Class-Decomposition and Augmentation for Imbalanced Data Sentiment Analysis

Carlos Francisco Moreno-García
School of Computing
Robert Gordon University
Aberdeen, AB10 7GJ, UK
Email: c.moreno-garcia@rgu.ac.uk

Chrisina Jayne
School of Computing, Engineering and
Digital Technologies Teesside University
Middlesbrough TS1 3BX , UK
Email: c.jayne@tees.ac.uk

Eyad Elyan
School of Computing
Robert Gordon University
Aberdeen, AB10 7GJ, UK
Email: e.elyan@rgu.ac.uk

***Abstract*—Significant progress has been made in the area of text classification and natural language processing. However, like many other datasets from across different domains, text-based datasets may suffer from class-imbalance. This problem leads to model's bias toward the majority class instances. In this paper, we present a new approach to handle class-imbalance in text data by means of unsupervised learning algorithms. We present class-decomposition using two different unsupervised methods, namely k-means and Density-Based Spatial Clustering of Applications with Noise, applied to two different sentiment analysis data sets. The experimental results show that utilizing clustering to find within-class similarities can lead to significant improvement in learning algorithm's performances as well as reducing the dominance of the majority class instances without causing information loss.**

***Index Terms*—Sentiment Analysis, Text Imbalanced Datasets, Class Decomposition**

I. Introduction

Significant progress has taken place in the area of text and sentiment analysis. This is partly due to the growing content on social media platforms such as Twitter and Facebook. It is also due to the significant progress that took place in the area of Natural Language Processing and Deep Learning. Sentiment analysis, in particular, attracted significant research efforts over the past decade. It is concerned with the analysis and understanding of user views and opinions and is often referred to as the sentiments [1].

In recent years, sentiment analysis has been used across a wide range of applications. Typical examples include investigating the relationship between user's tweets and the financial market, where high correlations between stock prices and tweets sentiment were uncovered [2]. Another common area of applications of sentiment analysis is the understanding of people's opinions and reviews on certain products or services. Examples include customers reviews on Amazon products [3], [4]. Politics is also another area where sentiment analysis has been successfully used to understand public opinions [5].

In almost all of these related applications, the analysis of people opinions (sentiments) can be treated as a supervised learning problem, where the input features are made of a set of attributes extracted from unstructured text (e.g. a tweet, customers review, political opinion of a user, etc...), and the target variable is a label that indicates whether the sentiment is positive, negative, or in some cases neutral.

Similar to other supervised learning problems, understanding the sentiment can be more challenging if the dataset is hugely imbalanced. In other words, if for example most of the sentiment of users in a particular dataset is negative. In this case, some data-sampling methods, or algorithmic modification needs to be carried out prior to the classification task [6]. The class-imbalance is a widely researched topic in the area of supervised machine learning [7], [8]. This is an inherently challenging problem to most state-of-the-art supervised learning algorithms and is common across a wide range of domains including sentiment analysis [9]. In a binary dataset, the problem happens when the distribution of the two classes is hugely imbalanced, which often leads learning algorithms to be biased toward the majority class-instances. In most literature, rare instances in the dataset are often referred to as the positive instance or the class of interest, while majority class instances are often referred to as negative instances [10].

The degree of the class-imbalance often determines how challenging the problem is. Often, this is defined as the imbalance ratio (IR) as shown in Equation 1, or the percentage of the minority-class instances as defined in Equation 2, where M and m represent the number of instances in the majority class and minority class, respectively [11].

$$IR = \frac{m}{M} \tag{1}$$

$$m_{percentage} = \frac{m}{M} \times 100 \tag{2}$$

A wide range of techniques is often employed to handle such problems. Such techniques range from data-sampling methods such as random or cluster-based sampling, [6], algorithmic-based solutions [7], [8] and more recently the use of Generative Adversarial Neural Network [12] to generate more data and capture more data variance to improve the learning process compared to traditional data augmentation methods.

One of the most common methods is random sampling. This can be either random under-sampling where negative instances

of the data are randomly removed to reduce the imbalance degree, or random oversampling to increase the number of positive instances in the datasets. The method is simple to implement and often leads to less-biased results. However, random under-sampling also can lead to information loss, while random oversampling may result in model's overfitting [13]. In some applications, random data sampling does not improve results [14].

