VUTTIPITTAYAMONGKOL, P., ELYAN, E. and PETROVSKI, A. 2021. On the class overlap problem in imbalanced data classification. *Knowledge-based systems* [online], 212, article number 106631. Available from: https://doi.org/10.1016/j.knosys.2020.106631

On the class overlap problem in imbalanced data classification.

VUTTIPITTAYAMONGKOL, P., ELYAN, E. and PETROVSKI, A.

2021

On the Class Overlap Problem in Imbalanced Data Classification

Pattaramon Vuttipittayamongkol*, Eyad Elyan, Andrei Petrovski

School of Computing Sciences and Digital Media, Robert Gordon University, UK

Abstract

Class imbalance is an active research area in the machine learning community. However, existing and recent literature showed that class overlap had a higher negative impact on the performance of learning algorithms. This paper provides detailed critical discussion and objective evaluation of class overlap in the context of imbalanced data and its impact on classification accuracy. First, we present a thorough experimental comparison of class overlap and class imbalance. Unlike previous work, our experiment was carried out on the full scale of class overlap and an extreme range of class imbalance degrees. Second, we provide an in-depth critical technical review of existing approaches to handle imbalanced datasets. Existing solutions from selective literature are critically reviewed and categorised as class distribution-based and class overlap-based methods. Emerging techniques and the latest development in this area are also discussed in detail. Experimental results in this paper are consistent with existing literature and show clearly that the performance of the learning algorithm deteriorates across varying degrees of class overlap whereas class imbalance does not always have an effect. The review emphasises the need for further research towards handling class overlap in imbalanced datasets to effectively improve learning algorithms' performance.

Keywords: Imbalanced data, Class overlap, Classification, Evaluation metric, Benchmark

*Corresponding author

Email addresses: p.vuttipittayamongkol@rgu.ac.uk (Pattaramon Vuttipittayamongkol), e.elyan@rgu.ac.uk (Eyad Elyan), a.petrovski@rgu.ac.uk (Andrei Petrovski)

1. Introduction

Learning from datasets with skewed class distributions remains a challenge in machine learning. such datasets are realised as imbalanced datasets and widely seen in many applications, for example, anomaly detection [1], medical prediction [2, 3], object recognition [4, 5] and business management [6]. In these domains, the minority class is usually the class of interest and has a higher misclassification cost than the majority class. Standard learning algorithms generally build classification models based upon the maximum accuracy, which often leads to biased classification towards the majority class and misclassification of minority class instances [7, 8]. However, such failure in classification of imbalanced datasets is not always caused by class imbalance solely. In fact, a linearly separable dataset can be perfectly classified by a typical classification algorithm no matter how skewed the class distribution is [9]. On the contrary, when class overlap is present, even a balanced dataset can be difficult for a learning task.

When dealing with classification of imbalanced data, rebalancing class distribution is among the most common approaches that researchers consider. Many traditional and recent resampling methods only aim at getting a more balanced version of the training data and do not factor in the problem of class overlap [10, 11, 12]. Some methods deal with instances in the overlapping region, especially those near the borderline areas; however, their resampling rates are controlled by the degree of class imbalance [13, 14]. Thus, in some scenarios, results can be highly influenced by class imbalance rather than class overlap. For instance, when a dataset suffers from high class overlap but its classes are slightly imbalanced, insufficient resampling may result in class overlap not being properly addressed. On the other hand, with low class overlap and high class imbalance, excessive resampling may occur.

The impacts of class imbalance, class overlap and other characteristics such as small disjunct and dataset size have been investigated [15, 16, 17, 18]. Class overlap frequently shows the highest negative influence among potential factors including class imbalance [17, 8]. This raises some important questions in handling classification of imbalanced datasets: 1) Are the solutions that mainly aim to rebalance the class distribution sufficiently effective? 2) Should the problem of class overlap be the main concern in developing new algorithms?

Although several reviews on the problem of imbalanced data in classification exist [8, 19, 20], the problem of class overlap in imbalanced data was not emphasised as the main issue and the discussions often lacked a support of sufficient experimental evidence. Das et al. [8] proposed that the two key challenges for standard learning algorithms are class imbalance and class overlap. Possible nature of learning outcomes in different scenarios of class imbalance and class overlap based on the dataset size were suggested; however, no experimental evidence was given. The authors also investigated other data irregularities such as small disjunct and missing features; thus, the discussion on the class overlap problem was limited. In [19], merely a brief description of other studies on the effect of class overlap in relation to class imbalance was given. The authors paid particular attention to the discussion of different techniques used in existing methods for handling imbalanced classification. Stefanowski [20] motivated the research community to develop new algorithms for imbalanced data that realise data factors, which included overlapping between classes. The author presented the analyses on characteristics of the minority class, which was divided into sub regions of safe, borderline, rare and outlier samples. This was studied along with the behaviours of different learning algorithms; however, this cannot yet be mathematically verified on real world datasets. Like in many other reviews [21, 22], Kaur et al. [23] conducted a comparative analysis of methods, which was mainly organised as data preprocessing and algorithmic approaches, and the problem of class overlap was barely discussed. Some other reviews focused on the issue of imbalanced data classification in specific contexts such as big data [24, 25], multi-class problem [24, 26] and neural networks [27, 28]. These clearly show that there is still a gap in the study of class overlap in the context of class imbalance.

In this paper, the importance of handling class overlap in imbalanced data classification is investigated. This was carried out through an extensive experiment and a critical review of solutions to imbalanced learning. The experiment provides an objective measurement of the impact of class overlap versus the impact of class imbalance. Unlike in previous studies [15, 16, 17, 18], which were based on limited ranges of class imbalance and class overlap degrees, we carried out a full-scale experiment using over 1,000 synthetic datasets. The in-depth review of existing solutions to classification of imbalanced datasets is presented in an alternative perspective rather than data and algorithm levels, which was commonly arranged in other review papers [19, 8, 29, 30, 7]. We considered the main objective of the solutions and categorised them into class distribution-based and class overlap-based approaches for better comparing and contrasting the two approaches. Class distribution-based methods mainly concern and aim to suppress the problem of imbalanced class distribution. Class overlap-based methods focus on improving the visibility of instances, especially positive instances, in the overlapping region. In addition, recent and emerging methods that do not particularly deal with the class imbalance or class overlap problems are also discussed. These include, for example, the use of one of the latest techniques in machine learning, Generative Adversarial Networks (GANs)[31, 32].

The main contributions of this review are listed below.

1. A technical discussion with advantages and disadvantages of evaluation metrics including how some of them can be misleading in certain imbalanced contexts
2. An extensive experiment illustrating the scales of impact of class overlap and class imbalance on imbalanced dataset classification
3. A critical discussion of methods and literature selected from leading peer-reviewed publications in the perspective of class overlap-based and class distribution-based approaches, as well as recent emerging technologies
4. An overview of benchmarking methods in the literature showing commonly-used ones that can be considered as good standards, but at the same time

suggesting a need for comparing against recent and state-of-the-art methods for more convincing and reliable evaluation

The remainder of this paper is organised as follows. In Section 2, we give the definitions of class imbalance and class overlap. Section 3 contains an in-depth discussion of evaluation metrics used in imbalanced learning. Section 4 provides the experimental results and discussion on the effects of class imbalance and class overlap on the learner's performance in an extensive range of scenarios. In section 5, we critically review existing approaches for handling classification of imbalanced datasets. Finally, the conclusion is delivered in Section 6.

2. Key Definitions

2.1. Class imbalance

An imbalanced dataset is a dataset with an unequal distribution of classes. This is depicted in Figure 1, where majority and minority class instances are represented by circles and triangles, respectively. In machine learning, class imbalance becomes an issue when the minority class is significantly smaller in size and is the primary class of interest with a relatively high misclassification cost. Thus, in a binary-class problem, the minority class is also realised as the positive class whereas the majority class is the negative class.

The degree of class imbalance can be measured as the imbalance ratio (imb) as expressed in Eq. 1 or the percentage of the minority class ($\%minority$) as shown in Eq.2, where M and m are the numbers of instances in the majority class and minority class, respectively.

$$imb = \frac{M}{m}, \tag{1}$$

$$minority(\%) = \frac{m}{M} * 100. \tag{2}$$

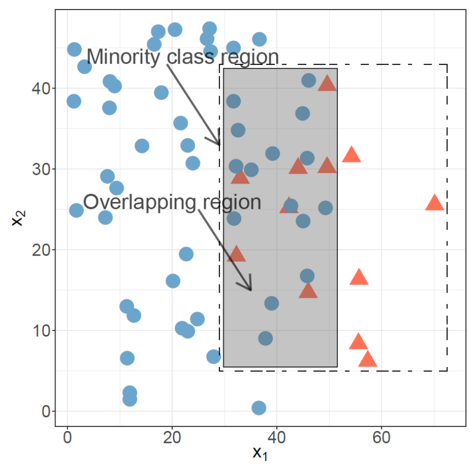

Figure 1: An illustration of an imbalanced and overlapped dataset. Reprinted with permission from ref. [33]. Copyright 2020, Elsevier

2.2. Class overlap

Class overlap occurs when instances of more than one class share a common region in the data space. These instances have similarities in feature values although they belong to different classes, and such a complication is a substantial obstacle in classification tasks. The class overlap problem becomes more serious when class imbalance is also present in the data, and vice versa [16]. In an imbalanced and overlapped dataset, the negative class is normally as well the dominant class in the overlapping region. As a result, negative instances are more frequently and clearly present to the learner than positive instances in such a region. This means that the decision boundary tends to shift towards the negative class leading to misclassification of positive instances near the class boundary, which is undesired in real-world problems.

Since class overlap has not been mathematically well characterised [34], a standard measurement of the overlap degree is not yet defined. Several approaches have been formulated to estimate the overlap degree, however, with limitations. For example, in [16], the overlap degree of a synthetic dataset was determined from the overlapping area with respect to the the total data space. In [33], the authors adapted such measurement so that class imbalance was also taken into account seeing that the minority class is relatively more overwhelmed by class overlap. The overlap degree was instead measured from the overlapping area with respect to the the total area of the positive class. Another common approach is using the classification error as the estimated overlap degree, e.g., the percentage of instances misclassified by the k-Nearest Neighbor rule [35] (kNN) with respect to the number of total instances [36, 37]. However, in [34], the authors showed that such an approach was inaccurate and proposed a use of the ridge curves of the probabilistic density function to quantify class overlap. The computation was based on the ratio of the saddle point to a smaller peak of the ridge curves of the two classes. This method is one of a few existing methods that measure overlap from the actual contour of data and can be extended to handle multi-class datasets. The main drawback of this approach is that it is only applicable to datasets with normal distributions of both data and features, which is impracticable to real-world datasets. In [38], the overlap degree was defined as the distance between the class centroids, which is likely to be inaccurate due to arbitrary shapes and non-uniformity of data in nature. Another approach [39] was based on Support Vector Data Description (SVDD) [40]. SVDD was used to locate approximated boundaries of each class in binary-class datasets, and the overlapping region was estimated based on the amount of the common instances found within both boundaries. Similar to the approach of [38], this method tends to introduce high errors in the overlap approximation since SVDD is only capable of discovering a spherical-shaped boundary of a class, which is not ideal for real-world datasets.

For our experiment discussed in Section 4, we follow the measure of class overlap proposed in [33] (Eq. 3). Figure 1 illustrates how the regions in the equation are approximated.

$$overlap(\%) = \frac{overlapping\ area}{minority\ class\ area} * 100 \tag{3}$$

3. Evaluation Metrics

Some typical evaluation metrics for classification are not affected by skewed class distributions while others can be misleading with biases towards the majority class. Common metrics for classification of imbalanced datasets such as sensitivity, specificity, balanced accuracy, G-mean, AUC and F1-score will be discussed in detail. For other assessment measures, the reader may refer to [41, 42, 43, 44].

