used with other RNA viruses such as measles and mumps.

Plus, pulse oxygen saturation (also called SpO2) is an important testing method for COVID-19. It is a fraction of the oxygen-saturated haemoglobin, compared to the total haemoglobin in the blood. Also, Chest X-ray can be used as an effective and a fast way to immediately triage COVID-19 patients when suspected, but Chest X-Rays have been found to produce many false positives in the detection of COVID-19, due to which they may not be an effective method to gauge the chances of the disease. Due to the high volume of cases, RT-PCR, SpO2 (Pulse oxygen saturation) and chest X-Ray are infeasible in many locations due to the global shortage of test kits and resources. Many tests turn out to be negative as a lot of people having a mere cough or cold are tested for COVID-19. This leads to a massive drain of resources provided by the government and the people really in need of the test are left out. Outlier-SMOTE acts as a filter by improving the prediction of a person having the virus. The model is trained upon the data provided by Kaggle,[1] and predicts whether the person, given particular symptoms, is likely to be affected by COVID-19. This dataset contains anonymized data from patients seen at the Hospital Israelita Albert Einstein, at São Paulo, Brazil, and who had samples collected to perform the SARS-CoV-2 RT-PCR and additional laboratory tests during a visit to the hospital. Outlier-SMOTE algorithm helps in filtering out people who are less likely to be affected and in turn leads to the optimum use of resources available. Using this algorithm, tests will only be conducted on the people who are most probable of being affected with the virus.

Outlier-SMOTE can reduce the chances of false negatives, by a unique method of oversampling for each data-point. The Euclidean distance of each minority sample is taken and is compiled as a Matrix. L2 norms can be used for finding the oversampling weights of the samples in the minority class. The farthest element will have the maximum priority for oversampling. Then, the matrix is normalized and converted into an oversampling matrix with the formula given below to show how much each data-point will be oversampled. Upon experimentation with five benchmark datasets, and comparison with SMOTE [12] and ADASYN [14], this paper has proved that Outlier-SMOTE performs better in the majority of cases, when tested upon Recall, Precision and F1-Score. As stated in Fig. 1, the authors aim to maximize the performance of the algorithm in the pre-processing stage.

Further sections of this paper are elaborated as follows: Section 2 briefly summarizes the past work on imbalanced datasets. Section 3 provides us with the details of the Outlier-SMOTE algorithm and its working. Section 4 presents the various testing methods that were used to evaluate the algorithm's performance. Section 5 elucidates the results by describing the comparison of Outlier-SMOTE with SMOTE and ADASYN. and Section 6 explains the inferences that were obtained from the results. Section 7 elaborates the COVID-19 data and tests the algorithm in a similar way as that of the five considered benchmark datasets. Section 8 has been included to discuss the nuances and the scope of improvements for this algorithm. Finally, the authors conclude the paper in Section 9.

2. Literature review

The quandary of imbalanced datasets has been prevailing a long time, and several methods have been proposed to improve the classification accuracy, like random over-sampling or under-sampling [15], NearMiss [16], Borderline SMOTE [17], and many more [18,19]. SMOTE chooses a minority sample and randomly selects one of it's k-nearest neighbors, multiplies the distance of the line joining the two with any number between 0 and 1, and places it in that line. This process is carried out for all other feature data in the minority class until the N% oversampling limit is reached. While a modified version known as Safe-Level SMOTE [20], synthesizes minority instances which are at a safe level and assigns weights accordingly. According to this paper, the gap between the majority and minority samples should be as much as possible. Many researchers also used different classifiers to gauge their performance such as [21], which uses Bagging [22], and SVMs [23] to deal with imbalanced datasets. It's an extrapolated version of Borderline-SMOTE.

Borderline SMOTE [17] effectively over-samples only those minority elements which are at a safe distance from the border of the majority samples. Doing this prevents the risk of overlapping the samples which happens in most of the cases while oversampling. If the minority samples are at a safe distance, then oversampling will be safe and it also eases the work of the classifier in extracting the important features. Many hybrid approaches are also followed, for instance Ref. [24], implementing SMOTE in appropriate searching algorithms such as PSO (Particle Swarm Optimization [25]) and classifiers such as C5 (Decision Tree [26]) can significantly improve the effectiveness of classification for massive imbalanced data sets.

The algorithms used for comparison with Outlier-SMOTE in this paper are ADASYN [14] and SMOTE [12]. The essential idea of ADASYN is to use a weighted distribution for various minority class examples according to their own level of difficulty in learning, where a greater amount of synthetic data is generated for minority samples that are harder to learn. The logic of these algorithms, though being similar, have a minor difference, which is that ADASYN is an improved version of SMOTE. In SMOTE, all the synthetically generated samples have a linear correlation with the original samples, whereas in ADASYN, instead of being linearly correlated, the generated samples have a minutia of variance in them, which make them look analogous to the real samples. ADASYN, though being adaptive as compared to SMOTE, generates imprecise samples often as more data is generated in neighbourhoods with high amounts of majority class samples. Because of this, the synthetic data generated might be very similar to the majority class data, potentially generating many false positives. This drawback can be countered by putting a cap on the oversampling rate.

Also, there are many popular algorithms such as [27] SMOTEBoost combines an intelligent oversampling technique (SMOTE) with AdaBoost, resulting in a highly effective hybrid approach to learning from imbalanced data. The authors say, combining RUS and Boost will give a much improved performance as compared to SMOTE. Also, there is SCUT [28], which is used to balance the number of training examples in such a multi-class setting. The SCUT approach oversamples minority class examples through the generation of synthetic examples and employs cluster analysis in order to undersample majority classes. It is done on multi-class imbalanced datasets.

[1] https://www.kaggle.com/einsteindata4u/covid19.