A better and more common approach is the Synthetic Minority Oversampling Technique (SMOTE) [15]. This method is designed to synthesise new data points by interpolating neighbouring instances. The method proved to be effective in handling class-imbalance and has been used across a wide range of real-world applications [16]–[18]. In addition, various extensions have been proposed based on the original methods, including DBSMOTE [19], SLSMOTE [20], MWMOTE [21] and others.

A more recent work that used SMOTE was presented in [10]. Here, the authors, proposed a new method called CDSMOTE based on SMOTE and class-decomposition [22], [23]. CDSMOTE works by under-sampling the majority class instances by means of unsupervised learning algorithms (e.g k-means) and over-sampling the minority class-instances based on some heuristics using SMOTE. The experiments showed that the proposed method does not lead to information loss. This is mainly because under-sampling here refers to clustering the majority class-instances into sub-clusters, which results in less imbalanced datasets and at the same time provides a more fine-grained training to the learning algorithms. Fig. 1 shows how the imbalance in a dataset can be reduced by applying CDSMOTE.

In this paper, we propose a new approach utilising the CDSMOTE and applying it to the analysis of sentiment in textual data. The main contributions of this paper are as follows:

- A new way to uncover within class similarities using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) instead of k-means to provide more meaningful sub-clusters within the majority class-instances.
- A novel application of the CDSMOTE method for non-binary datasets and textual data.
- Thorough experiments utilising the proposed method and two state-of-the-art classification algorithms, namely Support Vector Machines and Random Forests.

The intuition for using class decomposition to find within-class similarities is that the degree of positivity or negativity within a piece of text can be expanded beyond just negative, positive, or neutral. In other words, positive sentiment can also be clustered into very positive, positive, or moderately positive, and the same applies to the negative instances. By applying unsupervised machine learning algorithms such as k-means or DBSCAN, we can reveal these degrees of various sentiment, and at the same time enhance the classification performance.

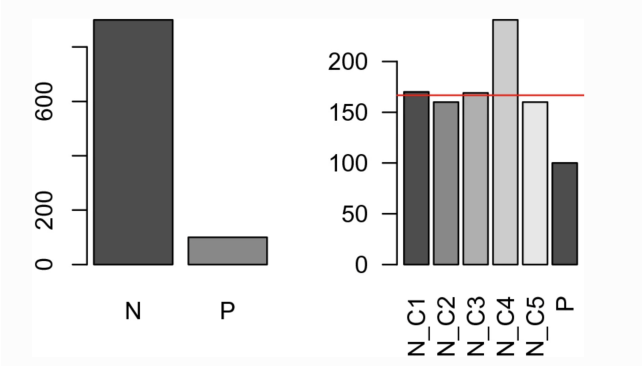

Fig. 1. An example of how the CDSMOTE method presented in [10] clusters data of the negative class N to create sub-classes which balance the dataset. Afterwards, data augmentation is applied to the positive class P to further balance the dataset.

The remaining parts of this paper are as follows: Section 2 presents the methods in detail, Section 3 presents the datasets, experiments, discussion, and results. Finally, conclusions and future directions are outlined in the last section.

II. Methods

A. Word Embeddings

In order to apply machine learning to text classification, the text has to be represented as numeric data. One way to convert text into vector representation with numbers is to use one-hot encoding i.e. associate a unique integer number with every word and turn the integer index into a binary vector. This results in the encoding of a text with very high dimensional vectors (i.e. the size of the vocabulary). Another way is to use word embeddings i.e. encoding of words or phrases from a language vocabulary to vectors of real numbers. Word embeddings encode very large vocabularies in low-dimensional vectors and these are learned from data.

In this work we utilise three widely established techniques for converting text data into numerical representations: Term Frequency-Inverse Document Frequency (TF-IDF) [24], Global Vectors for Word Representation (GloVe) [25] and Contextualized Word Representation [26] , [27].

TF-IDF technique involves calculating a value that reflects how important a word/term t is to a document d in a corpus D utilising two statistics: term frequency (tf) and inverse document frequency (idf).

$$tfidf(t, d, D) = tf(t, d) * idf(t, d, D),$$

where $tf(t, d)$ is the number of times that term t occurs in the document d and $idf(t, d, D) = \log \frac{N}{n(t,d,D)}$. Here N denotes the number of documents in D and $n(t, d, D)$ denotes the number of documents in the corpus where the term t appears.