In imbalanced problems, accurate detection of minority class instances is crucial. This is usually evaluated in terms of sensitivity, which is also known as true positive rate (TPR) or recall. The metric is formulated as in Eq. 4, where TP and FN denote true positive and false negative, respectively. As sensitivity only reflects the performance over one class, it is often reported in conjunction with another metric, such as specificity (i.e. true negative rate – Eq. 5, where TN and FP denote true negative and false positive, respectively), balanced accuracy, G-mean and AUC, to also explore the overall performance or the trade-off between the classes [45, 46, 47].

$$sensitivity = \frac{TP}{TP + FN} \tag{4}$$

$$specificity = \frac{TN}{TN + FP} \tag{5}$$

Balanced accuracy is the arithmetic mean of the accuracy over each class (Eq. 6) [45, 48, 49, 22]. It is also referred to as balanced mean accuracy [50], average accuracy [51, 11, 52], macro-accuracy [53], etc. The traditional accuracy (Eq. 7) can be significantly misleading when class imbalance is high and the negative class accuracies (TN and $TN + FP$) are highly dominant. For instance, a perfectly classified negative class of 1000 instances with an entirely misclassified positive class with 10 instances result in over 99% accuracy, which could be misleading as a good classification model. In fact, this same case yields 50% balanced accuracy, which more reflects the actual performance of the model. Thus, balanced accuracy often replaces the traditional accuracy, and it is among the most common measures used for imbalanced problems [54].

$$balanced\ accuracy = \frac{sensitivity + specificity}{2} \tag{6}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{7}$$

Another metric for evaluating the overall performance is G-mean [55]. It is the geometric mean of sensitivity and specificity (Eq. 8). Since both G-mean and balanced accuracy give the average values of sensitivity and specificity, they are often used interchangeably. Based on the literature reviewed in this paper, G-mean was more frequently used. This could be attributed to the fact that G-mean is also a widely-known metric for datasets with non-skewed class distributions whereas balanced accuracy roughly reduces to the traditional overall accuracy in such scenarios.

$$G - mean = \sqrt{specificity * sensitivity} \tag{8}$$

$$AM - GM\ inequality: \quad \frac{x + y}{2} \geq \sqrt{xy} \tag{9}$$

$$balanced\ accuracy \geq G - mean \tag{10}$$

According to the Arithmetic Mean-Geometric Mean Inequality theory (AM-GM inequality) (9), it can be said that balanced accuracy is always greater than or equal to G-mean (10). The equality holds when sensitivity and specificity are equal. For further analysis, consider Fig. 2, which presents values of G-mean and balanced accuracy across varying scenarios in terms of the difference between sensitivity and specificity values. On the x-axis, all possible combinations of sensitivity and specificity at a step of 10% are shown. It can be seen that the difference between G-mean and balanced accuracy becomes greater when the difference between sensitivity and specificity increases. This is due to the fact that the geometric mean is affected more by the lower value.

Here are some examples to illustrate the differences in G-mean and balanced accuracy in difference scenarios. In Fig. 2, at $specificity = 90\%$ and $sensitivity = 60\%$, G-mean is 73.48% and balanced accuracy is 75%. The difference between G-mean and balanced accuracy in this case is not significant. In an extreme case where $specificity = 100\%$ and $sensitivity = 10\%$, the resulting G-mean is 31.62% while balanced accuracy is 55%. It is clearly seen that G-mean is affected more by sensitivity. For another extreme case when there is zero accuracy of any class, G-mean$= 0$. This suggests that G-mean is able to reflect these unfavourable scenarios where balanced accuracy only provides average values. Thus, to determine a more suitable metric between G-mean and balanced accuracy, the user will need to carefully make a selection based on the application domain and the main objective of the classification task.

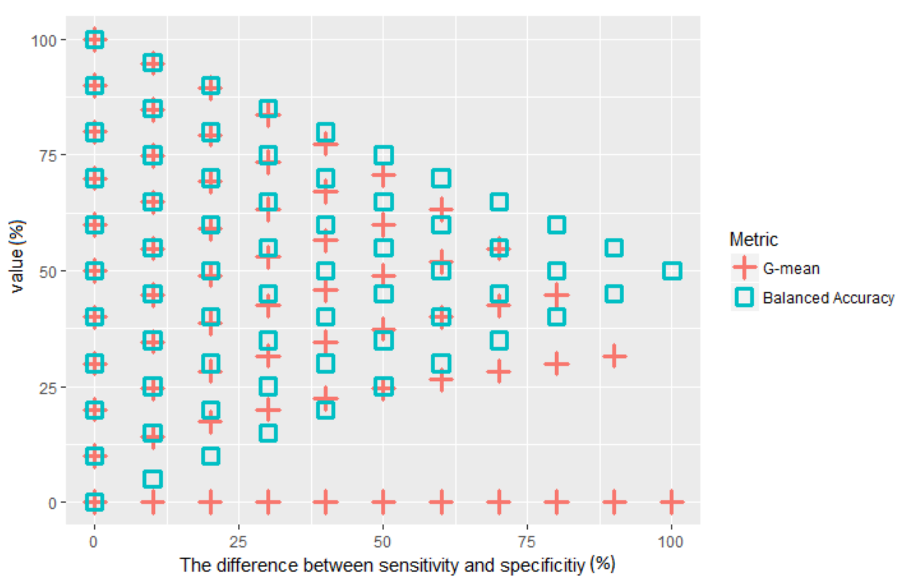

Figure 2: Variants of the two means

Another common metric, F1-score, is the harmonic mean of sensitivity and precision as expressed in Eq.11. It is also a widely-used metric for imbalanced problems [37, 53, 56]. However, unlike G-mean and balanced accuracy, F1-score takes into account of precision instead of specificity. As shown in Eq. 12, precision is dependent of FP and TP. Since FP and TP are not normalised with respect to the class size, FP can be excessively higher than TP in an extremely imbalanced case. This high FP can be deceptive when in fact the true false positive rate (FPR) is insignificant. In such case, precision is strongly influenced by FP and does not reflect the actual performance on the positive class. As a consequence, F1-score will be misleading.

$$F1 - score = \frac{2}{\frac{1}{sensitivity} + \frac{1}{precision}}, \tag{11}$$

$$precision = \frac{TP}{TP + FP} \tag{12}$$

To demonstrate such an issue, consider an example of a dataset with 10:1000 positive to negative class instances and the classification result of 10 true positives and 10 false positives. This indicates 100% sensitivity and 1% FPR, which is generally highly desirable. Yet, the precision turns out to be 50% leading to a 67% F1-score, which very much underestimates and deviates from the actual performance.

It is also worth pointing out is that using F1-score alone may not be sufficient to compare models. In other words, any two models that yield similar FP, TP and sensitivity, will have similar F1-score regardless of their difference in FPRs. Consider an example of two models predicting on datasets with 10:100 and 10:10000 positive to negative class instances where the models achieve 10% and 0.1% FPR, respectively. Given the same sensitivity gained, the models have the same value of F1-score accordingly, which is 67%. In fact, the former case with 10% FPR is less favourable than the latter case with 0.1% FPR, but F1-score does not convey that. Thus, the use of F1-score alone may not be sufficient to indicate the quality of a classification model in imbalanced domains. Yet, it can be meaningful when carefully considered along with other measures.

Another commonly-used metric is the area under the receiver operating characteristic curve (AUC). A receiver operating characteristic curve (ROC) visualises the values of TPR against FPR at varying probability thresholds. AUC gives a summary of the ROC curve as a single value. AUC is often used to compare the performance among classifiers; however, there have been some arguments raised against its usage [57]. Firstly, ROC curves are useful when misclassification costs and class distributions are not specified [44]; so is AUC[58]. This suggests that ROC and AUC can be used for inspecting and summarising the general performance of a classifier. However, in a real-life application, the error costs are known and a model can be fine-tuned for the optimal results, which eventually falls onto a single point on the RUC curve. Thus, a classifier with a higher AUC does not necessarily give a better result. This leads to the second argument that visual inspection of ROC curves should be carried out instead of considering only AUC values [57]. However, often there is no clear winning between the two ROC curves making it difficult to compare [58]. Last but foremost, AUC weights the positive and negative class errors equally while in many application domains, misclassification costs are unequal. In this case, summarising over all possible threshold values is unconvincing [59].

In summary, it is recommended that for evaluation of imbalanced dataset classification, individual class accuracy, especially sensitivity, is considered along with an overall performance measure such as balanced accuracy or G-mean. F1-score and AUC can also be assessed; however, they should be carefully discussed due to the constraints addressed above.

4. Impacts of Class Overlap vs Class Imbalance

When handling classification of imbalanced data, rebalancing the class distribution is often an approach that researchers take. However, it should also be realised that class overlap is another common issue in classification tasks, which becomes more serious when it occurs in an imbalanced context. Many traditional and recent resampling methods for handling imbalanced datasets only aim at making the class distribution balanced and do not factor in the problem of class overlap [10, 11, 12]. On the other hand, some resampling methods deal with instances in the overlapping region, especially those near the borderline areas, without concerning the resulting class distribution [33, 56, 60]. There also exist methods that address both of the class overlap and class imbalance problems [13, 14]. In the last type of methods, problematic instances in the overlapping region are resampled until the class distribution becomes balance. This means that the problem of class overlap is handled according to the class imbalance degree and regardless of the class overlap degree. As a result, insufficient resampling may occur when class imbalance is low. On the contrary, when class imbalance is high, these methods may lead to excessive resampling. All of these approaches have shown their potentials in improving classification results in different ways. Countless variations of existing methods make it impossible to compare and find out which approach is better. Instead, we can consider the scale of effect of class imbalance in comparison to class overlap. This will advise which of the problems should be more concerned.

Previous literature suggested that class overlap had a higher negative effect on the learner's performance than class imbalance [16, 17, 18]. In [17], the authors showed that imbalanced datasets with no presence of class overlap could be perfectly classified using fuzzy sets. Moreover, when the class overlap degree was low, class imbalance had no significant effects on the classification results. It has to be pointed out that these observations were based on the maximum overlap degree of 64% (see [17] for their measurement of the overlap degree) although a wide range of class imbalance degrees was used in the experiment. Similar findigns were reported in [18] when using decision tree (DT), rule based and k-NN classifiers. Interestingly, the authors of [61] showed that when sufficient training data was available, class imbalance did not degrade the performance of support vector machine (SVM). Unlike in the case of class overlap, SVM performance significantly degraded. These experiments however were carried out with only a few variations of class imbalance and class overlap. For instance, in [16], some datasets were simulated such that the positive class became dominant in the overlapping region. This created inconsistencies in the overall imbalance degree

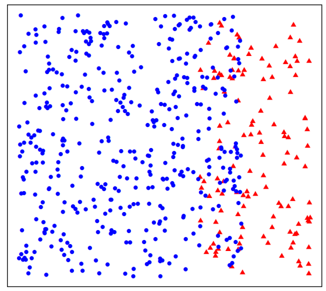

Figure 3: An example of a synthetic dataset with 40%overlap and 30%minority

and the imbalance situation in the overlapping region, which led to inconclusive results. To establish these results at the full scale of class overlap with a wide range of class imbalance, including extreme cases, we have carried out a thorough experiment detailed as follows.

4.1. Experimental Setup

4.1.1. Datasets

To enable the study at the full spectrum of class overlap, synthetic datasets were used. A total of 1,010 binary datasets were created to cover all possible combinations of 101 class overlap degrees and 10 class imbalance degrees. The overlap degrees (%overlap) ranged from 0% to 100% with a step of 1. The percentage of the minority class with regard to the majority class (%minority), as defined in Eq. 2, ranged from 10% to 100% with a step of 10. In each dataset, there were 1,000 majority class instances and the number of minority class instances was based on the imbalance degree. Datasets were uniformly distributed, which means that the data densities of the majority class and the minority class were equal. The rationale behind this was two-folded. First, there was no class imbalance in the overlapping region to ensure that each of the components, e.g. class imbalance and class overlap, was solely investigated with no interfering effect of the other. Second, there would be no effects on the learning algorithm caused by differences in the data density. An example of such synthetic datasets is illustrated in Figure 3.