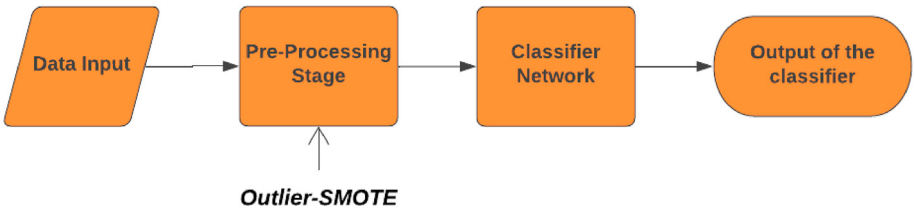

Fig. 1. An illustration describing the stage at which the algorithm works.

3. Outlier-SMOTE algorithm

3.1. Principle

The traditional SMOTE algorithm equally oversamples each data-point in the minority class. But more often than not, it is not necessary to give equal importance to every data-point. The clustered data in the feature space can be oversampled less as compared to its counterparts as the classifier can classify them easily. The authors strongly believe that high importance should be given to the data-points which are far away from the cluster, as they are the samples which are challenging to classify. Outlier-SMOTE works on the same principle. Fig. 2 shows the workflow of the proposed algorithm.

3.2. Steps involved in the algorithm

1. After cleaning the dataset, separate the minority samples, and feed it into the algorithm of Outlier-SMOTE. Let us say, the minority dataset consists of N samples and W features. Now, a Euclidean Matrix is generated using the minority data samples. The dimension of the matrix can be represented as [N,W]. The distance of each point with respect to the others is calculated using the euclidean distance formula mentioned below. After looping through, a matrix of dimension [N,N] is obtained.

$$EUCLIDEAN\ \ DISTANCEi\ \ (m_i\ ,\ m_j)\ =\ \sqrt{\sum_{z=1}^{N}(\,m(i,z)\ -\ m(j,z)\,)} \tag{1}$$

In this equation, i ∈ [0,N] and j ∈ [0,N]. In the Euclidean Matrix [29] all the diagonal elements(where j = i) are zero, as the distance of a sample from itself is zero.

2. For constructing the oversampling matrix, each sample is assigned probability weight p in the range (0, 1) to decide how much importance it gets. Therefore, the sum of the column elements is taken. So, the dimension now changes to [N,1]. This summed matrix denotes the total distance of the samples from the cluster of the minority class. 'ED' in equation (2) signifies the Euclidean distance.

$$\text{EUCLIDEAN SUM (ES)} = [\textstyle\sum ED_{1i}, \sum ED_{2i}, \sum ED_{3i} \ldots\ldots\ldots, \sum ED_{Ni}]_{\ N\ x\ 1} \tag{2}$$

Now, the [N,1] dimensioned matrix may contain all the numerals from [0,∞]. Bringing all the elements under a scale is necessary to decide the importance of each sample. Therefore, normalize the matrix using the formula given below.

$$Normalized\ \ Matrix(NM) = \frac{ES}{sum(ES)} \tag{3}$$

3. This normalized matrix is able to indicate the apt amount of oversampling required for the samples. Now, round off the values to two decimal points, and then multiply them with the percentage of oversampling (T%) mentioned.

$$Oversampling\ \ Matrix = \frac{N*T}{100}[NM] \tag{4}$$

This algorithm is vividly illustrated using the example given below. Here, let us take an example of 5 minority samples (N = 5) which have to be sampled 5 times (T = 500%). This dataset is just an example to illustrate how the algorithm actually functions. Let us start off by considering a euclidean distance matrix (summed) of five synthetic sample data-points whose amount of oversampling has to be deciphered. The ES Matrix is calculated by summing over the columns of the euclidean distance matrix as shown in Point no. 2. Table 1 will clearly show how the algorithm calculates by the oversampling rate of each sample.

In traditional SMOTE, each of the given samples, N = (1,2,3,4,5) would have been given equal priority and would have been oversampled 5 times; which means the closer and farther samples would have the same importance. Whereas, Outlier-SMOTE leverages upon this factor of SMOTE and oversamples each minority data according to its position in the feature space. Farther samples get more importance and the samples in proximity of each other get comparatively lesser importance.

At the same time the total number of samples created stays the same.

- In traditional SMOTE: (5+5+5+5+5) = 25 synthetic samples created.
- In Outlier-SMOTE: (3+5+2 +9+6) = 25 synthetic samples created.

So, Outlier-SMOTE prioritizes the samples and oversamples each one of them accordingly without changing the total amount of synthetic samples generated.

Although, it is a known fact that each dataset has a unique classifier tailored for it which can give the best Precision, F1 and Recall Score, here

Table 1
An example showing the working of Outlier-SMOTE algorithm.

N	1	2	3	4	5
Sum of Euclid Dist Matrix	96	159	51	264	192
Normalized Matrix	0.12598	0.20866	0.06693	0.34645	0.25196
Rounded Off Values	0.13	0.21	0.07	0.35	0.25
Oversampling Rates	3	5	2	9	6

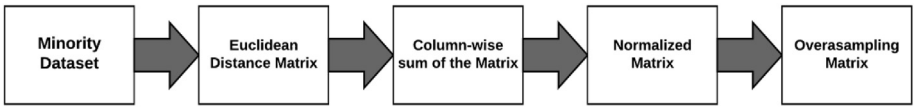

Fig. 2. Workflow of Outlier-SMOTE.

the authors have used Logistic Regression as a blueprint for gauging the performance of the algorithm, as it has a wide reputation of its generalizing capability. The hyper-parameter K-Nearest Neighbors to oversample has been set to a constant (k = 5) as it did not lead to much variation in performance with different values. Since this algorithm is used at a pre-processing stage, the authors have used one classifier and multiple oversampling rates to experiment on its performance.

Future work regarding this field will concentrate on finding an apt distance measure for high-dimensional datasets. Euclidean Distance, though being lucid and robust, fails to capture the intricacies of more than 3 dimensions. Therefore, an alternative distance measure such as Mahalanobis Distance [30] or Manhattan Distance [31] must be taken into consideration. Further sections detail the experiments conducted on the algorithm and the intuitions obtained from them.

4. Methods for evaluating performance

Selecting an apt performance measure for evaluation of the algorithm is a pivotal step, because classifiers trained on imbalanced datasets give very majority-class biased results. An apt performance measure will assist us in judging the algorithm's adaptability in an efficient manner. The main aim should be to reduce the false negatives (FNs) as much as possible.