Global Vectors for Word Representation (GloVe) is an unsupervised learning algorithm for obtaining vector representations for words. It is based on a global log bilinear regression model that combines global matrix factorization and local context window methods [25]. The GloVe model is trained on aggregated global word-word co-occurrence

matrix from a corpus which captures the frequency of words that co-occur with one another in a given corpus. GloVe6.b provides pre-trained word vectorizations with 100, 200, 300 dimensions trained over large corpora, including Wikipedia 2014, Gigaword 5 and Twitter content [1]. In this particular work, we use a word vectorization with dimension 300.

Contextualized Word Representation is a word embedding technique that enables learning an embedding that captures the meaning of the word from the text so that similar words have similar embeddings. It was introduced for the first time in [26] using bidirectional long short-term memory (LSTM). In this work, we learn word embedding by training a deep learning model with an Embedding layer, LSTM layer, dropout, and batch normalisation on the specific classification task. The trained embeddings are then used as input to the class decomposition followed by training of a classifier e.g. SVM, Random Forest. We denote this embedding with CWR-LSTM.

Text pre-processing is performed before applying the vectorization/embedding methods. This includes tokenisation (breaking a stream of text into words), contractions (resolving expressions like you're, I'm, etc.), removing URL, non-ascii and specials charters, removing punctuations, stop words, and stemming (modifying words to obtain variant word forms using different linguistic processes such as adding of affixes [28]).

B. CDSMOTE for multi-class datasets

The original CDSMOTE method presented in [10] is comprised of two steps: 1) class decomposition to redistribute the number of samples per class without losing any sample and 2) oversampling the new minority class(es) to reduce the dominance of the new majority class(es). Regarding the first step, class decomposition can be broadly described as the process of clustering class-instances into smaller groups by means of unsupervised learning algorithms. As a result, the dominance of a class can be greatly reduced without losing any information. To address multi-class imbalance in sentiment analysis datasets, we present two adaptations of the original CDSMOTE method. In the first one, called *CDSMOTE-kmeans*, we use k-means clustering (with a range of different fixed k values) to target only the majority class and produce a more balanced dataset, reducing the bias of the classification models towards the minority classes. In the second variant, called *CDSMOTE-DBSCAN*, we use *DBSCAN* to cluster all classes in the dataset, even if this means that minority classes are further decomposed in smaller ones. This is done with the aim of finding hidden patterns in data and augmenting samples with respect to their most similar instances only.

This approach enables detecting genuine subclasses and it improves accuracy. A key element of the class decomposition is the choice of the k value, which can influence the overall performance of the learning algorithms. Methods in the literature to select the k value can be either based on experimental work or using some optimisation methods. A typical example is presented in [23], where Random Forests (RF) over class decomposed medical diagnosis data sets has been adopted. The authors performed an exhaustive search over a set of iterations to find the best k values for each class and then decomposed the classes accordingly. A heuristic was used to discard minority classes from the decomposition process. Experiments showed that by decomposing the datasets into subclasses favourable results can be achieved. The improvement of the results was attributed to the diversified search space resulting from the decomposition process. In [22], an evolutionary-based method namely Genetic Algorithm was used to optimise a set of parameters including the best k values, and again an improved classification accuracy was achieved when the proposed method was tested on 22 different life science and medical datasets. More recently, class-decomposition was successfully applied to handle class-imbalance across various public and common imbalanced binary datasets [10]. The authors applied class-decomposition to reduce the dominance of the majority class instances, to then oversample the minority class instances.

Intuitively speaking, consider a dataset where the two classes represent a patient condition (sick, healthy). By applying class-decomposition to the sick instances, we may end with the sick instances re-grouped into three clusters: mildly sick, sick and very sick.

Let us consider a set of instances $x_i = x_1, ..., x_n$ belonging to a dataset D, where each instance x_i is mapped to a discrete class label in $Y = \{P, NA, NB\}$. Moreover, P is the majority class (i.e. that the majority of samples in D is mapped to this class label), and both NA and NB being minority classes. We do not consider any imbalance ratio IR at this stage (as defined in 1); this means that the difference in samples between the majority class and any of the minority classes is not relevant.