4.1.2. Methodology

Random Forests (RF) was chosen as the learning algorithm for the following reasons. First, it is a representative of standard learning algorithms that aim at maximising the overall classification accuracy such as DT, SVM, neural net, naive bayes, etc. Without an appropriate adjustment, these learning algorithms tend to be influenced more by the dominating class, which will result in biased classification. Second, RF is robust to overfitting [62], which helps minimise the effect of different sample sizes. Lastly, though RF is one of the most widely-used learning algorithms for classification of imbalanced datasets [22], it was not experimented in previous studies [16, 17, 18].

The default parameter settings of RF in *caret*[1] package in *R* were used. That is the number of trees was set to 500. The number of variables used at each split was $\sqrt{n}$, and the training sample size for each tree was $0.632 * n$, where n is the number of the total instances in the dataset. The datasets were partitioned into training and testing sets at the ratio of 80 to 20, and 10-fold cross validation was applied for model selection. The resulting models were evaluated in terms of sensitivity, specificity, balanced accuracy, and AUC.

4.1.3. Results and discussion

Classification results are shown in Figure 4. It can be seen that both class imbalance and class overlap caused degradation in sensitivity. However, changes in the imbalance degree barely showed an effect on sensitivity when imbalance and overlap were not very high. This is evidenced at %minority = 70 to 100, where it is not apparent that sensitivity values were impacted by a change in class imbalance when %overlap = 0 to 50. For example, at %overlap = 50, all the plots of %minority = 70, 80, 90 and 100 showed approximately 70% sensitivity. At higher imbalance degrees, the drops in sensitivity due to class imbalance became more clearly visible. At %overlap = 0 to 25, there were no apparent differences in sensitivity at different imbalance degrees, except at high imbalance,

[1] https://CRAN.R-project.org/package=caret

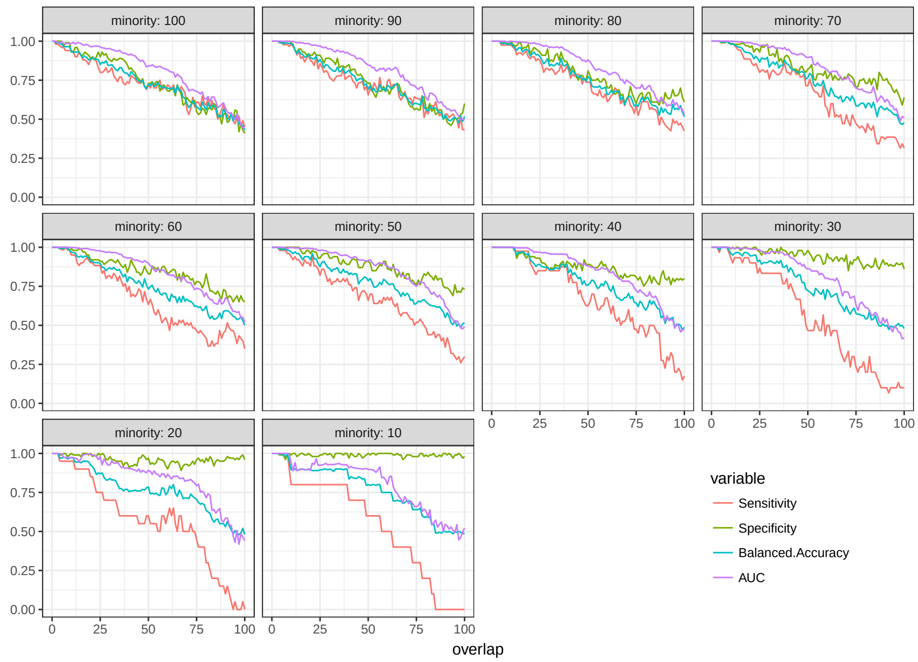

Figure 4: Classification results corresponding to various degrees of class imbalance and class overlap. The number on the top of each plot indicates the percentage of the minority class with respect to the majority class.

i.e %minority = 10 and 20. This can be attributed to the effect of sample size. That is, at lower imbalance degrees, the sample sizes were larger, which might be sufficient to suppress the effect of class imbalance [61, 18]. Finally, the effect of class imbalance was more obvious when the overlap degree was higher. These results suggest that the impact of class imbalance on sensitivity highly depended on the level of class overlap as well as the sample size.

On the other hand, class overlap clearly degraded sensitivity at all degrees. A higher decrease in sensitivity per change in %overlap can be seen when class imbalance was high enough. This was the symmetrical effect that class imbalance and class overlap had on each other. That is, the presence of one element can strengthen the scale of impact of the other element. However, this only applied in some certain scenarios, e.g. when class imbalance was sufficiently high. It is worth pointing out that even with no presence of class imbalance (%minority =

100), the influence of class overlap on sensitivity was apparent. In contrast, when there was no class overlap, the ideal sensitivity value was achievable regardless of imbalance degrees. Thus, it can be said that the effect of class imbalance is dependent of the presence of class overlap, but not the other way around. Finally, all of these results suggest that class overlap hurts sensitivity more than class imbalance.

Figure 4 also shows that specificity increased as class imbalance increased. This is expected because the increase in size of the dominating class was in favour of specificity. On the other hand, specificity was negatively affected by class overlap due to the decrease in visibility of majority class instances. It can be observed that class overlap had a higher impact on sensitivity than on specificity. This was because class overlap was measured with respect to the data space of the minority class. Thus, the overlapping region occupied larger data space of the minority class than that of the majority class, relatively to the class size. In an extreme case, the overlapping region occupied the entire minority class but only some part of the majority class.

Interestingly, it can be seen in Figure 4 that class imbalance had no apparent impact on BA and AUC. In contrast, it is clear that BA and AUC decreased as class overlap was higher. This was due to the fact that when class overlap increases, the number of hard-to-classify samples in both class is higher. This is another evidence that researchers should pay more attention to the problem class overlap.

4.2. Conclusion

The experiment clearly shows that class overlap hurt the learning algorithm's performance more than class imbalance. This is evidenced by the results in sensitivity, balanced accuracy and AUC. While class overlap always degraded the results, class imbalance had an impact only in the presence of class overlap. Moreover, the scale of impact of class imbalance on sensitivity highly depended on the degree of class overlap. That is class imbalance was more impactful when class overlap was high and it seemed insignificant when class overlap was low. Lastly, class overlap showed apparent influence on the trade-off between sensitivity and specificity, i.e. BA and AUC, whereas class imbalance did not.

5. Existing Solutions

Existing literature often discussed solutions to imbalanced datasets as data-level and algorithm-level methods [63, 64, 65]. Data oversampling and undersampling are among the most common data-level techniques. At the algorithm level, new learning algorithms and modifications of standard learning algorithms are developed. Algorithm-level methods have an advantage of incorporating user's requirements into the model [19]. However, as opposed to data resampling methods, they do not allow flexible choices of learning algorithms. The combinations of data-level and algorithm-level methods, i.e., ensemble-based methods, have also been used. Ensemble-based methods have advantages in both data and algorithm levels, and are less likely to suffer from overfitting than data resampling [66].

To serve the purpose of this paper, we categorised existing methods into class distribution-based and class overlap-based focuses. Class distribution-based solutions mainly focus on reducing the effect of imbalanced class distributions. Class overlap-based methods deal with instances in the overlapping region to improve classification results. Additional recent methods using emerging techniques are also discussed. The overview of the reviewed methods is provided in Table 1.

5.1. Class distribution-based methods

We categorised methods that are designed to reduce the bias in class distribution as class distribution-based methods. Figure 5(b) and (c) illustrates solutions that rebalance the data by means of oversampling and undersampling, respectively.

Random resampling, the simplest and most common approach, is the process of either randomly eliminating majority class instances (undersampling) or

Table 1: Overview of existing methods

category	technique	resulting classs distribution	methods
class distribution-based	random sampling	varied by settings	Random undersampling; Random oversampling
	linear interpolation	varied by settings	SMOTE[10]
	clustering	balanced	DBSMOTE [47]; k-means SMOTE[12]; k-means undersampling [11]; Sensitivity-based undersampling[64]; density peak-based undersampling[67]; DBMUTE [56]
	neighbourhood search	varied by settings	SLSMOTE[68]; Borderline SMOTE[13]
		balanced weights	Adaptive kNN[69]
		inversed	k-INOS[70]
	neural networks	balanced	GRSOM[71]; Radial-based oversampling[72]; SMOTE-CSELM[73]; UBRKWELM[74]; UBKELM[75]
	ensemble	balanced	RUSBagging[55, 76]; RUSBoost[77]; SMOTEBagging[78]; BalancedEnsemble[79]; GRSOM[71]; UBRKWELM[74]; UBKELM[75]
		varied by settings	SMOTEBoost[80]
		inversed	Inverse-imbalance Bagging[81]
class overlap-based	clustering	not balanced	OBU[45]; BoostOBU[3]; DBMUTE[56]
		not balanced	MWMOTE[60]
		balanced	A-SUWO[14]; NI-MWMOTE[82]
	neighbourhood search	balanced	ADASYN[83]; LR-SMOTE[84]
		not balanced	NCL[85]; PNN[86]; NB-based undersampling[33]
	noise removal	not balanced	SMOTE-IPF[87]; Redency-driven Tomek-link undersampling[51]
	SVM	not balanced	VIRTUAL[88]; OSM[37]; ACFSVM[89]; IEFSVM[90]
		balanced	DCS[91]
	ensemble	not balanced	HardEnsemble [30]; EVINCI[92]
	region splitting	not balanced	Soft-Hybrid [93]
Others	clustering	balanced	Hierachical decomposition[94]
		not balanced	PSS [95]
	ensemble	not balanced	PT-bagging[53]; Random Balance[96]
	Evolutionary algorithm	balanced	EBUS[97]; EUSBoost [98]; EGIS-CHC[99]
		not balanced	EUSCM[97]; EPRENNID [100]
	neural networks	balanced	LMLE-kNN[52]; cGAN oversampling[31]; MFC-GAN[32]
		balanced error weights	DNN-MFE[101]
		not balanced	Attention Aggregation[50]; CoSen[102]; CSDNN[103]; Focal Loss[104]; DQNimb[105]
	SVM	not balanced	Adaptive FH-SVM[106]

synthesising new minority class instances (oversampling) to achieve the balanced class distribution. Although it is simple to employ, random undersampling can potentially lead to a loss of important information while random oversampling is prone to overfitting [7]. Moreover, it was shown that randomly rebalancing class distribution did not guarantee better results [107].

One of the most well-established methods, Synthetic Minority Over-sampling Technique (SMOTE), was designed to create new instances using linear interpolation between minority class neighbouring points [10]. The authors suggested that the method could expand the decision regions of the minority class and as a results caused less overfitting than random oversampling. Due to its simplicity yet decent performance, SMOTE has been widely applied to real-world problems [108, 109, 110]. However, its weaknesses has been presented. In [99], it was shown that by applying SMOTE, their classification results were not improved. This could have been because the method does not include any selection criteria for linear interpolation; hence, synthesised instances may not be useful unless they are near the decision boundary. For more detailed discussion on drawbacks of SMOTE, the reader is referred to [111]. These disadvantages have led to many extensions of SMOTE such as DBSMOTE [47], DBMUTE [56], Borderline-SMOTE [13], Safe-Level-SMOTE [68] (SLSMOTE) and others [12, 80, 78].

DBSMOTE [47] is an oversampling method relying on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [112] to locate instances in different areas. SLSMOTE [68] is another oversampling method based on neighbourhood searching. The main objective of both methods is to synthesise more minority class instances in the non-overlapping region and minimise the synthesis in the overlapping and borderline areas. Although both DBSMOTE and SLSMOTE often achieved improvement over SMOTE, other extensions of SMOTE showed superior performance. In particular, these were DBMUTE [56] and Borderline-SMOTE [13], which also utilize DBSCAN and neighbourhood searching, respectively. It is worth noting that, however, these two methods synthesise more minority class instances near the borderline regions, which

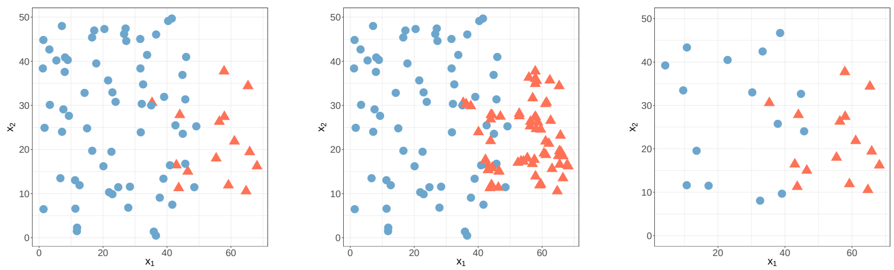

Figure 5: Class distribution-based resampling applied on (a) the original imbalanced and overlapped dataset using (b) SMOTE and (c) k-means undersampling

is the opposite approach to DBSMOTE and SLSMOTE. Detailed discussion of DBMUTE, Borderline-SMOTE and other class overlap-based extensions of SMOTE is provided in the following subsection.