The most ubiquitous method used in this situation is the confusion matrix [32], which indicates the amount of True Positives(TP), True Negatives(TN), False Positives(FP) and False Negatives(FN). The visual representation of the matrix in Table 2.

In the confusion matrix, the correctly classified samples in the positive (TP) and negative (TN) classes should be as high as possible. This matrix treats the predicted samples class-wise but not as a whole. In this case, the minority samples should be classified as correctly as possible. Since, the goal is the correct classification of minority classes, a well-suited performance metric is needed, such as Precision, Recall and F1-Score.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

Precision is a ratio between the correctly classified positive samples and the total number of positive samples. Precision is a good measure to gauge when the costs for False Positives are high and is normally used in E-Mail spam detection. Recall or Sensitivity is the measure of number of correctly classified positive samples out of the number of observations in the positive class. After considering the above two equations, let us consider the F1-Score which is the harmonic mean of Precision and Recall.

$$F1\ \ Score = \frac{2*Recall*Precision}{Recall + Precision} \tag{7}$$

The authors also remark that the Receiver Operating characteristic (ROC) curve could also be applied to evaluate imbalanced data. It uses the Sensitivity parameter and it plots a graph between False Positive Rate (FPR) and True Positive Rate (TPR). ROC curve assesses the overall classification performance and does not place more emphasis on class-wise performance. Therefore, this paper tests the algorithm on Recall, Precision and F1-Score on various oversampling rates to maximize the fairness while dealing with minority classes.

Table 2
Illustration of the confusion matrix.

	Predicted + ve	Predicted -ve
Actual + ve	TP	FN
Actual -ve	FP	TN

5. Experiment

5.1. Datasets

The authors tested this algorithm on five benchmark datasets with imbalance ratios (minority: majority) ranging from 1:9 to 1:40. The datasets were taken from University of California at Irvine (UCI) Repository,[2] and were fetched using the imblearn Python library [33]. As this work is concentrated on binary imbalanced datasets, this paper takes one class in each dataset as minority and treats the rest as majority. The researchers first test the algorithm on the following datasets before applying it to the COVID-19 data. Table 3 gives a brief overview of the datasets used to test the algorithm.

Fig. 3 shows the majority to minority ratio of the datasets. The barplots clearly indicate that the data used was highly-imbalanced. In the further sections, the authors go on to prove that Outlier-SMOTE drastically increased the Precision, Recall and F1-Score while classifying this data.

5.2. Classifier used

As mentioned above, the work is concentrated in the preprocessing stage, so the authors used just one classifier and gauged the algorithm's performance. Binary Logistic Regression was used in this paper because of its adaptability. The authors have tested the algorithm using different oversampling rates to prove its superiority over other preprocessing algorithms.

5.3. Procedure

Table 4 shows us how the data was segregated while feeding it into the classifier. The table is a visualization of the working scheme of *k-Fold Cross Validation* where $k = 5$. The table is just shown as an illustration of the process. This paper has used the value of k as 10. This method splits the data into 'k' equally distributed parts and uses one of them as a validation set each time. After doing this process k-times, take an average of all scores to get a correct interpretation of the result. This paper took the value of $k = 10$ [34], a value that has been found through experimentation to generally result in a model skill estimate with low bias and a modest variance.

Table 3
Description of the datasets.

	DESCRIPTION	MIN: MAJ	# SAMPLES	# FEATURES
ECOLI DATASET	Classification of proteins based on their amino acid sequences.	35 : 336 (1 : 9)	371	8
ABALONE DATASET	Prediction of the age of abalone	42 : 689 (1 : 16)	731	8
YEAST DATASET	Contains the data of localization of yeast bacteria	51 : 1270 (1 : 24)	1321	9
WINE QUALITY DATASET	Signifies the quality of white wine	175 : 4898 (1 : 27)	5073	11
MAMMOGRAPHY DATASET	Test for breast cancer	260 : 11,183 (1 : 42)	11,443	6

[2] https://archive.ics.uci.edu/ml/datasets.php.

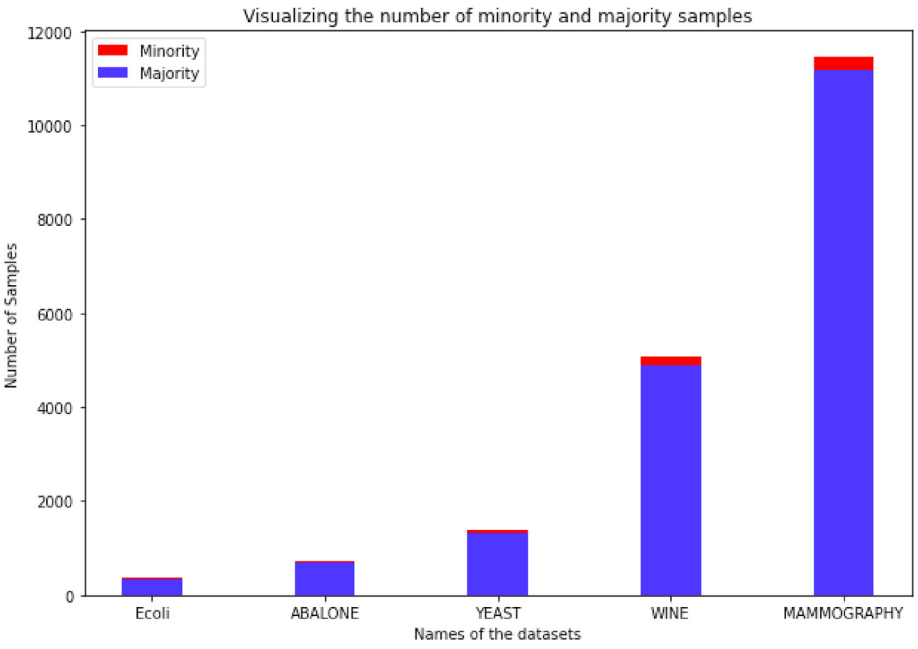

Fig. 3. Bar Graph illustrating the number of majority and minority samples.