For the *CDSMOTE-kmeans* variant of our method, we segregate all instances $x_i \in Y = P$ into a new subset D'. Then, we apply k-means clustering to D^P, which results in the samples of D^P being mapped to a new set of classes P', where $P' = \{p'_1, ..., p'_k\}$, being k the number of clusters selected for k in advance. Previous experiments in [10] showed that k values between 2 and 5 are optimal, provided that the imbalance ratio between majority and minority classes is not too high (i.e. $IR > 50$).

For the *CDSMOTE-DBSCAN* variant, we segregate the samples of each class into different subsets depending on the label. In this example, three sub-datasets D^P, D^{NA} and D^{NB} are created. That is, D^P being the set with samples with $Y = P$, D^{NA} where $Y = NA$ and D^{NB} where $Y = NB$. Then, *DBSCAN* clustering is applied to automatically find different numbers of clusters for each subset. Finally, each sample is assigned a new label based on this clustering.

After the class decomposition stage, both variants use the following augmentation approach for the second step. Firstly, we calculate the average number of samples avg for all classes and subclasses. Then, a threshold τ is set. If the total number of samples of a given subclass is smaller than $|avg - \tau|$, then this class is augmented using SMOTE [15];

[1]https://nlp.stanford.edu/projects/glove/

otherwise, the class is left untouched. Notice that even in the case that the subclass belongs to the original majority class, the augmentation is still carried out. Our experiments in Section III show that this approach improves or maintains the prediction accuracy of the majority class as well as the prediction accuracy for the minority classes.

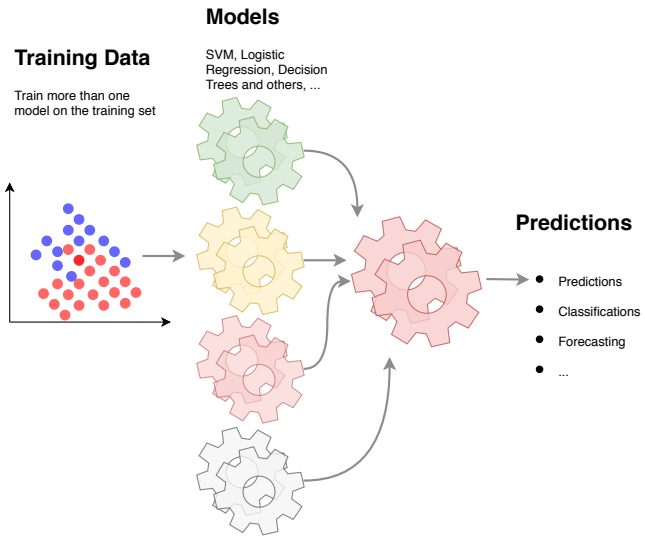

Fig. 2. Schematic diagram of ensemble

C. Classification Models

Wide range of supervised machine learning algorithms can be applied to map an instance x_i to a particular class label y. In this paper, we used two different learning algorithms to assess the impact of class-decomposition on class-imbalance. These are Random Forest (RF) and Support Vector Machines (SVM).

RF is an ensemble classification and regression technique introduced by Breiman et al. [29] that has proved to be a highly accurate prediction and classification technique. The ensemble is designed to train more than one classifier, and then aggregate the predictions of all models and perform predictions by majority voting as can be seen in Fig. 2. A good ensemble needs models to be diverse enough and independent from each other to ensure good performance. Broadly speaking, diversifying the ensemble can either include training more than one type of machine learning algorithm (e.g. SVM, Logistic Regression, ...) or alternatively, training one machine learning algorithm on various and diverse subsets of the training set. RF generates a diversified ensemble using Bootstrap aggregating (Bagging). Bagging is a sampling method that samples data from the training set with replacement. With such an approach an instance in the dataset can be sampled more than one time for the same model. At the same time, other instances may not appear at all during the training process. It is estimated that following this approach, more than 63% unique instances from the training set will be used during the training process, while almost 37% of the instances will not be sampled at all, and will be used to estimate the "out-of-bag" error. In addition, and to ensure more diversified ensemble RF and at each node split, only a subset of features are drawn randomly to assess the quality of each feature.