In [12], the authors proposed a method to account for possible amplification of noise created by SMOTE. They applied k-means clustering to discover clusters dominated by the positive class. This was followed by oversampling in such clusters with the oversampling amount inversely proportional to the number of positive instances. A similar approach was presented in [70]. Both methods however led to significant decreases in the minority class accuracy. This could have been attributed to the exclusion of essential positive instances that were sparse and overlapped with dominating negative instances, especially those near the borderline.

Although undersampling has an advantage of reducing the training set size, which results in lower computational costs [113], this could lead to information loss at the same time. To address this issue, clustering is among the common techniques employed during undersampling to ensure the diversity of the remaining instances. In [64] and [11], the authors applied k-means clustering on the majority class and selected representative instances from each cluster. Similar approach was proposed by Di et al. [114]. The authors used a more recent clustering algorithm, density peak-based clustering [67], which not only considers the distance but also the local density. These clustering-based methods resulted in reduced training sets with diversified samples. However, since balanced class distribution was aimed, when applying these method on a dataset with a relatively very small minority class size, they nonetheless resulted a significant loss of information.

Several solutions based on neural networks have also been recently proposed [71, 72, 73, 75]. In [71], instance generation was based on self-organising map and growing ring technique, which are neural network algorithms, aiming at preserving the topology of the original data while rebalancing the class distribution. Unlike other typical data generation methods, this method involves synthesising instances of both majority and minority classes. When majority class undersampling is needed, an entirely new majority class instances are created to replace the original minority class instances. Raghuwanshi and Shukla have recently proposed many variants of methods based on extreme learning machine (ELM) [73, 75, 115, 74, 116]. ELM is a single-layer feed-forward neural network that uses a random approach to generate the hidden layer weights. This enables its training speed to be faster than other gradient-based algorithms [74]. The authors exploited this benefit of ELM, and since the traditional ELM was not designed for imbalanced data, they proposed to use many techniques to rebalance the data such as class-specific regularization parameters computed based on the class distribution [115], SMOTE [73] and UnderBagging [74, 75].

Another neural network-based method was introduced in [72]. The authors used radial basis functions to locate overlapping and non-overlapping regions and avoided synthesising new minority class instances in the overlapping region. However, by doing so, the density of the minority class instances in the overlapping region became relatively sparser. As a consequence, they had a higher tendency to appear as noise to the learning algorithm. Results showed that the method improved specificity but led to lower sensitivity, which is undesired in imbalanced problems. This was consistent with the results obtained with DBSMOTE [47] and SLSMOTE [68] discussed earlier, and underlines the need of improving the visibility of the minority class instances in the overlapping region.

Ensemble-based classifiers, which are known to often outperform single classifiers [22], have been extensively adopted to handle imbalanced datasets.

In [79], the authors proposed an approach to preserve all available information in building an ensemble-based classifier. This was achieved by subsetting the majority class and combining with the minority class instances with equal class distribution. Other than preventing information loss, another advantage of this method is that every base classifier is trained with no bias in class distribution.

Several widely-known ensemble-based methods are the integrations of ensemble algorithms, such as Bagging (i.e. Bootstrap aggregating) [117] and Boosting [118], and common class distribution-based methods. These are, for example, the combinations of random undersampling and Bagging [55, 76], random undersampling and Boosting (RUSBoost) [77], SMOTE and Boosting [80], and SMOTE and Bagging [78]. These methods provided promising results, however, at the cost of higher computational complexity.

Unlike typical class distribution-based methods, which attempt to rebalance the class distribution, an inversion of class imbalance was proposed in [81]. This was done by randomly undersampling the negative class until the positive class became over-represented. As a result, higher positive class accuracy was obtained. At the same time, this caused lower negative class accuracy. The authors addressed this issue by combining the approach with Bagging. Results showed that by doing so, the trade-off between TPR and FPR was improved.

5.2. Class overlap-based methods

Class overlap-based methods mainly address the class overlap problem in classification of imbalanced datasets. Methods in this category deal with either overlapped instances near the borderline or instances in the entire overlapping region. Folllowing [93], we define borderline instances as those along the borderline region between the two classes whereas overlapped instances may reside further from the border. Therefore, we can say that borderline instances are a subset of overlapped instances. The common objective of overlap-based approaches is to emphasise the presence of the minority class in the overlapping region. This is depicted in Figure 6, which shows the resulting datasets after applying simple class overlap-based resampling methods. In Figure 6(b), (c) and (d),

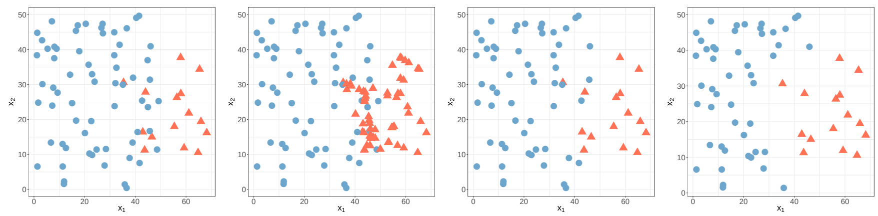

Figure 6: Class overlap-based resampling applied on (a) the original imbalanced and overlapped dataset using (b) Borderline SMOTE (c) borderline-based undersampling and (d) overlap-based undersampling

oversampling of borderline minority class instances, undersampling of borderline majority class instances, and removing entire majority class instances from the overlapping region was performed, respectively. As can be seen from these examples, it is worth pointing out that class overlap-based methods may not necessarily produce a balanced class distribution. With the the risk of potential information loss, most existing overlap-based methods focused specifically on borderline instances, whereas few dealt with the entire overlapping region [33].

Overlap-Based Undersampling (OBU) [45] is among few methods that consider the entire overlapping region. The method was designed to maximise the visibility of minority class instances by eliminating all majority class instances in the overlapping region. A soft-clustering algorithm was employed to assign membership degrees to instances, which enabled identification of indecisive instances. This led to detection and removal of majority class instances that potentially resided in the overlapping region. OBU showed significant improvement in classification and outperformed class distribution-based k-means undersampling [11]. Some drawbacks of OBU such as the need for parameter tuning and possible inaccurate identification of overlapped majority class instances were addressed in [3]. The authors developed two new extensions of OBU, namely AdaOBU and BoostOBU. An adaptive threshold designed to control the elimination process based on the degree of class overlap was incorporated in the methods. The adaptive threshold also replaced the parameter setting requirement of OBU allowing generalisation of the two extensions across different problems. BoostOBU improved the process of identifying potential overlapped instances by emphasising borderline minority class instances prior to data clustering. AdaOBU and BoostOBU substantially improved classification results over the original OBU and outperformed many state-of-the art algorithms, especially on sensitivity, on extensive datasets covering a wide range of scenarios. The methods also often achieved higher sensitivity than robust ensemble-based methods namely SMOTEBagging [78] and RUSBoost [77]. The ability of the methods to provide outstanding results in sensitivity makes them suitable for many real-world problems that require high predication accuracy on the minority class such as in the medical domain and security-related issues, where the accuracy of the majority class can be more compromised.

A neighbourhood-based (NB-based) undersampling framework was proposed in [33] aiming at accurately removing problematic majority class instances from the overlapping region to prevent excessive elimination while enhancing the presence of the minority class. Four methods based on neighbourhood searching to locate overlapped majority class instances were used. Competitive results with other state-of-the-art methods were achieved. Also, its superior results over OBU [45] suggesting that under the same objective to maximise the presence of minority class instances, this NB-based framework, which uses a more local technique, was more efficient. Its successful application in the medical domain was also demonstrated [3]. One of the NB-based methods was selected to handled public medical-related datasets and showed the highest sensitivity on average, which is usually preferable in a medical problem, while often obtaining higher trade-offs between sensitivity and specificity than other methods. However, these NB-based methods are solely based on Euclidean distance. There may be some variations in results on real-world problems due to other data characteristics such as density and class densities, which was not considered in the work.

DBMUTE [56] is another overlap-based undersampling method. The authors employed DBSCAN to discover sub-clusters within the minority class, which was the same technique used in DBSMOTE [47]. However, DBMUTE has a different objective that is to eliminate majority class instances near the minority class sub-clusters. Results showed that DBMUTE significantly outperformed

DBSMOTE, which in contrast avoids improving the visibility of the minority class in the overlapping region.

As an alternative to the above methods, Adaptive Synthetic sampling (ADASYN) was introduced to enhance the presence of the minority class by selectively oversampling in the overlapping region [83]. Instance generation was based on the neighbouring condition. That is, the amount of new instances generated from each minority class instance was proportional to the number of its majority class nearest neighbours. Consequently, more instances were created in the overlapping region while unnecessary syntheses outside such a region were avoided. However, a major drawback of ADASYN is that sparse minority class instances that are highly overlapped with majority class instances will be excluded from oversampling.

An ensemble-based method, HardEnsemble, incorporating both oversampling and undersampling to address overlapped instances of both classes was proposed in [30]. Undersampling was performed based on the contribution to the classification accuracy of instances, which potentially facilitated removal of majority class instances in the overlapping region. Using the same criterion, oversampling was done particularly on minority class instances in the overlapping area. These two resampling processes were carried out in parallel and the resulting datasets were used to train RUSBoost [77]. HardEnsemble showed comparable performance with other solutions. Moreover, it has a benefit over many other existing solutions of no parameter tuning required.

Another method based on ensemble and an Evolutionary Algorithm (EA) was proposed in [92]. An EA was employed so that negative instances were selectively removed from the overlapping region and relatively more minority class instances were present. The method was applied to multi-class imbalanced problems and outperformed other state-of-the-art ensemble-based methods. However, by utilising both EA and emsemble techniques, this method requires high computation costs.

In [93], the authors proposed to use different learning algorithms for classifying different regions of a dataset. Non-overlapping, overlapping, and borderline regions were identified using information based on the data characteristics such as the maximum Fisher's discriminant ratio, probability distributions of the two classes, and the distance between the centers of the two classes. This was followed by using different learning algorithms in the different regions. DBSCAN was selected to learn the borderline region due to its ability in discovering arbitrary-shaped clusters. At the same time, Radial Basis Function Network (RBFN) was used to classified instances in the other regions. This approach showed improvement in classification results. However, it is only applicable to datasets with Gaussian distribution, which is not ideal for handling real-world problems.

With a lower risk of information loss, several methods only focus on overlapped instances that reside near the decision boundary, which we realise as borderline instances. An early and well-known method, Neighbourhood Cleaning Rule (NCL) [85], was adapted from the Edited Nearest Neighbor algorithm (ENN) [119]. NCL is based on removing negative instances that are either misclassified or cause misclassification of positive instances using the *3-NN* classifier. Since NCL only considers three nearest neighbours, it is likely that many negative instances would still remain in the overlapping region, especially in a highly imbalanced and overlapped case. Similar approach was developed by Liang et al. [84], where negative nearest neighbours of misclassified positive samples by SVM were all removed. Further to that, SMOTE was applied to positive instances near the class center to avoid enlarging the effect of noise, which is the drawback caused by the random process of SMOTE. Both of these process contributed to the improvement in the visibility of the positive class, and this was reflected by higher sensitivity achieved. Moreover, since information loss of the negative class was minimised, good trade-offs between sensitivity and specificity were obtained.