Table 4
An illustration of 5-Fold Cross Validation.

DATA				
VALIDATION	Train	Train	Train	Train
Train	VALIDATION	Train	Train	Train
Train	Train	VALIDATION	Train	Train
Train	Train	Train	VALIDATION	Train
Train	Train	Train	Train	VALIDATION

One iteration of the process on a fictitious imbalanced dataset looks as follows:

1) The cleaned dataset is fetched, and is split into majority and minority classes.
2) Further, both the classes are split into 90% and 10% of their respective sets.
3) So, now there are 4 classes:
   a) 90% + 10% of the majority set; and
   b) 90% + 10% of the minority set.
4) Now, combine the 90% of the majority set from 3(a) and the 90% of the minority set from 3(b) and use it as a training set. Similarly, club the 10% majority from 3(a) and 10% minority from 3(b) and make it a validation set.
5) Train on the decided classifier obtained and store the Recall, Precision and F1-Score.

After training for k-times, take an average of all the observations obtained to get the most unbiased result. An important point to note is that K-Fold Cross Validation is only applied on the training data to avoid any data leakage. This process ensures the correct result in any dataset that has tested the algorithm.

6. Dataset results

6.1. Quantitative results

The observed values for Recall, Precision and F1-Score are presented in the tables below. Each table represents the datasets that the authors have tested the algorithm upon. Each table has 5 rows having the oversampling rates varying from 100% to 500%. The performance of Outlier-SMOTE was compared with two benchmark algorithms, traditional SMOTE [12] and ADASYN [14]. Each algorithm uses the same classifier: Logistic Regression. After preprocessing it with the three oversamplers mentioned above, they are fed into the classifier and the results are noted. The comparisons show that, in most cases, Outlier-SMOTE performs better as compared to its counterparts. Tables 5–9 show the results obtained after experimentation where 'OS Rate' represents oversampling rate and O-SMOTE represents Outlier-SMOTE.

6.2. Inferences drawn from the results

For the ECOLI dataset in Table 5, the algorithm surpasses the other two by a substantial amount in recall and F1-Score. The performance

Table 5
Results obtained on **ECOLI** dataset.

	Recall			Precision			F1 - Score		
OS Rate	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN
100	**99.9**	99.4	99.8	88.9	**89.7**	88.6	94.14	94.14	93.9
200	**99.1**	93.6	96.4	85.8	**88.6**	85.2	**92.1**	90.5	90.4
300	**94.2**	88.6	91.1	85.1	**90.0**	88.9	**89.6**	88.6	89.4
400	**91.1**	87.7	89.4	**91.5**	91.23	88.8	**91.1**	88.5	88.6
500	**88.4**	84.33	87.6	**91.7**	93.3	90.1	**90.7**	87	88.5

Table 6
Results obtained on **ABALONE** dataset.

	Recall			Precision			F1 - Score		
OS Rate	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN
100	**99.9**	99	99.89	**90**	84	86.2	**95**	91.3	92.5
200	**99**	94	97.6	81	**81.6**	81	**89.5**	87.6	89.3
300	89.5	88	**89.6**	**85**	84.7	79.6	**86.5**	86.3	84.4
400	**87**	85	85.4	**85.6**	84.8	81.1	**86.2**	85.3	83.4
500	82.5	81.7	**82.6**	**86**	84.8	80.8	83.2	**83.3**	81.7

Table 7
Results obtained on **YEAST** dataset.

	Recall			Precision			F1 - Score		
OS Rate	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN
100	99.99	99.9	99.9	93.1	**95.2**	94.2	**97.5**	96.06	97.0
200	99.99	99.9	99.8	92.4	**94.6**	91.9	**96.1**	95.1	95.8
300	**99.9**	99.3	99.6	**90**	89.6	89.6	**94.4**	94.2	94.1
400	**99.6**	96.6	98.8	**89.4**	86.6	88.6	**93.4**	91.3	93.0
500	**98.4**	96.7	97.2	**87.8**	87.4	86.4	**92.2**	92.04	91.5

Table 8
Results obtained on **WINE-QUALITY** dataset.

	Recall			Precision			F1 - Score		
OS Rate	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN
100	**96.9**	94.9	95.1	**88.3**	86.8	87.2	**92.4**	90.7	91
200	92.9	**93.3**	93	84.5	86.6	**84.7**	88.5	**89.8**	87
300	**89.2**	87.6	88.5	**84.6**	83.8	84	**86.4**	85.7	85.2
400	**87.1**	84.9	85	**83.3**	82.6	82.8	**84.3**	83.7	83.9
500	**82.8**	81.6	81.3	**83.7**	83.5	82.9	**83.2**	82.55	83.1

Table 9
Results obtained on **MAMMOGRAPHY** dataset.

	Recall			Precision			F1 - Score		
OS Rate	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN
100	**99.6**	99.4	99.5	**98.6**	97.7	98.2	**99.1**	98.5	98.9
200	**99.5**	99.3	99.4	97.6	**97.7**	97.0	**98.6**	98.5	98.2
300	99.19	**99.2**	99.2	**97.1**	96.95	95.5	**98.1**	98.0	97.3
400	**99.3**	98.9	98.9	96.2	**96.6**	94.3	97.6	**97.8**	96.5
500	**98.9**	98.5	98.6	**95.9**	95.6	93.4	97.1	**97.2**	95.9

drops down at 300% oversampling but is still higher than the performance of the two algorithms. Recall in the Abalone dataset gives marginally better results than the other two, as it is always on the higher side as shown in Table 6. SMOTE's performance varies a lot in the F1-Score, but nevertheless, Outlier-SMOTE performs better every time. Precision values seem unpredictable here as a major drop in performance is noticed at 200% oversampling. Outlier-SMOTE's performance on the Yeast dataset (refer Table 7) has been the most commendable so far, as the recall and F1-scores are very distinctive and Outlier-SMOTE has performed consistently well in both of them as shown in Table 7. As shown in Table 8, SMOTE and ADASYN have approximately the same Recall rates in the Wine Quality dataset, while Outlier-SMOTE performs marginally better. Here, in F1-Score, SMOTE's performance increases at 200% oversampling, but Outlier-SMOTE surpasses it in the rest. Recall and F1-Score give a more accurate representation of the performance in imbalanced datasets rather than precision. Despite giving the performance at 300%, the recall score is consistently higher for Outlier-SMOTE in Mammography dataset (refer Table 9), and while the F1-Score gradually goes down with the increase in oversampling, Outlier-SMOTE manages to perform better for majority of the oversampling rates.