According to the winning solutions in *Kaggle*[2], the state-of-the-art ensemble methods are RF [29] and Gradient Boosting trees [30]. In one of the largest experiments where more than 179 classifiers were used on 121 different datasets from the UCI repository[3] [31], RF came first, followed by SVM with Gaussian Kernels.

SVM [32] is another supervised machine learning algorithm that boosts classification accuracy by projecting the data points to a higher dimensional space aiming at finding an optimal hyperplane that separates positive and negative classes. It has also proven its superiority over other classification methods. In [31] and when compared to other widely adopted learning algorithms, SVM with Gaussian kernel ranked second after RF without statistically significant difference. A recent systematic review of the literature shows that SVM is considered among the most common approaches in handling class-imbalanced datasets [6].

III. Experiments and Discussions

A. Data Repositories

We utilised two data repositories: the first one is related to sentiment analysis on customer satisfaction reviews directed to six major US airlines on the *Twitter* social media platform [4]. It is composed of 9178 (62.69%) negative reviews (from now on referred to as class 0), 3099 (21.17%) neutral reviews (class 1) and 2363 (16.14%) positive reviews (class 2). Some of the information that appears on this dataset is a normalised confidence score for the sentiment, the characters that constitute the reasons to consider the statement negative, the airline to which the tweet is directed, the user, location, and the number of retweets.

The second data repository used was also based on tweets, but this time related to the convictions of people to believe in global warming[5]. This dataset only has three features: content, sentiment, and sentiment score. The tweets can claim either no existence of global warming (class 0), a neutral or informative position on the issue (class 1), or an affirmation of the existence of this phenomenon (class 2). There is a total of 1117 (18.34%) class 0 tweets, 1862 (30.57%) class 1 tweets and 3111 (51.09%) class 2 tweets.

It is important to highlight that while class 0 is the majority one on the *Airline* data repository, in the Global Warming one it is class 2.

B. Experimental Set-up

The experimental validation was carried out as follows. First, we selected only the tweet content and the label from both the *Airline* and the *GlobalWarming* data repositories. Afterward the three word embedding methods presented in

[2]Kaggle: www.kaggle.com
[3]UCI repository: http://archive.ics.uci.edu/ml/
[4]https://www.kaggle.com/crowdflower/twitter-airline-sentiment
[5]https://data.world/crowdflower/sentiment-of-climate-change

TABLE I
SUMMARY OF THE CHARACTERISTICS OF THE DATASETS USED FOR EXPERIMENTATION, DERIVED FROM THE *Airline* (AIR) AND *Global Warming* (GW) REPOSITORIES.

Name	No. Features	No. Samples	No. Subclasses (0\1\2)	Samples per class (0/1/2)	Samples per subclass 0	1	2
Air_Original	300 (Glove) 3720 (CWR-LSTM) 13634 (TF-IDF)	14640	-	9178\3099\2363	-	-	-
Air_GloVe_kmeans	300	17387	2\0\0	9178\4100\4100	5078\4100	4100	4100
Air_GloVe_DBSCAN	300	53436	3\11\7	9178\23914\15218	9140\26\12	2174 per subclass	2174 per subclass
Air_CWR-LSTM_kmeans	3720	20726	2\0\0	9178\5774\5774	5774\3404	5774	5774
Air_CWR-LSTM_DBSCAN	3720	27400	3\7\3	9178\9312\3510	8652\513\13	2292\1170\1170\1170 1170\1170\1170	1170 per subclass
Air_TF-IDF_kmeans	13634	22616	2\0\0	9178\6719\6719	2459\6719	6719	6719
Air_TF-IDF_DBSCAN	13634	39438	1\4\9	9178\9875\20385	9178	3080\2265\2265\2265	2265 per class
GW_Original	300 (Glove) 3720 (CWR-LSTM) 12112 (TF-IDF)	6090	-	1117\1862\3111	-	-	-
GW_GloVe_kmeans	300	6618	0\0\2	1645\1862\3111	1645	1862	1645\1466
GW_GloVe_DBSCAN	300	12692	2\6\8	2218\7363\3111	1109 per subclass	1818\1109\1109 1109\1109\1109	3044\14\8 7\7\9\14\18
GW_CWR-LSTM_kmeans	3720	6618	0\0\2	1645\1862\3111	1645	1862	1645\1466
GW_CWR-LSTM_DBSCAN	3720	9105	5\8\7	2048\3947\3111	1035\336 336\336\5	1594\336\336\336 336\336\336\336	2732\336 8\9\7\7\12
GW_TF-IDF_kmeans	12112	6695	0\0\2	1722\1862\3111	1722	1862	1722\1389
GW_TF-IDF_DBSCAN	12112	18118	3\10\11	3306\11701\3111	1102 per class	1783\1102\1102\1102\1102 1102\1102\1102\1102\1102	3004\10\8\14\8 14\8\7\7\7\9\7\8