In [51], aiming at minimising information loss, only negative instances with high similarities and low contribution to classification were removed. However, no thresholds were defined as a stopping criterion for undersampling, and instead negative instances were progressively eliminated according to the similarity and contribution factors until a balanced class distribution was obtained. Applying this method on a highly imbalanced datasets could anyway result in excessive elimination of negative instances.

SMOTE-IPF [87] was proposed in an attempt to remove noisy instances in the original data as well as those generated by SMOTE. This was done by simply applying a noise filter after SMOTE. The authors suggested that this approach had the following advantages over other methods by removing noise prior to oversampling. Firstly, sparse positive instances near the borderline mistaken as noise would no longer appear as noise after applying oversampling and hence would not be filtered out. This would preserve highly important information, e.g. rare cases, as well as expand the decision boundary of the positive class. Secondly, having more positive instances in the overlapping region could result in some negative instances being filtered out, hence enhancing the visibility of the positive class in such a region to the learning algorithm.

A modification of kNN to improve the classification of imbalanced datasets, Positive-biased Nearest Neighbour (PNN), was presented in [86]. The classification decision was adjusted to be biased towards the positive class, particularly in the regions where positive instances were found under-represented. This benefited the positive class especially in the overlapping region. The method showed superior performance over other neighbourhood-based algorithms with significantly lower computational cost.

In addition to class overlap, the problems of small sub-clusters and within-class imbalance were also addressed in [60, 14, 82]. Majority Weighted Minority Oversampling Technique (MWMOTE) [60] and Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) [14] were proposed. In both methods, borderline minority class instances were discovered using kNN and assigned higher weights for oversampling. In addition to kNN, a semi-unsupervised hierarchical clustering was applied to improve the identification of such instances in A-SUWO. Subsequently, new instances were synthesised within each sub-cluster. MWMOTE created more new instances in sparser sub-clusters whereas A-SUWO focused on oversampling more instances in sub-clusters with higher misclassification errors. Both methods showed improvement in classification results, however, with many parameters needed to be fine-tuned. Moreover, A-SUWO uses complex techniques that may cause poor sampling when it overcomplicates sub-clusters [82]. Wei et al [82] further improved these approaches and proposed NI-MWMOTE, which was developed based on MWMOTE [60]. They introduced adaptive noise removal based on distance and neighbour density before considering borderline instances for oversampling to avoid generation of new noise.

Support Vector Machine (SVM) is one of the most frequently-used classifiers with imbalanced problems [22]. It has also been adapted in several methods for handling imbalanced datasets [88, 91, 120]. This includes the use of support vectors to identify and resample potential borderline instances [88, 91] considering that support vectors are mostly composed of such instances [91]. In [88], an SVM-based active learning algorithm was combined with SMOTE to adaptively synthesise instances between positive support vectors in each active learning. Unlike typical data resampling, this oversampling was repeatedly performed during the training process. Similarly, Jian et al. [91] resampled instances based on support vectors. They made use of Biased SVM [121], which is a learning algorithm implemented specifically to handle imbalanced datasets, to identify support and non-support vectors in the training data. Oversampling and undersampling were then applied to support and non-support vectors, respectively. By doing so, more informative instances were emphasised and information loss was minimised. Cho et al. [120] developed IEFSVM based on EFSVM [90] with a modified entropy for the fuzzy SVM algorithm (FSVM) [122]. IEFSVM reduced the importance of majority class instances that were detected close to minority ones. This was considered from the changes in the nearest neighbours' classes when the number of nearest neighbours (k) was increased. However, this technique may not be sufficiently effective when the majority class highly dominates in the overlapping region unless k is set high enough.

An algorithmic solution based on SVM, an overlap-sensitive margin classifier (OSM), was proposed in [37]. It began with instance weighting that was proportional to the degrees of class imbalance and class overlap, and locating different regions, i.e. highly overlapping and low overlapping, using the FSVM. Then, different learning algorithms were employed in different regions. In the low overlapping region, the classification was carried out using fuzzy SVM. An extreme local search algorithm, *1-NN*, which had shown better results than other classifiers for highly imbalanced and overlapped data [16], was used in the highly overlapping region. Results showed that OSM outperformed other well-known SVM-based classifiers while consuming lower training time. In [89], FSVM was employed with modified membership values to give lower importance to borderline majority class instances. Such instances as well as potential majority-class outliers were identified using techniques based on SVDD [40] and the kernel kNN. This allowed the classification boundary to shift toward the minority class. The method outperformed other SVM-based techniques for imbalanced data. However, with infeasibility of SVM on large datasets due to the huge memory requirement [106], these SVM-based methods face the same limitation.

5.3. Emerging methods

Rather than focusing on the class overlap and class imbalance problems, many recent solutions use alternative approaches in handling classification of imbalanced datasets. These include the use of emerging techniques such as deep neural networks-based algorithms, genetic algorithms and one of the latest technologies, deep reinforcement learning. Unlike traditional solutions, some of these methods have the main objective to preserve the topology of the original data and in some methods, undersampling is not limited to majority class instances but removal of minority class instances is also allowed.

A hierarchical classification method that is an integration of basic methods, e.g. clustering, outlier detection and feature selection, was proposed in [94]. The authors pointed out that clustering of outliers and minority class instances may provide similar results. Thus, they employed an outlier detection method to detect minority class instances in each level of the hierarchy. The method was able to effectively handle highly imbalanced and highly overlapped datasets. Data clustering was also used in [95], however for a different purpose. The authors employed such a technique to allow parallel sampling in large datasets. All discovered clusters of the majority class were simultaneously undersampled to speed up the learning process. Undersampling was carried out in a way that minimum negative class instances were remained for effective training of an SVM classifier. That is, only negative instances near the class boundary were kept in the training set. The method proved a substantial reduction in the computational complexity while comparable results to other existing methods were achieved.

As distinct from typical algorithm-level methods, PT-bagging [53] was designed to calibrate the decision probability at the learner's output aiming at reducing the bias in classification decisions towards the majority class. A threshold-moving technique was used to consider the best threshold for each class instead of the commonly-used cut-off probability of 0.5. The technique was combined with Bagging for improved results. Without changing the natural class distribution of data, this approach showed competitive results with other state-of-the-art ensemble-based methods. Another ensemble-based method is developed by Díez-Pastor et al. [96]. In this work, an ensemble was simply built upon subsets of the training data with random class distributions. To obtain different class distributions, random undersampling and SMOTE were applied. The variety of the training subsets resulted in diversified weak classifiers, which is beneficial for constructing a good ensemble-based model [123]. Despite its simplicity, results showed that this method performed better than some other state-of-the-art ensembles that are more complex.

The application of Evolutionary Algorithms (EAs) has been extensively seen in recent solutions to imbalanced problems [99, 97, 98, 100]. An undersampling framework based on evolutionary prototype selection algorithms was introduced in [97]. The framework aimed at maximising classification results while minimising the training set size. Many variations of methods under this framework were proposed. Both balanced and imbalanced training sets were obtained using the proposed variations, and unlike most undersampling methods, removing minority class instances was allowed. Substantial reductions in sizes of both positive and negative classes were reported while comparable results with well-established methods were achieved. An ensemble-based extension of this evolutionary-based undersampling approach, EUSBoost, was presented in [98]. EUSBoost is the integration of Boosting and the evolutionary-based undersampling with a modified fitness function to obtain diversified weak classifiers. The extension showed better performance over many state-of-the-art ensembles.

EPRENNID is an integratation of ensemble, undersampling and oversampling based on evoluationary algorithms [100]. In particular, evolutionary prototype selection and prototype generation were used as undersampling and oversampling techniques, respectively. By employing evolutionary prototype selection on both positive and negative instances, several reduced subsets were obtained. Then, only well-performing subsets were selected for subsequently applying prototype generation on. To avoid overfitting, which may be introduced by prototype generation, combinations of several resampled subsets were used for ensemble-based classification. EPRENNID produced relatively robust results on different densities of the minority class compared to some existing solutions while reducing instances of both classes. The method showed better performance than many well-known methods; however, its training time was far higher than those methods. This was attributed to the use of an EA together with an ensemble technique, which are both computationally expensive.

Another evolutionary-based method was proposed in [99]. The authors applied an EA for selecting the generalised exemplars, i.e. representative instances, that maximised classification results, particularly in AUC. Classification decisions of new instances were made based on their distances to these generalised exemplars. Experiments showed that the method performed better than other exemplar-based learning algorithms.

One of the most recent approaches for handling imbalanced datasets is the use of neural network algorithms. Like other learning algorithms, deep Neural Networks have been used to learn imbalanced datasets, and to improve performance, data resampling and cost-sensitive learning methods were applied [52, 124, 103]. A great number of new loss functions for handling class imbalance have been introduced recently. In [101, 50], new loss functions were formulated to reduce the bias in imbalanced class distribution. The authors of [101] proposed to use loss functions that considered the error rates of individual classes; however, results showed trivial improvement over the mean square error (MSE), a commonly-used loss function in deep learning. In [103] and [102], novel loss functions were introduced for the purposes of both neural network training and feature extraction. These loss functions were shown to improve the classification performance. A recent work of Tsung-Yi et al. [104] on Focal Loss has received a significant amount of attention. Focal Loss was developed based on the standard cross entropy loss function. It down-weights the loss assigned to easy-to-classified majority class samples. This allows the focus on hard samples during the training process. The method was shown to outperform state-of-the-art Faster R-NN variants [125] in object detection. Due to its simplicity and effectiveness, many later methods have been designed based on Focal Loss. This includes its adaptation in the loss function of standard SVM to handle imbalanced data [106].

In [69], two novel adaptive kNN algorithms for imbalanced classification were proposed. Neural networks were applied in the first proposed algorithm to find the minimum value of k that correctly classified each instance in the training set. In the second algorithm, the value of k was inversely proportional to the local density. This allowed a relatively smaller k value to be used in the overlapping region, which was suggested to be more effective in classifying overlapped instances than a high value of k [37, 16].

Over the past few years, extensions of a state-of-the-art data augmentation algorithm, Generative Adversarial Net (GAN) [126], have been used as oversampling methods for imbalanced datasets [31, 32, 127]. GAN consists of two models – the generative model, which generates new samples as similar to the original data as possible, and the discriminative model, which attempts to distinguish between the original data and the generated data. The objective of GAN is to simultaneously optimise the two models so that the overall distance between the original and the generated distribution is minimised. This ability of GAN was employed as an oversampling technique in [31] and [32] to synthesise minority class instances. In [31], Conditional GAN (cGAN) [128] was directly applied as an oversampling method. Since GAN is an unsupervised learning algorithm, the authors included class labels as an additional learning condition required in cGAN. Results showed that the method outperformed common resampling methods such as Borderline SMOTE [13], ADASYN [83]. However, there was inconsistency in the results, which migh have been attributed to insufficient numbers of training data [129, 130]. In [32], Multiple Fake Class GAN (MFC-GAN) was proposed specifically as an oversampling technique to rebalance class distribution. Unlike common GAN extensions, MFC-GAN was designed to created multiple fake classes to improve the classification accuracy of the minority class. This method was evaluated on multi-class image datasets and results showed that it outperformed SMOTE and other GAN extensions [131, 132]. Despite promising results achieved using these GAN-based methods, a limitation on the size of training data when applying a deep learning model is a concern.

One of the latest technologies, deep reinforcement learning (DRL), has been recently used to handle imbalanced classification tasks [105]. DRL is a combination of deep learning and reinforcement learning. It has recently gained significant interests by the research community due to its ability to successfully learn complex decision-making tasks, which may not be achievable by other standard learning algorithms [133]. In [105], the authors formulated the classification problem as a sequential decision-making process and solve it using DRL, which followed the approach of Wiering et al. [134] to apply reinforcement learning in classification tasks. This approach is considered relatively new in this research topic and need to be further investigated. Although it is a powerful method, DRL has a major has a major drawback on extreme complexity and computational performance [135]. Moreover, it is limited to only very large dataset as DRL is known to be data-hungry. Despite these advantages, this DRL-based method has revealed a new alternative for handling imbalanced data classification and paved the way for researchers to develop new emerging approaches in this field.