Clearly, Outlier-SMOTE performs better than SMOTE or ADASYN in majority of the cases. The performance is especially commendable on ECOLI, Yeast and US-Crimes dataset where it surpasses the performance in almost all the oversampling rates. Here, the paper focuses more on the recall and F1-scores only as they are more indicative of the algorithm's generalization capability.

7. Comparative analysis of Outlier-SMOTE with SMOTE and ADASYN on the highly imbalanced COVID-19 dataset

This section presents the analysis of various crucial and pivotal symptoms of COVID-19 using the dataset obtained from Kaggle. The authors hope their work can be utilized to garner further insights, and pray for the world to return to its normalcy as quickly as possible.

7.1. Dataset description

This dataset contains anonymized data from patients seen at the Hospital Israelita Albert Einstein, at São Paulo, Brazil, and who had samples collected to perform the SARS-CoV-2 RT-PCR and additional laboratory tests during a visit to the hospital. The link to the dataset is provided here. This dataset has 5644 test samples of various patients being tested for COVID-19. The patients were evaluated on 111 attributes such as Haemoglobin, Platelets, Arterial Blood gas analysis, etc. Out of

5644 tests, only 553 people tested positive for COVID-19. This statistic clearly indicates a 1 : 9 (minority to majority class) ratio. Since, only one reliable dataset was found which denoted the clinical symptoms of COVID-19 clearly, the authors tested their hypothesis on only this original and trusted dataset from Kaggle. In the future, if any dataset is obtained from an unverifiable source, the authors suggest to cross-check the results with other similar datasets to negate the chances of mislabeling in the data.

The data had lots of Null and NaN values, because of lack of data from the patients. Therefore, the features with >90% of Null values were removed. Further, to clean the data, the variables with zero variance were removed as they consist of only one value and do not have any significance. The last 19 columns were related to the presence of antigens (binary values - 0 or 1), and the null values were immense in these, therefore a row-wise sum was taken. All the 19 columns were combined into one column named 'other_disease'. It was found that 13% of the patients tested positive for the presence of at least one antigen. The rest of the scattered null values were replaced by the mean of their counterparts. After trimming all the unnecessary attributes, the authors narrowed down to 39 variables on which the model will be trained. Fig. 4 shows the dependency of each attribute with respect to the other in the form of a correlation matrix.

As illustrated in Fig. 4, there are a myriad of variables which are dependent on one another such as Haemoglobin and Hematocrit, Red Blood cells and Hematocrit, Platelets and Leukocytes and many more. Higher the value, the more importance the feature has in determining the probability of COVID-19. Fig. 4, describes some of the pivotal features which may turn out to be useful in the detection of the disease.

7.2. Classification strategies

The authors concentrate on improving the performance only in the pre-processing stages. Therefore, they use only one best performing classifier with multiple oversampling rates and tested it on Recall, Precision and F1-Scores. The main focus is on the *Recall* factor as the main aim is on minimizing the False Positives.

Although Logistic Regression and Random Forest classifier gave us roughly the same results, Random Forest classifier with n_estimators = 100 was preferred, as the authors wanted to present a classifier which will be able to perform regardless of substantial change in the data in the future. This paper solely focuses on the effect of various oversampling rates on the accuracy of the classifiers. The five benchmark datasets were classified using Logistic Regression because of its better generalizing capability, and because it gave the highest possible result. Also, Random Forest is popular for handling more complex datasets as compared to Logistic Regression. The authors specifically chose Random-Forest for only COVID-19 datasets so that they could achieve a state-of-the-art combination of the Outlier-SMOTE algorithm and a well-structured classifier in the field. It could potentially help to gain more insights about the disease.

7.3. Experiments

Table 10 illustrates the results obtained after experimenting with COVID-19 dataset. The results are obtained after performing 10-fold cross validation on the dataset. The model was tested on five oversampling rates (between 100% and 500%) with three parameters, Recall, Precision and F1-Score, similar to the five benchmark datasets. Upon obtaining results, it is noted that the COVID-19 dataset's peak performance is obtained at 500% oversampling, and since the dataset had to be trimmed to remove the Null and NaN values, the performance rates are quite unconventional as opposed to the five datasets tested above. As seen, since COVID-19 dataset was an imbalanced dataset with the imbalance ratio of 1:9, the algorithm performed efficiently as compared to the other two. With no oversampling, the classifier almost fails to distinguish between the positive and negative classes, and as oversampling rate increases, the performance boosts up, which gets us to believe that for detecting the crucial symptoms of COVID-19, it is absolutely essential to get more authentic data. For the specific dataset presented above, the authors conclude Outlier-SMOTE is by-far the best option to improve the performance of the classifier.

Random Forest classifier comes with a feature importance attribute that outputs an array between 0 and 1 representing how useful the feature is to the model while predicting the output. Random forest uses gini importance or mean decrease in impurity (MDI) to calculate the importance of each feature. As shown in Fig. 5, Leukocytes and Platelets amass the greatest importance in prediction of the COVID-19 disease. Surprisingly, a patient's age doesn't contribute much in the detection of the disease. The feature importance graph is pivotal in distinguishing features which actually contributed to the accuracy and the ones which were just noise. Further extrapolating this technique, we use the very useful SHAP [35] to visualize the variation in output of the model with the change in values of individual features. SHAP and Gini Importance agree on a lot of features such as the fact that leukocytes, platelets and eosinophils indicate a greater chance of presence of COVID-19. Gini can be defined as a statistical measure of the degree of variation represented in a set of values. It should be noted that Fig. 5 only provides the names of the features which were pivotal in detection of the disease. The way through which the values (higher or lower) of these features affect the symptomatology of COVID-19 is discussed through SHAP (Fig. 6).