Section II-A were implemented: 1) GloVe, which yielded 300 features on both data repositories, 2) CWR-LSTM with 3720 features on both repositories and 3) TF-IDF with 13634 features for the *Airline* data repository and 12112 for the *GlobalWarming* one. Besides, for each of these six newly created datasets, we applied the two variants of the CDSMOTE method presented in this paper (i.e. *CDSMOTE-kmeans* and *CDSMOTE-DBSCAN*). For *CDSMOTE-kmeans*, a k value of 2 was selected. This value was chosen empirically and showed better results. This is also consistent with the results reported in previous work [10]. Recall that in this case, only the majority class is decomposed. Moreover, for *CDSMOTE-DBSCAN*, a maximum distance between two samples threshold set to $eps = 0.5$. DBSCAN automatically yielded a different number of clusters for each of the three classes.

Table I summarises the datasets presented in the experimental validation. For instance, the rows with the indexes *Air_Original* and *GW_Original* describe the initial versions of the *Airline* and *Global Warming* data repositories respectively. As mentioned before, the number of features extracted (second column) depend on the extraction method used. The third column shows the number of total samples (i.e. classes 0, 1, and 2) for the repository. Since these rows describe the original repositories, there are no subclasses for any of the main classes (thus columns 6 to 8 are also empty). Still, in the fifth column, we show the sample distribution for each class, which was mentioned in the previous section.

In contrast, the remaining rows show examples where either *CDSMOTE-kmeans* (with the kmeans suffix) or *CDSMOTE-DBSCAN* (with the *DBSCAN* suffix) was implemented. In this case, we have separated the datasets also by feature extractor used, thus yielding six datasets per initial repository, as explained before. In this case, we also show the number of features (second column), the number of samples after the oversampling has taken place (third column), the number of subclasses found for each main class by the clustering algorithm (fourth column), the number of samples in each class (fifth column) and finally, the distribution of all of those samples within the subclasses (columns 6 to 8). For example, *Air_GloVe_kmeans* is the dataset derived from extracting 300 GloVe features to the *Airline* data repository. After class decomposition using *k-means*, class 0 was clustered in two subclasses, and classes 1 and 2 were not clustered. After SMOTE, classes 1 and 2 increased in size (now with 4100 samples per class), and the distribution of these new samples within the subclasses can be seen in the last three columns. Most notably, the sixth column shows that the 9178 samples of class 0 have been split in a way that 5078 are grouped in the first sub-class, and the remaining 4100 on the second sub-class. Notice that in the cases where *CDSMOTE-DBSCAN* is applied, it is not always the majority class the one in which more clusters are obtained. This also leads to the *CDSMOTE-DBSCAN* method to perform more data augmentation than that of the *k-means* variant.

To compare the classification accuracy for the different datasets, we used two of the most popular classifiers used in related literature, i.e. Support Vector Machine (SVM) with a Gaussian kernel and Random Forests (RF). Since we are interested in evaluating the performance of the different datasets rather than the classifiers themselves, we used these with no parameter optimisation.

All code was implemented using the *sklearn* library in Python 3.7 on a Windows 10 Machine. The source code and a demo notebook can be found here[6].

C. Results & Discussion

Tables II and III present the Precision, Recall and F1-score obtained when classifying the datasets using SVM and RF respectively. The highest values obtained for the three data variants (i.e. Baseline, *CDSMOTE-kmeans* denoted as kmeans and *CDSMOTE-DBSCAN* denoted as *DBSCAN*) combined

[6]https://github.com/carlosfmorenog/CDSMOTE-NLP-NONBIN

with the word embedding methods (i.e. GloVe, CWR-LSTM and TF-IDF) of the two data repositories (i.e. Airline and Global Warming) are marked in italics. Besides, the best values obtained for each data repository are highlighted in bold.