Table 2: Overview of benchmarking methods

benchmark		compared methods
data level	CNN(1968) [136]	[51]; [97]
	Tomek-link(1976) [137]	[51]; [97]; [56]
	NCL(2001) [85]	[97]; [72]
	SMOTE(2002) [10]	[14]; [81]; [70]; [91]; [83]; [68]; [88]; [86]; [53]; [102]; [72]; [56]; [71]; [60]; [79]; [87]; [93]; [37]; [31]; [12]; [3]; [33]; [13]; [94]; [106]; [89]; [84]; [67]; [82]
	Borderline SMOTE(2005) [13]	[14]; [31]; [12]; [70]; [56]; [100]; [47]; [72]; [87]; [3]; [33]; [89]; [82]
	ADASYN(2008) [83]	[70]; [60]; [72]; [31]; [82]
	SLSMOTE(2009) [68]	[14]; [70]; [56]; [47]; [87]; [82]
	MWMOTE(2014) [60]	[14]; [70]; [82]
	k-means undersampling(2017) [11]	[45]; [3]; [33]; [67]
algorithm level	1-NN(2008) [16]	[99]; [97]
	PANDA(2014) [138]; FACENET(2015) [139]; Anet(2015) [140]	[52]
	Fast R-CNN(2015) [141]; GoogleNet(2015) [142]	[50]
	ResNet(2016) [143]	[104]; [50]
	Faster R-CNN(2016) [125]	[104]
ensemble	SMOTEBoost(2003) [80]	[77]; [96]; [79]; [98]
	BalanceCascade(2009) [113]	[64] ; [75]
	SMOTEBagging(2009) [78]	[92]; [96]; [11]; [98]; [3]
	EasyEnsemble(2009) [113]	[81]; [64]; [79]; [98]; [67]; [90]; [73]; [75]
	UnderBagging(2009) [78]	[11]; [79]; [98]
	RUSBoost(2010) [77]	[96]; [11]; [30]; [79]; [98]; [95]; [3]; [90]; [73]; [75]
	Random Balance(2015) [96]	[53]

5.4. Benchmarking methods

An overview of common and well-known methods that were used in the reviewed literature for evaluation and comparison purposes is presented in this subsection. Table 2 outlines these benchmarking methods mapped with their compared methods and listed in the order of publishing year. Table 3, 4 and 5 provide further details based on category of the compared methods, namely class distribution, class overlap and emerging techniques, respectively. In the tables, data type indicates the type of datasets used in the experiments – real-world (real) or simulated (sim). The ranges of class imbalance are shown by the minimum and maximum imbalance levels denoted by min imb and max imb, respectively. We defined the levels based on the gaps in imbalance degrees of datasets used in the reviewed literature, which are as follows: *balanced = 1-1.5, slightly imbalanced = 1.7-3.4, moderately imbalanced = 8-16.4, highly imbalanced = 21.9-46.6, very highly imbalanced = 51-87, and extremely imbalanced = 115 and above*. Finally, the right most column of the tables contains the reviewed methods that were shown to be competitive with the benchmarks with the learning algorithms used.

The information provided in Table 2 - 5 suggests common and reliable

Table 3: Benchmarks for class distribution-based methods

benchmark		data type	min imb	max imb	compared methods
data level	NCL(2001) [85]	real	slightly	very highly	multi(DT,kNN,SVM,NB): Radial-based oversampling [72]
	SMOTE(2002) [10]	real	balanced	highly	DT: Inverse undersampling [81]; multi(DT,kNN,GBM,SVM,RF): k-INOS [70]
			balanced	extremely	Inverse-imbalance Bagging [81]
			slightly	moderately	DT: SLSMOTE [68]
			slightly	highly	multi(BPN, SVM): GRSOM [71]
			slightly	very highly	multi(DT,kNN,SVM,NB): Radial-based oversampling [72]
			slightly	extremely	SVM: density peak-based undersampling[67]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [79]
		real+sim	balanced	extremely	multi(LR, kNN, Gradient tree boosting): k-means SMOTE [12]
			slightly	moderately	DT: Borderline SMOTE [13]
	Borderline SMOTE(2005) [13]	real	balanced	extremely	multi(LR, kNN, Gradient tree boosting): k-means SMOTE [12]
			balanced	highly	multi(DT,kNN,GBM,SVM,RF): k-INOS [70]
			slightly	highly	multi(DT,MLP,NB,kNN,SVM,LR,RF): DB-MUTE [56]
			slightly	very highly	multi(DT,kNN,SVM,NB): Radial-based oversampling [72]
	ADASYN(2008) [83]	real	balanced	highly	multi(DT,kNN,GBM,SVM,RF): k-INOS [70]
			slightly	very highly	multi(DT,kNN,SVM,NB): Radial-based oversampling [72]
	SLSMOTE(2009) [68]	real	balanced	highly	multi(DT,kNN,GBM,SVM,RF): k-INOS [70]
			slightly	very highly	multi: DBSMOTE [47]
	MWMOTE(2014) [60]	real	balanced	highly	multi(DT,kNN,GBM,SVM,RF): k-INOS [70]
ensemble	SMOTEBoost(2003) [80]	real	slightly	very highly	multi(DT,NB): RUSBoost [77]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [79]
	BalanceCascade(2009) [113]	real	balanced	moderately	UBKELM[75]
			slightly	moderately	RBFNN: Sensitivity-based undersampling [64]
	SMOTEBagging(2009) [78]	real	slightly	extremely	multi(DT, MLP): k-means undersampling [11]
	EasyEnsemble(2009) [113]	real	balanced	highly	DT: Inverse undersampling [81];
			balanced	extremely	Inverse-imbalance Bagging [81]; SMOTE-CSELM[73]; UBKELM[75]
			slightly	moderately	RBFNN: Sensitivity-based undersampling [64]
			slightly	extremely	SVM: density peak-based undersampling[67]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [79]
	UnderBagging(2009) [78]	real	slightly	extremely	multi(DT, MLP): k-means undersampling [11]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [79]
	RUSBoost(2010) [77]	real	balanced	extremely	SMOTE-CSELM[73]; UBKELM[75]
			slightly	extremely	multi(DT, MLP): k-means undersampling [11]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [79]

Table 4: Benchmarks for class-overlap based methods

benchmark		data type	min imb	max imb	compared methods
data level	CNN (1968) [136]	real	balanced	slightly	multi(BPN,kNN,SVM,NB): Redency-driven Tomek-link undersampling [51]
	Tomek-link(1976) [137]	real	balanced	slightly	multi(BPN,kNN,SVM,NB): Redency-driven Tomek-link undersampling [51]
	SMOTE (2002) [10]	real	balanced	moderately	multi(SVM, kNN, LR, A-SUWO [14]
			balanced	highly	multi(SVM, ANN, RF, kNN): NI-MWMOTE[82]
			balanced	very highly	SVM: DCS [91]
			slightly	moderately	DT: ADASYN [83]
			slightly	highly	SVM-AL: VIRTUAL [88]; multi(DT,kNN): PNN [86]; multi(DT,MLP,NB,kNN,SVM,LR,RF): DB-MUTE [56]; multi(kNN, DT): MWMOTE [60]
		real+sim	balanced	moderately	DT: SMOTE-IPF [87]; multi(SVM, RBFN): Soft-Hybrid [93]
			balanced	highly	SVM: ACFSVM [89]
			balanced	very highly	SVM:OSM [37]
			balanced	extremely	multi(SVM,RF): NB-based undersampling [33]; multi(SVM,RF,DT,knn) BoostOBU [3]
			slightly	moderately	multi(SVM,RF): LR-SMOTE[84]
	Borderline SMOTE (2005) [13]	real	balanced	moderately	multi(SVM, kNN, LR, A-SUWO [14]
			balanced	highly	SVM: ACFSVM[89]; multi(SVM, ANN, RF, kNN): NI-MWMOTE[82]
			slightly	highly	multi(DT,MLP,NB,kNN,SVM,LR,RF): DBMUTE [56]
		real+sim	balanced	moderately	DT: SMOTE-IPF [87]
			balanced	extremely	multi(SVM,RF): NB-based undersampling [33]; multi(SVM,RF,DT,knn) BoostOBU [3]
	ADASYN (2008) [83]	real	balanced	highly	multi(SVM, ANN, RF, kNN): NI-MWMOTE[82]
			slightly	highly	multi(kNN, DT): MWMOTE [60]
	SLSMOTE (2009) [68]	real	balanced	moderately	multi(SVM, kNN, LR, A-SUWO [14]
			balanced	highly	multi(SVM, ANN, RF, kNN): NI-MWMOTE[82]
			slightly	highly	multi(DT,MLP,NB,kNN,SVM,LR,RF): DBMUTE [56]
			slightly	very highly	multi: DBSMOTE [47]
		real+sim	balanced	moderately	DT: SMOTE-IPF [87]
	MWMOTE (2014) [60]	real	balanced	moderately	multi(SVM, kNN, LR, A-SUWO [14]
			balanced	highly	multi(SVM, ANN, RF, kNN): NI-MWMOTE[82]
	k-means undersampling (2017) [11]	real	slightly	extremely	RF: OBU [45]
		real+sim	balanced	extremely	multi(SVM,RF): NB-based undersampling [33]; multi(SVM,RF,DT,knn) BoostOBU [3]
ensemble	SMOTEBagging (2009) [78]	real	balanced	highly	DT: EVINCI[92]; IEFSVM[90]
		real+sim	balanced	extremely	multi(SVM,RF,DT,knn) BoostOBU [3]
	RUSBoost (2010) [77]	real	balanced	highly	IEFSVM[90]
		real	slightly	extremely	RUSBoost: HardEnsemble [30]
		real+sim	balanced	extremely	multi(SVM,RF,DT,knn) BoostOBU [3]

methods that can be considered as good standards for evaluating purposes. However, it is worth pointing out that some of these methods such as SMOTE and Borderline SMOTE are long-established and have been outperformed by a number of more recent methods. This suggests that there is a need for benchmarking new algorithms against recent and state-of-the-art methods for more convincing and reliable evaluation.

6. Conclusion

In this paper, we provided a comprehensive review on the impact of class overlap in classification of imbalanced datasets. This was presented through an extensive experiment, an in-depth discussion on existing solutions, a technical

Table 5: Benchmarks for other emerging methods

benchmark		data type	min imb	max imb	compared methods
data level	CNN(1968) [136]	real	slightly	extremely	kNN:EA undersampling [97]
	Tomek-link(1976) [137]	real	slightly	extremely	kNN:EA undersampling [97]
	NCL(2001) [85]	real	slightly	extremely	kNN:EA undersampling [97]
	SMOTE(2002) [10]	real	slightly	very highly	DNN: CoSen [102]
			slightly	highly	ensembles(DT,kNN): PT-bagging [53]; SVM: Adaptive FH-SVM[106]
		real+sim	balanced	extremely	multi(LR,SVM,kNN,DT, Gradient tree boosting): cGAN oversampling [31]
			slightly	highly	proposed(SMOTE+ kNN,SVM,DT): Hierachical decomposition [94]
	Borderline SMOTE(2005) [13]	real	balanced	extremely	multi(LR,SVM,kNN,DT, Gradient tree boosting): cGAN oversampling [31]
			slightly	highly	knn: EPRENNID [100]
	ADASYN(2008) [83]	real+sim	balanced	extremely	multi(LR,SVM,kNN,DT, Gradient tree boosting): cGAN oversampling [31]
algorithm level	1-NN(2008) [16]	real	slightly	extremely	EGIS-CHC [99]; kNN:EA undersampling [97]
	PANDA(2014) [138]	real	balanced	highly	LMLE-kNN [52]
	FACENET(2015) [139]	real	balanced	highly	LMLE-kNN [52]
	Anet(2015) [140]	real	balanced	highly	LMLE-kNN [52]
	Fast R-CNN(2015) [141]	real	balanced	highly	Attention Aggregation [50]
	GoogleNet(2015) [142]	real	balanced	highly	Attention Aggregation [50]
	ResNet(2016) [143]	real	balanced	highly	Attention Aggregation [50]
			extremely	extremely	Focal Loss [104]
ensemble	SMOTEBoost(2003) [80]	real	slightly	extremely	DT: RB-Boost [96]
			moderately	extremely	DT: EUSBoost [98]
	SMOTEBagging(2009) [78]	real	moderately	extremely	DT: EUSBoost [98]
	EasyEnsemble(2009) [113]	real	moderately	extremely	DT: EUSBoost [98]
	UnderBagging(2009) [78]	real	moderately	extremely	DT: EUSBoost [98]
	RUSBoost(2010) [77]	real	moderately	extremely	DT: EUSBoost [98]
		real+sim	moderately	extremely	SVM: PSS [95]
	Random Balance(2015) [96]	real	slightly	highly	ensembles(DT,kNN): PT-bagging [53]

discussion on evaluation metrics, and an overview of benchmarking methods. The experiment was carried out at the full scale of class overlap and extreme degrees of class imbalance. Results showed that classification errors increased with the degree of class overlap regardless of imbalance degree. Moreover, the effect of class imbalance highly depended on the presence of class overlap. We also critically discussed related literature and methods for handling imbalanced dataset classification selected from leading peer-reviewed publications. The methods were categorised into class distribution-based approach, class overlap-based approach and other emerging techniques for the discussion. Our experimental results and literature review highlighted the importance of the class overlap problem. In general, the choice of suitable methods will vary across problems due to different misclassification costs and variations in objectives or requirements of the users. However, based on the experimental results, the problem of class overlap should be addressed in all cases whereas handling class imbalance may not be necessary. Suggested approaches are as follows:

- When there is no class overlap, classification tasks can be handled using standard learning algorithms regardless of the imbalance degree. Thus, no application of additional methods is needed.