This paper used the very interesting SHAP (Shapely Additive Explanations) [35] method to visualize the readings. SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. The authors would like to remark that SHAP technique was used as an orthogonal validation of their model. As shown in Fig. 6, SHAP actually denotes which features are important or which features are dominant while giving the result. For instance, a high value of Mean Corpuscular Volume (MCV) drives the result towards a negative output which means a high level of MCV may be a symptom of a COVID-negative person. SHAP values are calculated by taking the average of the marginal contributions of all the features. SHAP repeatedly changes each variable's value and sees how it would affect the result and plot the graph. The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The X-axis contains the impact of the feature on the result, negative values mean a low impact and positive values mean a higher impact. The Y-Axis has all the features that have been observed.

The authors found certain interesting pointers through the correlation matrix and extracting features from the data using SHAP in Fig. 5, which are noted:

- If a patient has a high value of Monocytes, it is a strong indicator of the presence of COVID-19.
- COVID-19 presence is also indicated by low values of Eosinophils, Leukocytes and Platelets.
- If the patient is tested positive for 'other_disease' (i.e., if its value is 1), it is highly unlikely that the person will have COVID-19. This means that presence of other similar diseases does not indicate the presence of coronavirus.
- Since Outlier-SMOTE works efficiently on this dataset, it can be inferred that the remotely placed minority samples in the distribution space have features which are a strong indication of corona-virus.
- False Positives or negatives can be reduced only upto a certain extent by researchers using a couple of datasets. Further assessments and inferences can only be drawn by an experienced doctor.

8. Discussion

Outlier-SMOTE is an algorithm which oversamples the minority datapoint according to its position in the sample space. The data-point which is outlying or is in a remote position gets oversampled more than the others, and the data-point which is near to the cluster gets a lesser

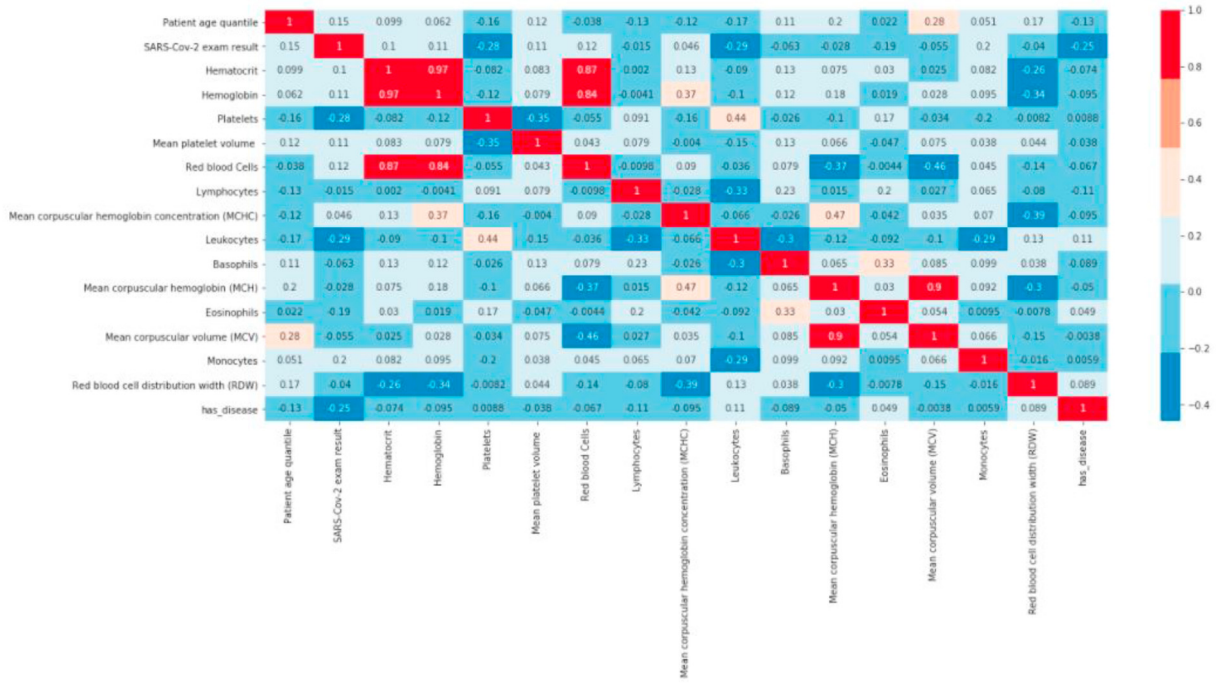

Fig. 4. Correlation Matrix denoting the importance of the various features in the considered COVID-19 dataset.

Table 10
Results obtained on **COVID-19** dataset.

OS Rate	Recall			Precision			F1 - Score		
	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN	O-SMOTE	SMOTE	ADASYN
100	69	**69.8**	64.2	**73.2**	69.5	69.2	**70.4**	68.1	58.0
200	**78.2**	71.3	70.2	**82.8**	77.3	76.6	**80.3**	74.1	71.4
300	**89.9**	82.13	81.5	**87.2**	80.8	80.8	**90.6**	81.44	80.8
400	**87.6**	86.7	87.45	**90.4**	84.12	83.9	**88.9**	85.37	85.5
500	**88.8**	88	86.9	**91.4**	85.5	85.07	**90.1**	86.77	85.86

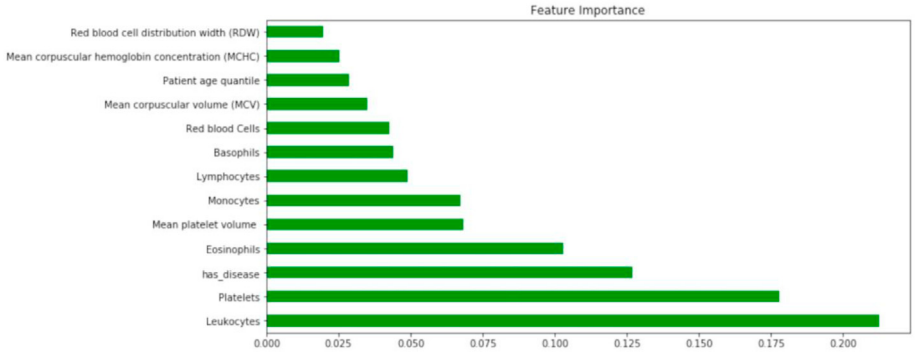

Fig. 5. Feature Importance of each attribute in the dataset.