Notice that for the SVM classification presented in Table II, the best performance is always obtained for the *CDSMOTE-DBSCAN* datasets. Almost the same applies when RF is used, as shown in Table III, except for the Airline repository with TF-IDF features (where *CDSMOTE-kmeans* with $k = 2$ yields vastly better results), and the recall of the Global Warming repository with CWR-LSTM features; in this case by a small margin. These results confirm that, as expected, *DBSCAN* is in most cases a more suitable method to find clusters between the features extracted for these text repositories, due to the distance used to calculate the centroids.

Finally, it is also interesting to observe the effect on the obtained results based on the number of features extracted. When classifying using SVM, for the Airline data repository, the CWR-LSTM feature extraction method yielded arguably the best results, despite extracting around four times fewer features than TF-IDF. In contrast, in the Global Warming data repository, it is the TF-IDF feature extractor the one that yielded the best results with the same ratio of features extracted compared to CWR-LSTM. When classifying using RF, results for the Airline data repository are superior when using the feature extractor that obtains the least amount of features (i.e. GloVe); however, for the Global Warming data repository, it is TF-IDF, the extractor obtaining the largest number of features, which yields the best result (except for the recall, in which GloVe is marginally better). For all cases, SVM appears to yield better classification results compared to RF.

TABLE II
PRECISION, RECALL AND F1-SCORE OBTAINED WHEN CLASSIFYING THE DATASETS WITH SVM

repo		Air			GW		
features		GloVe	CWR-LSTM	TF-IDF	GloVe	CWR-LSTM	TF-IDF
Original	prec	0.716	0.894	0.792	0.623	0.828	0.668
	rec	0.735	0.896	0.799	0.629	0.827	0.671
	F1	0.705	0.894	0.791	0.617	0.825	0.662
kmeans	prec	0.691	0.899	0.858	0.608	0.823	0.704
	rec	0.696	0.899	0.857	0.608	0.822	0.708
	F1	0.692	0.899	0.858	0.605	0.821	0.704
DBSCAN	prec	*0.817*	**0.935**	*0.93*	*0.838*	*0.879*	**0.9**
	rec	*0.806*	**0.935**	*0.926*	*0.826*	*0.876*	**0.891**
	F1	*0.807*	**0.935**	*0.927*	*0.829*	*0.877*	**0.894**

TABLE III
PRECISION, RECALL AND F1-SCORE OBTAINED WHEN CLASSIFYING THE DATASETS WITH RF

repo		Air			GW		
features		GloVe	CWR-LSTM	TF-IDF	GloVe	CWR-LSTM	TF-IDF
Original	prec	0.541	0.419	0.392	0.436	0.552	0.261
	rec	0.647	0.628	0.627	0.512	0.512	0.511
	F1	0.526	0.485	0.483	0.349	0.349	0.345
kmeans	prec	0.472	0.469	*0.685*	0.468	0.35	0.571
	rec	0.599	0.496	*0.685*	0.462	*0.474*	0.434
	F1	0.502	0.443	*0.685*	0.409	0.327	0.388
DBSCAN	prec	**0.817**	*0.636*	0.206	*0.846*	*0.783*	**0.877**
	rec	**0.729**	*0.658*	0.259	**0.595**	0.444	*0.576*
	F1	**0.737**	*0.605*	0.134	*0.613*	*0.366*	**0.615**

IV. CONCLUSION

In this paper, we presented a new approach to handle class-imbalance in text-based datasets utilizing class-decomposition. Using two different datasets from the public domain for predicting sentiments within the text, we showed that using k-means and DBSCAN to re-engineer the datasets and find within-class similarities improves the performance even in the presence of the class-imbalance. Unlike other data-sampling methods, our method does not cause any information loss. We do not remove any instance from the majority class, instead, by using unsupervised methods such as kmeans or DBSCAN, we show that the dominance of the majority class-instances can be reduced and hence, improve the visibility of the minority class (class of interest). Future work will focus on the utilization of other clustering methods and optimizing the parameters which derive the numbers of clusters for each class. Possible future directions can also explore other application areas such as medical images, where unequal distributions of classes are common.

REFERENCES

[1] Q. T. Ain, M. Ali, A. Riaz, A. Noureen, M. Kamran, B. Hayat, and A. Rehman, "Sentiment analysis using deep learning techniques: A review," *International Journal of Advanced Computer Science and Applications*, vol. 8, 2017.