- For datasets with slight to moderate imbalance degrees, overlap-based methods are likely to be a better approach when improvement in sensitivity is expected. Also, those methods with no concern of rebalancing the class distribution may be preferable since results show clearly that in such scenarios, class imbalance barely has an impact on the learner's performance.

- For highly imbalanced datasets, both class imbalance and class overlap should be addressed. Thus, overlap-based methods that also rebalance the class distribution or class weights are potentially more efficient in improving the classification.

This finding emphasises that more research effort is put into improving class overlap-based algorithms. Development of well-formulated definition and measurement of class overlap, especially on real-world data, should as well be urgently put forward.

The overview of benchmarking methods show frequently-used solutions for evaluation and comparison purposes, which can be seen as good standards for future work. At the same time, some of these methods are long-established and have been constantly outperformed. This suggests the need for further comparison against recent and state-of-the-art methods for more convincing and reliable assessments.

Finally, our review also showed that emerging techniques such as deep learning algorithms and evolutionary algorithms have constantly gained the community's attention. This is because they are self-learning and capable of providing optimal results. Although the use of such algorithms have been widely proposed to address the class imbalance problem [32, 98, 100], class overlap was rarely discussed. Also, these techniques have some well-known limitations. Besides high computational complexity, neural network-based techniques generally require large training data, which is not often available in certain imbalanced domains, e.g, medical-related fields. Thus, another possible future direction may include the development of methods using emerging techniques, for example, GAN-based methods to deal with overlapped instances of small-sized datasets.

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (3) (2009) 15.

[2] B. Krawczyk, M. Galar, Ł. Jeleń, F. Herrera, Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy, Applied Soft Computing 38 (2016) 714–726.

[3] P. Vuttipittayamongkol, E. Elyan, Overlap-based undersampling method for classification of imbalanced medical datasets, in: IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer, 2020, pp. 358–369.

[4] X. Zhang, Y. Zhuang, W. Wang, W. Pedrycz, Transfer boosting with synthetic instances for class imbalanced object recognition, IEEE transactions on cybernetics 48 (1) (2018) 357–370.

[5] E. Elyan, L. Jamieson, A. Ali-Gombe, Deep learning for symbols detection and classification in engineering drawings, Neural Networks (2020).

[6] W. Lin, Z. Wu, L. Lin, A. Wen, J. Li, An ensemble random forest algorithm for insurance big data analysis, Ieee Access 5 (2017) 16568–16575.

[7] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Information Sciences 250 (2013) 113–141.

[8] S. Das, S. Datta, B. B. Chaudhuri, Handling data irregularities in classification: Foundations, trends, and future challenges, Pattern Recognition (2018).

[9] G. E. Batista, R. C. Prati, M. C. Monard, Balancing strategies and class overlapping, in: International Symposium on Intelligent Data Analysis, Springer, 2005, pp. 24–35.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

[11] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, J.-S. Jhang, Clustering-based undersampling in class-imbalanced data, Information Sciences 409 (2017) 17–26.

[12] G. Douzas, F. Bacao, F. Last, Improving imbalanced learning through a heuristic oversampling method based on k-means and smote, Information Sciences (2018).

[13] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, Springer, 2005, pp. 878–887.

[14] I. Nekooeimehr, S. K. Lai-Yuen, Adaptive semi-unsupervised weighted oversampling (a-suwo) for imbalanced datasets, Expert Systems with Applications 46 (2016) 405–416.

[15] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study, Intelligent data analysis 6 (5) (2002) 429–449.

[16] V. García, R. A. Mollineda, J. S. Sánchez, On the k-nn performance in a challenging scenario of imbalance and overlapping, Pattern Analysis and Applications 11 (3-4) (2008) 269–280.

[17] S. Visa, A. Ralescu, Learning imbalanced and overlapping classes using fuzzy sets, in: Proceedings of the ICML, Vol. 3, 2003.

[18] J. Stefanowski, Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data, in: Emerging paradigms in machine learning, Springer, 2013, pp. 277–306.

[19] P. Branco, L. Torgo, R. P. Ribeiro, A survey of predictive modeling on imbalanced domains, ACM Computing Surveys (CSUR) 49 (2) (2016) 31.

[20] J. Stefanowski, Dealing with data difficulty factors while learning from imbalanced data, in: Challenges in computational statistics and data mining, Springer, 2016, pp. 333–363.

[21] A. Fernández, S. del Río, N. V. Chawla, F. Herrera, An insight into imbalanced big data classification: outcomes and challenges, Complex & Intelligent Systems 3 (2) (2017) 105–120.

[22] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: Review of methods and applications, Expert Systems with Applications 73 (2017) 220–239.

[23] H. Kaur, H. S. Pannu, A. K. Malhi, A systematic review on imbalanced data challenges in machine learning: Applications and solutions, ACM Computing Surveys (CSUR) 52 (4) (2019) 1–36.

[24] E. Rendón, R. Alejo, C. Castorena, F. J. Isidro-Ortega, E. E. Granda-Gutiérrez, Data sampling methods to deal with the big data multi-class imbalance problem, Applied Sciences 10 (4) (2020) 1276.

[25] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, F. Herrera, Big data preprocessing: methods and prospects, Big Data Analytics 1 (1) (2016) 9.

[26] J. Bi, C. Zhang, An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme, Knowledge-Based Systems 158 (2018) 81–93.

[27] M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural Networks 106 (2018) 249–259.

[28] J. M. Johnson, T. M. Khoshgoftaar, Survey on deep learning with class imbalance, Journal of Big Data 6 (1) (2019) 27.

[29] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge & Data Engineering (9) (2008) 1263–1284.

[30] L. Nanni, C. Fantozzi, N. Lazzarini, Coupling different methods for overcoming the class imbalance problem, Neurocomputing 158 (2015) 48–61.

[31] G. Douzas, F. Bacao, Effective data generation for imbalanced learning using conditional generative adversarial networks, Expert Systems with applications 91 (2018) 464–471.

[32] A. Ali-Gombe, E. Elyan, Mfc-gan: Class-imbalanced dataset classification using multiple fake class generative adversarial network, Neurocomputing 361 (2019) 212–221.

[33] P. Vuttipittayamongkol, E. Elyan, Neighbourhood-based undersampling approach for handling imbalanced and overlapped data, Information Sciences 509 (2020) 47–70.

[34] H. Sun, S. Wang, Measuring the component overlapping in the gaussian mixture model, Data mining and knowledge discovery 23 (3) (2011) 479–502.

[35] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE transactions on information theory 13 (1) (1967) 21–27.

[36] H. Yu, K. Liu, Classification of multi-class microarray datasets using a minimizing class-overlapping based ecoc algorithm, in: Proceedings of the 5th International Conference on Bioinformatics and Computational Biology, ACM, 2017, pp. 51–54.

[37] H. K. Lee, S. B. Kim, An overlap-sensitive margin classifier for imbalanced and overlapping data, Expert Systems with Applications (2018).

[38] R. C. Prati, G. E. Batista, M. C. Monard, Class imbalances versus class overlapping: an analysis of a learning system behavior, in: Mexican international conference on artificial intelligence, Springer, 2004, pp. 312–321.

[39] H. Xiong, J. Wu, L. Liu, Classification with classoverlapping: A systematic study, in: Proceedings of the 1st International Conference on E-Business Intelligence (ICEBI2010),, Atlantis Press, 2010.

[40] D. M. Tax, R. P. Duin, Support vector data description, Machine learning 54 (1) (2004) 45–66.

[41] M. Bekkar, H. K. Djemaa, T. A. Alitouche, Evaluation measures for models assessment over imbalanced data sets, J Inf Eng Appl 3 (10) (2013).

[42] L. A. Jeni, J. F. Cohn, F. De La Torre, Facing imbalanced data–recommendations for the use of performance metrics, in: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, IEEE, 2013, pp. 245–251.

[43] M. Hossin, M. Sulaiman, A review on evaluation metrics for data classification evaluations, International Journal of Data Mining & Knowledge Management Process 5 (2) (2015) 1.

[44] Q. Gu, L. Zhu, Z. Cai, Evaluation measures of the classification performance of imbalanced data sets, in: International symposium on intelligence computation and applications, Springer, 2009, pp. 461–471.

[45] P. Vuttipittayamongkol, E. Elyan, A. Petrovksi, C. Jayne, Overlap-based undersampling for improving imbalanced data classification, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2018, pp. 1–9.

[46] W. A. Rivera, P. Xanthopoulos, A priori synthetic over-sampling methods for increasing classification sensitivity in imbalanced data sets, Expert Systems with Applications 66 (2016) 124–135.

[47] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Dbsmote: density-based synthetic minority over-sampling technique, Applied Intelligence 36 (3) (2012) 664–684.

[48] K. H. Brodersen, C. S. Ong, K. E. Stephan, J. M. Buhmann, The balanced accuracy and its posterior distribution, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 3121–3124.

[49] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, A. Hussain, Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study, IEEE Access 4 (2016) 7940–7957.

[50] N. Sarafianos, X. Xu, I. A. Kakadiaris, Deep imbalanced attribute classification using visual attention aggregation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 680–697.

[51] D. Devi, B. Purkayastha, et al., Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance, Pattern Recognition Letters 93 (2017) 3–12.

[52] C. Huang, Y. Li, C. Change Loy, X. Tang, Learning deep representation for imbalanced classification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5375–5384.

[53] G. Collell, D. Prelec, K. R. Patil, A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data, Neurocomputing 275 (2018) 330–340.

[54] A. Luque, A. Carrasco, A. Martín, A. de las Heras, The impact of class imbalance in classification performance metrics based on the binary confusion matrix, Pattern Recognition (2019) 6829.

[55] R. Barandela, R. M. Valdovinos, J. S. Sánchez, New applications of ensembles of classifiers, Pattern Analysis & Applications 6 (3) (2003) 245–256.

[56] C. Bunkhumpornpat, K. Sinapiromsaran, Dbmute: density-based majority under-sampling technique, Knowledge and Information Systems 50 (3) (2017) 827–850.

[57] J. M. Lobo, A. Jiménez-Valverde, R. Real, Auc: a misleading measure of the performance of predictive distribution models, Global ecology and Biogeography 17 (2) (2008) 145–151.

[58] J. Huang, C. X. Ling, Using auc and accuracy in evaluating learning algorithms, IEEE Transactions on knowledge and Data Engineering 17 (3) (2005) 299–310.

[59] N. M. Adams, D. J. Hand, Comparing classifiers when the misallocation costs are uncertain, Pattern Recognition 32 (7) (1999) 1139–1147.