amount of oversampling. Outlier-SMOTE reduces the chances of the synthetic samples overlapping with the other samples, as compared to SMOTE as it prioritizes only the remotely placed samples. Although Outlier-SMOTE variably oversamples each minority data-point, the total amount of oversampling for the complete minority set remains constant. The primary incorporation of Outlier-SMOTE in automated frameworks may be in mobile applications which predict the probability of COVID-19. If the incorporation of COVID-19 questionnaire and the algorithm is done together, the prediction of probability of the disease can be bolstered.

There are a few areas of improvement in this algorithm which the researchers will look forward to as their future work. First one would be the Euclidean Distance. In case of lower-dimensional datasets, Euclidean distance is a perfect measure to calculate the distance from one point to another. But for higher dimensions, the nuances of the data might get lost while calculation which might lead to reduction in performance. Although this paper considered a fixed oversampling rate of 100%–500%, the authors claim that there is no fixed oversampling rate for any dataset to ensure maximum performance. Each dataset has a unique oversampling rate which will help the classifier to achieve maximum accuracy.

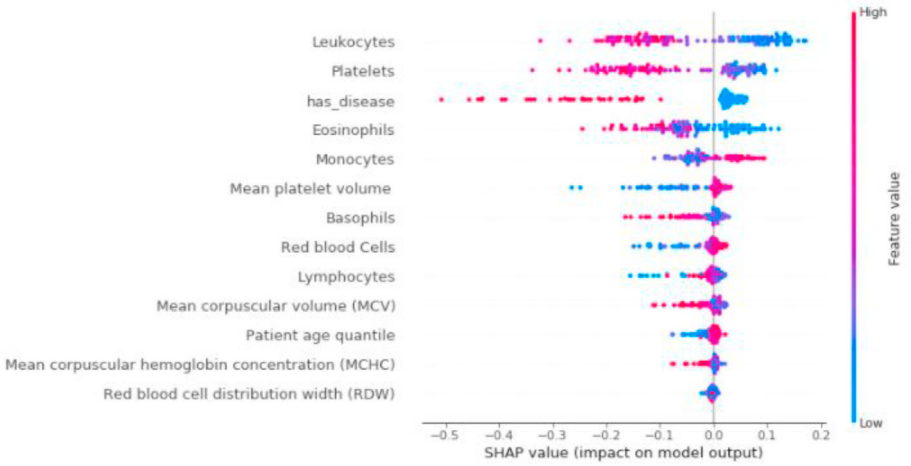

Fig. 6. SHAP values for each feature in the dataset.

9. Conclusion and future work

A classifier trying to learn from an imbalanced dataset can cause a major bottleneck for researchers trying to train a model. Imbalanced datasets often decrease the performance of the classifiers by a substantial amount. One of the many ways to counter this problem is to oversample the minority datasets, where SMOTE (Synthetic Minority Oversampling Technique) amassed a lot of attention. This paper presents an improved version of SMOTE called the Outlier-SMOTE. This algorithm calculates the amount of times each minority data-point has to be oversampled. O-SMOTE gives greater priority to the outlying/remote minorities and oversamples them more as compared to their counterparts. Before applying it to the COVID-19 dataset, the algorithm was rigorously tested on 5 benchmark datasets and compared with 2 other oversampling algorithms, SMOTE and ADASYN, and it was found that Outlier-SMOTE performed remarkably well on every dataset. To contribute further to the literature, the researchers outlined the importance of each feature using the dataset using the correlation matrix and SHAP Analysis. This paper also throws light on certain trivial symptoms which might be a crucial detector of the presence of COVID-19 disease. After confirming its performance on the normal datasets, the authors tested it on the COVID-19 symptoms dataset on which Outlier-SMOTE surpassed in almost every parameter. The authors express their gratitude to the extensive work done on the data by Lucas Moda on Kaggle [36]. The authors salute the heroic efforts taken by the doctors [1], police and the government to combat this pandemic.