[2] T. Rao and S. Srivastava, "Analyzing stock market movements using twitter sentiment analysis," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ser. ASONAM '12. USA: IEEE Computer Society, 2012, p. 119–123.

[3] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale amazon product reviews," in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, 2018, pp. 1–6.

[4] Z. Singla, S. Randhawa, and S. Jain, "Statistical and sentiment analysis of consumer product reviews," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2017, pp. 1–6.

[5] E. Kušen and M. Strembeck, "Politics, sentiments, and misinformation: An analysis of the twitter discussion on the 2016 austrian presidential elections," *Online Social Networks and Media*, vol. 5, pp. 37 – 50, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2468696417301088

[6] P. Vuttipittayamongkol, E. Elyan, and A. Petrovski, "On the class overlap problem in imbalanced data classification," *Knowledge-Based Systems*, vol. 212, p. 106631, 2021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705120307607

[7] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220 – 239, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416307175

[8] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "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)*, vol. 42, no. 4, pp. 463–484, 2012.

[9] J. Prusa, T. M. Khoshgoftaar, D. J. Dittman, and A. Napolitano, "Using random undersampling to alleviate class imbalance on tweet sentiment data," in *2015 IEEE International Conference on Information Reuse and Integration*, 2015, pp. 197–202.

[10] E. Elyan, C. Moreno-García, and C. Jayne, "Cdsmote: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification," *Neural Computing and Applications*, Jul 2020.

[11] P. Vuttipittayamongkol and E. Elyan, "Neighbourhood-based undersampling approach for handling imbalanced and overlapped data," *Information Sciences*, vol. 509, pp. 47 – 70, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025519308114

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[13] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113 – 141, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025513005124

[14] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *Journal of artificial intelligence research*, vol. 19, pp. 315–354, 2003.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[16] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with dte-sbd: Decision tree ensemble based on smote and bagging with differentiated sampling rates," *Information Sciences*, vol. 425, pp. 76–91, 2018, cited By 103. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85042332829&doi=10.1016%2fj.ins.2017.10.017&partnerID=40&md5=6cd752a20a6505030c067df5d29a4d9f

[17] M. Ijaz, G. Alfian, M. Syafrudin, and J. Rhee, "Hybrid prediction model for type 2 diabetes and hypertension using dbscan-based outlier detection, synthetic minority over sampling technique (smote), and random forest," *Applied Sciences*, vol. 8, no. 8, p. 1325, 2018.

[18] S. Wang, D. Wang, J. Li, T. Huang, and Y.-D. Cai, "Identification and analysis of the cleavage site in a signal peptide using smote, dagging, and feature selection methods," *Molecular omics*, vol. 14, no. 1, pp. 64–73, 2018.

[19] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Dbsmote: density-based synthetic minority over-sampling technique," *Applied Intelligence*, vol. 36, no. 3, pp. 664–684, 2012.

[20] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "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*. Springer, 2009, pp. 475–482.

[21] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang, "Ni-mwmote: An improving noise-immunity majority weighted minority oversampling technique for imbalanced classification problems," *Expert Systems with Applications*, p. 113504, 2020.

[22] E. Elyan and M. M. Gaber, "A genetic algorithm approach to optimising random forests applied to class engineered data," *Information Sciences*, vol. 384, no. Supplement C, pp. 220 – 234, 2017.

[23] E. Elyan and M. M. Gaber, "A fine-grained random forests using class decomposition: an application to medical diagnosis," *Neural Computing and Applications*, vol. 27, no. 8, pp. 2279–2288, Nov 2016.

[24] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[25] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://www.aclweb.org/anthology/D14-1162

[26] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning generic context embedding with bidirectional LSTM," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 51–61. [Online]. Available: https://www.aclweb.org/anthology/K16-1006

[27] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365

[28] J. Singh and V. Gupta, "Text stemming: Approaches, applications, and challenges," *ACM Comput. Surv.*, vol. 49, no. 3, Sep. 2016. [Online]. Available: https://doi.org/10.1145/2975608

[29] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[30] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[31] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.

[32] Y. Zhang, *Support Vector Machine Classification Algorithm and Its Application*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 179–186.