[60] S. Barua, M. M. Islam, X. Yao, K. Murase, Mwmote–majority weighted minority oversampling technique for imbalanced data set learning, IEEE Transactions on Knowledge and Data Engineering 26 (2) (2014) 405–425.

[61] M. Denil, T. Trappenberg, A characterization of the combined effects of overlap and imbalance on the svm classifier, arXiv preprint arXiv:1109.3532 (2011).

[62] K. Fawagreh, M. M. Gaber, E. Elyan, Random forests: from early developments to recent advancements, Systems Science & Control Engineering: An Open Access Journal 2 (1) (2014) 602–609.

[63] S.-J. Yen, Y.-S. Lee, Cluster-based under-sampling approaches for imbalanced data distributions, Expert Systems with Applications 36 (3) (2009) 5718–5727.

[64] W. W. Ng, J. Hu, D. S. Yeung, S. Yin, F. Roli, Diversified sensitivity-based undersampling for imbalance classification problems, IEEE transactions on cybernetics 45 (11) (2015) 2402–2412.

[65] N. Ofek, L. Rokach, R. Stern, A. Shabtai, Fast-cbus: A fast clustering-based undersampling method for addressing the class imbalance problem, Neurocomputing 243 (2017) 88–102.

[66] A. Dal Pozzolo, O. Caelen, G. Bontempi, When is undersampling effective in unbalanced classification tasks?, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2015, pp. 200–215.

[67] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.

[68] C. Bunkhumpornpat, K. Sinapiromsaran, 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.

[69] S. S. Mullick, S. Datta, S. Das, Adaptive learning-based k-nearest neighbor classifiers with resilience to class imbalance, IEEE Transactions on Neural Networks and Learning Systems (2018).

[70] R. F. de Morais, G. C. Vasconcelos, Boosting the performance of over-sampling algorithms through under-sampling the minority class, Neurocomputing 343 (2019) 3–18.

[71] D. Chetchotsak, S. Pattanapairoj, B. Arnonkijpanich, Integrating new data balancing technique with committee networks for imbalanced data: Grsom approach, Cognitive neurodynamics 9 (6) (2015) 627–638.

[72] M. Koziarski, B. Krawczyk, M. Woźniak, Radial-based oversampling for noisy imbalanced data classification, Neurocomputing 343 (2019) 19–33.

[73] B. S. Raghuwanshi, S. Shukla, Smote based class-specific extreme learning machine for imbalanced learning, Knowledge-Based Systems 187 (2020) 104814.

[74] B. S. Raghuwanshi, S. Shukla, Underbagging based reduced kernelized weighted extreme learning machine for class imbalance learning, Engineering Applications of Artificial Intelligence 74 (2018) 252–270.

[75] B. S. Raghuwanshi, S. Shukla, Class imbalance learning using underbagging based kernelized extreme learning machine, Neurocomputing 329 (2019) 172–187.

[76] B. C. Wallace, K. Small, C. E. Brodley, T. A. Trikalinos, Class imbalance, redux, in: 2011 IEEE 11th international conference on data mining, IEEE, 2011, pp. 754–763.

[77] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: A hybrid approach to alleviating class imbalance, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 40 (1) (2010) 185–197.

[78] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on, IEEE, 2009, pp. 324–331.

[79] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, Y. Zhou, A novel ensemble method for classifying imbalanced data, Pattern Recognition 48 (5) (2015) 1623–1637.

[80] N. V. Chawla, A. Lazarevic, L. O. Hall, K. W. Bowyer, Smoteboost: Improving prediction of the minority class in boosting, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2003, pp. 107–119.

[81] M. A. Tahir, J. Kittler, F. Yan, Inverse random under sampling for class imbalance problem and its application to multi-label classification, Pattern Recognition 45 (10) (2012) 3738–3750.

[82] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, D. Huang, Ni-mwmote: An improving noise-immunity majority weighted minority oversampling technique

for imbalanced classification problems, Expert Systems with Applications (2020) 113504.

[83] H. He, Y. Bai, E. A. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, IEEE, 2008, pp. 1322–1328.

[84] X. Liang, A. Jiang, T. Li, Y. Xue, G. Wang, Lr-smote–an improved unbalanced data set oversampling based on k-means and svm, Knowledge-Based Systems (2020) 105845.

[85] J. Laurikkala, Improving identification of difficult small classes by balancing class distribution, in: Conference on Artificial Intelligence in Medicine in Europe, Springer, 2001, pp. 63–66.

[86] X. Zhang, Y. Li, A positive-biased nearest neighbour algorithm for imbalanced classification, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2013, pp. 293–304.

[87] J. A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, Information Sciences 291 (2015) 184–203.

[88] Ş. Ertekin, Adaptive oversampling for imbalanced data classification, in: Information Sciences and Systems 2013, Springer, 2013, pp. 261–269.

[89] X. Tao, Q. Li, C. Ren, W. Guo, Q. He, R. Liu, J. Zou, Affinity and class probability-based fuzzy support vector machine for imbalanced data sets, Neural Networks 122 (2020) 289–307.

[90] Q. Fan, Z. Wang, D. Li, D. Gao, H. Zha, Entropy-based fuzzy support vector machine for imbalanced datasets, Knowledge-Based Systems 115 (2017) 87–99.

[91] C. Jian, J. Gao, Y. Ao, A new sampling method for classifying imbalanced data based on support vector machine ensemble, Neurocomputing 193 (2016) 115–122.

[92] E. R. Fernandes, A. C. de Carvalho, Evolutionary inversion of class distribution in overlapping areas for multi-class imbalanced learning, Information Sciences 494 (2019) 141–154.

[93] P. Vorraboot, S. Rasmequan, K. Chinnasarn, C. Lursinsap, Improving classification rate constrained to imbalanced data between overlapped and non-overlapped regions by hybrid algorithms, Neurocomputing 152 (2015) 429–443.

[94] C. Beyan, R. Fisher, Classifying imbalanced data sets using similarity based hierarchical decomposition, Pattern Recognition 48 (5) (2015) 1653–1672.

[95] A. D'Addabbo, R. Maglietta, Parallel selective sampling method for imbalanced and large data classification, Pattern Recognition Letters 62 (2015) 61–67.

[96] J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, L. I. Kuncheva, Random balance: ensembles of variable priors classifiers for imbalanced data, Knowledge-Based Systems 85 (2015) 96–111.

[97] S. García, F. Herrera, Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy, Evolutionary computation 17 (3) (2009) 275–306.

[98] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, Pattern Recognition 46 (12) (2013) 3460–3471.

[99] S. García, J. Derrac, I. Triguero, C. J. Carmona, F. Herrera, Evolutionary-based selection of generalized instances for imbalanced classification, Knowledge-Based Systems 25 (1) (2012) 3–12.

[100] S. Vluymans, I. Triguero, C. Cornelis, Y. Saeys, Eprennid: An evolutionary prototype reduction based ensemble for nearest neighbor classification of imbalanced data, Neurocomputing 216 (2016) 596–610.

[101] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, P. J. Kennedy, Training deep neural networks on imbalanced data sets, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 4368–4374.

[102] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, R. Togneri, Cost-sensitive learning of deep feature representations from imbalanced data, IEEE transactions on neural networks and learning systems 29 (8) (2018) 3573–3587.

[103] Y.-A. Chung, H.-T. Lin, S.-W. Yang, Cost-aware pre-training for multiclass cost-sensitive deep learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI16, AAAI Press, 2015, pp. 1411–1417.

[104] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[105] E. Lin, Q. Chen, X. Qi, Deep reinforcement learning for imbalanced classification, Applied Intelligence (2020) 1–15.

[106] Q. Wang, Y. Tian, D. Liu, Adaptive fh-svm for imbalanced classification, IEEE Access 7 (2019) 130410–130422.

[107] G. M. Weiss, F. Provost, Learning when training data are costly: The effect of class distribution on tree induction, Journal of artificial intelligence research 19 (2003) 315–354.

[108] J. Sun, J. Lang, H. Fujita, H. Li, Imbalanced enterprise credit evaluation with dte-sbd: decision tree ensemble based on smote and bagging with differentiated sampling rates, Information Sciences 425 (2018) 76–91.

[109] M. Ijaz, G. Alfian, M. Syafrudin, 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 8 (8) (2018) 1325.

[110] S. Wang, D. Wang, J. Li, T. Huang, Y.-D. Cai, Identification and analysis of the cleavage site in a signal peptide using smote, dagging, and feature selection methods, Molecular omics 14 (1) (2018) 64–73.

[111] G. Douzas, F. Bacao, Geometric smote: Effective oversampling for imbalanced learning through a geometric extension of smote, arXiv preprint arXiv:1709.07377 (2017).

[112] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proc. of 2nd International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[113] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2) (2009) 539–550.

[114] Z. Di, Q. Kang, D. Peng, M. Zhou, Density peak-based pre-clustering support vector machine for multi-class imbalanced classification, in: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), IEEE, 2019, pp. 27–32.

[115] B. S. Raghuwanshi, S. Shukla, Class-specific kernelized extreme learning machine for binary class imbalance learning, Applied Soft Computing 73 (2018) 1026–1038.

[116] B. S. Raghuwanshi, S. Shukla, Class-specific extreme learning machine for handling binary class imbalance problem, Neural Networks 105 (2018) 206–217.

[117] L. Breiman, Bagging predictors, Machine learning 24 (2) (1996) 123–140.

[118] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of computer and system sciences 55 (1) (1997) 119–139.

[119] D. L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man, and Cybernetics (3) (1972) 408–421.

[120] P. Cho, M. Lee, W. Chang, Instance-based entropy fuzzy support vector machine for imbalanced data, Pattern Analysis and Applications (2019) 1–20.

[121] C.-H. Hoi, C.-H. Chan, K. Huang, M. R. Lyu, I. King, Biased support vector machine for relevance feedback in image retrieval, in: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Vol. 4, IEEE, 2004, pp. 3189–3194.

[122] C.-F. Lin, S.-D. Wang, Fuzzy support vector machines, IEEE transactions on neural networks 13 (2) (2002) 464–471.

[123] S. Wang, X. Yao, Relationships between diversity of classification ensembles and single-class performance measures, IEEE Transactions on Knowledge and Data Engineering 25 (1) (2013) 206–219.

[124] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Transactions on Knowledge & Data Engineering (1) (2006) 63–77.

[125] A. Shrivastava, R. Sukthankar, J. Malik, A. Gupta, Beyond skip connections: Top-down modulation for object detection, arXiv preprint arXiv:1612.06851 (2016).

[126] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[127] A. Ali-Gombe, E. Elyan, C. Jayne, Multiple fake classes gan for data augmentation in face image dataset, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[128] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).

[129] T. Shaikhina, N. A. Khovanova, Handling limited datasets with neural networks in medical applications: A small-data approach, Artificial intelligence in medicine 75 (2017) 51–63.

[130] S. Feng, H. Zhou, H. Dong, Using deep neural network with small dataset to predict material defects, Materials & Design 162 (2019) 300–310.

[131] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 2642–2651.

[132] A. Ali-Gombe, E. Elyan, Y. Savoye, C. Jayne, Few-shot classifier gan, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[133] X. B. Peng, P. Abbeel, S. Levine, M. van de Panne, Deepmimic: Example-guided deep reinforcement learning of physics-based character skills, ACM Transactions on Graphics (TOG) 37 (4) (2018) 1–14.

[134] M. A. Wiering, H. van Hasselt, A.-D. Pietersma, L. Schomaker, Reinforcement learning algorithms for solving classification problems, in: 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), IEEE, 2011, pp. 91–96.

[135] G. Caminero, M. Lopez-Martin, B. Carro, Adversarial environment reinforcement learning algorithm for intrusion detection, Computer Networks 159 (2019) 96–109.

[136] P. Hart, The condensed nearest neighbor rule (corresp.), IEEE transactions on information theory 14 (3) (1968) 515–516.

[137] I. Tomek, Two modifications of cnn, IEEE Trans. Systems, Man and Cybernetics 6 (1976) 769–772.

[138] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, L. Bourdev, Panda: Pose aligned networks for deep attribute modeling, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1637–1644.

[139] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[140] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 3730–3738.

[141] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[142] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[143] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.