In this paper, the five benchmark datasets from UCI repository played a major role in determining the performance of the algorithm. To maximize the performance of Outlier-SMOTE, an algorithm for determining a custom oversampling rate for each dataset has to be devised, which the authors look forward to as future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] "How the world's scientists, doctors, and nurses are uniting to fight COVID-19," unfoundation.org. Apr. 28, https://unfoundation.org/blog/post/how-worlds-scientists-doctors-and-nurses-uniting-fight-covid-19/. [Accessed 23 July 2020].
[2] "Class imbalance: a classification headache | by gonzalo ferreiro volpi | towards data science.". accessed Jul. 23, 2020, https://towardsdatascience.com/class-imbalance-a-classification-headache-1939297ff4a4.
[3] Fernández A, García S, Herrera F. "Addressing the classification with imbalanced data: open problems and new challenges on class distribution. Berlin, Heidelberg: in Hybrid Artificial Intelligent Systems; 2011. p. 1–10. https://doi.org/10.1007/978-3-642-21219-2_1.
[4] Chawla NV, Japkowicz N, Kotcz A. Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explor Newsl 2004;6(1):1–6. https://doi.org/10.1145/1007730.1007733.
[5] Luque A, Carrasco A, Martín A, de las Heras A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recogn 2019;91:216–31. https://doi.org/10.1016/j.patcog.2019.02.023.
[6] Awoyemi JO, Adetunmbi AO, Oluwadare SA. "Credit card fraud detection using machine learning techniques: a comparative analysis,". In: In 2017 international conference on computing networking and informatics (ICCNI); 2017. p. 1–9. https://doi.org/10.1109/ICCNI.2017.8123782.
[7] Rahman MM, Davis DN. Addressing the class imbalance problem in medical datasets. Int J Mach Learn Comput 2013:224–8. https://doi.org/10.7763/IJMLC.2013.V3.307.
[8] Lusa L. SMOTE for high-dimensional class-imbalanced data. BMC Bioinf 2013;14(1):106.
[9] Rodda S, Erothi USR. "Class imbalance problem in the network intrusion detection systems,". In: 2016 international conference on electrical, electronics, and optimization techniques (ICEEOT); 2016. p. 2685–8. https://doi.org/10.1109/ICEEOT.2016.7755181.
[10] He H, Garcia EA. Learning from imbalanced data. IEEE Trans Knowl Data Eng 2009;21(9):1263–84. https://doi.org/10.1109/TKDE.2008.239.
[11] Prusty MR, Jayanthi T, Velusamy K. Weighted-SMOTE: a modification to SMOTE for event classification in sodium cooled fast reactors. Prog Nucl Energy 2017;100:355–64. https://doi.org/10.1016/j.pnucene.2017.07.015.
[12] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 2002;16:321–57.
[13] Song Z, Roussopoulos N. K-nearest neighbor search for moving query point. In: Advances in spatial and temporal databases; 2001. p. 79–96. https://doi.org/10.1007/3-540-47724-1_5. Berlin, Heidelberg.
[14] He H, Bai Y, Garcia EA, Li S. "ADASYN: adaptive synthetic sampling approach for imbalanced learning,". In: 2008 IEEE int. Jt. Conf. Neural netw. IEEE world congr. Comput. Intell.; 2008. https://doi.org/10.1109/IJCNN.2008.4633969.
[15] Random Oversampling and Undersampling for Imbalanced Classification (accessed Jul. 23, 2020), https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/.
[16] Ahn G, Park Y-J, Hur S. "A membership probability–based undersampling algorithm for imbalanced data. J Classif, Jan 2020. https://doi.org/10.1007/s00357-019-09359-9.
[17] Han H, Wang W-Y, Mao B-H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing; 2005. p. 878–87.
[18] Ma L, Fan S. CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. BMC Bioinf 2017;18(1):169. https://doi.org/10.1186/s12859-017-1578-z. Mar.
[19] Sáez JA, Luengo J, Stefanowski J, Herrera F. SMOTE–IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Inf Sci 2015;291:184–203.
[20] Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-level-smote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia conference on knowledge discovery and data mining; 2009. p. 475–82.
[21] Hooda S, Mann S. Imbalanced data learning with a Novel ensemble technique: extrapolation-SMOTE SVM bagging. Int J Grid Distrib Comput 2020;13. 01, Art. no. 01, http://sersc.org/journals/index.php/IJGDC/article/view/21360. Accessed: Jul. 27, 2020. [Online]. Available.

[22] Breiman L. Bagging predictors. Mach Learn 1996;24(2):123–40. https://doi.org/10.1007/BF00058655.
[23] Wang Q, Luo Z, Huang J, Feng Y, Liu Z. A Novel ensemble method for imbalanced data learning: bagging of extrapolation-SMOTE SVM. Comput Intell Neurosci 2017. Jan. 30, https://www.hindawi.com/journals/cin/2017/1827016/. Jul. 23, 2020.
[24] Wang K-J, Makond B, Chen K-H, Wang K-M. A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients. Appl Soft Comput 2014;20:15–24. https://doi.org/10.1016/j.asoc.2013.09.014.
[25] Particle swarm optimization - IEEE conference publication." https://ieeexplore.ieee.org/document/488968 (accessed Jul. 23, 2020).
[26] Chawla NV. C4.5 and Imbalanced Data sets: investigating the eect of sampling method, probabilistic estimate, and decision tree structure. 2003.
[27] Seiffert C, Khoshgoftaar TM, Hulse JV, Napolitano A. RUSBoost: a hybrid approach to alleviating class imbalance. IEEE Trans Syst Man Cybern - Part Syst Hum 2010. https://doi.org/10.1109/TSMCA.2009.2029559.
[28] Agrawal A, Viktor HL, Paquet E. SCUT: multi-class imbalanced data classification using SMOTE and cluster-based undersampling. In: Proceedings of the 7th international joint conference on knowledge discovery. Lisbon, Portugal: Knowledge Engineering and Knowledge Management; 2015. p. 226–34. https://doi.org/10.5220/0005595502260234.
[29] Dokmanic I, Parhizkar R, Ranieri J, Vetterli M. Euclidean distance matrices: essential theory, algorithms and applications. IEEE Signal Process Mag 2015;32(6):12–30. https://doi.org/10.1109/MSP.2015.2398954.
[30] Martos G, Muñoz A, González J. On the generalization of the Mahalanobis distance. In: Progress in pattern recognition, image analysis. Berlin, Heidelberg: Computer Vision, and Applications; 2013. p. 125–32. https://doi.org/10.1007/978-3-642-41822-8_16.
[31] Craw S. Manhattan distance. In: Sammut C, Webb GI, editors. Encyclopedia of machine learning and data mining. Boston, MA: Springer US; 2017. p. 790–1.
[32] Visa S, Ramsay B, Ralescu A, VanDerKnaap E. Confusion matrix-based feature selection. Fac Artic 2011:120–7 [Online]. Available, https://openworks.wooster.edu/facpub/88.
[33] imbalanced-learn API — imbalanced-learn 0.5.0 documentation. accessed Jul. 22, https://imbalanced-learn.readthedocs.io/en/stable/api.html; 2020.
[34] Brownlee J. A gentle introduction to K-Fold cross-validation,". Machine Learning Mastery 2018. May 22, https://machinelearningmastery.com/k-fold-cross-validation/. Oct. 24, 2020.
[35] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. Advances in neural information processing systems 30. Curran Associates, Inc.; 2017. p. 4765–74.
[36] COVID-19. Optimizing recall with SMOTE. accessed Nov. 07, https://kaggle.com/lukmoda/covid-19-optimizing-recall-with-smote; 2020.