Egyptian Informatics Journal xxx (xxxx) xxx

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

TGT: A Novel Adversarial Guided Oversampling Technique for Handling Imbalanced Datasets

Ayat Mahmoud [a,*], Ayman El-Kilany [b], Farid Ali [c], Sherif Mazen [b]

[a] Faculty of Computer Sciences, October University for Modern Sciences and Arts Cairo, Egypt
[b] Information Systems Department, Faculty of Computers and Artificial Intelligence, Cairo University, Cairo, Egypt
[c] Information Technology Department, Faculty of Computers and Artificial Intelligence, Beni-suef University, Egypt

ARTICLE INFO

Article history:
Received 5 June 2020
Revised 15 October 2020
Accepted 19 January 2021
Available online xxxx

Keywords:
Imbalance
Oversampling
Classification

ABSTRACT

With the volume of data increasing exponentially, there is a growing interest in helping people to benefit from their data regardless of its poor quality. One of the major data quality problems is the imbalanced distribution of different categories existing in the data. Such problem would affect the performance of any possible of analysis and mining on the data. For instance, data with an imbalanced distribution has a negative effect on the performance achieved by most traditional classification techniques. This paper proposes TGT (Train Generate Test), a novel oversampling technique for handling imbalanced datasets problem. Using different learning strategies, TGT guarantees that the generated synthetic samples reside in minority regions. TGT showed a high improvement in performance of different classification techniques when was experimented with five imbalanced datasets of different types.

1. Introduction

A collection of data is called imbalanced if one class instances - was higher in number than the other. The class with more instances is referred to as the majority class, and the one with fewer instances is called the minority class [1,2]. Numerous recent researches on imbalanced datasets have generally agreed that because of this skew distribution of classes, the classifiers are biased towards the majority class and give very low classification accuracy towards the smaller classes. Classifier may also classify any sample as the majority class and ignore the minority class [3].

Different strategies have been proposed to address class imbalance problem [4], they range from data-level methods to algorithm-level methods in addition to hybrid methods. Sampling is the most widely used method to imbalanced dataset classification [5]. The idea of sampling is based on changing the dataset so that a more balanced class distribution is created. Methods of sampling can be subdivided into oversampling and undersampling. Undersampling eliminates the number of instances of the majority class while oversampling generates minority class synthetic instances during preparation.

In this paper, we argue that using an oversampling technique that simulates the adversarial architecture can yield better results during the oversampling process and consequently handle binary classification of imbalanced dataset in a better way. In more specific words, we argue that generation process of oversampling can be guided by two classifiers where the first classifier is responsible for the generation of new samples and the second classifier is responsible for the verification process of their correctness. This will ensure that the generated samples reside in minority region. The evaluation results show that adversarial guided oversampling technique outperform the standard oversampling algorithms.

The rest of this paper is organized as follows. Section 2 presents the related work while section 3 introduces the proposed tech-

* Corresponding author.
E-mail addresses: amahmoud@msa.edu.eg (A. Mahmoud), a.elkilany@fci-cu.edu.eg (A. El-Kilany), fared.ali@fcis.bsu.edu.eg (F. Ali), s.mazen@fci-cu.edu.eg (S. Mazen).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2021.01.002
1110-8665/

nique to handle imbalanced data classification problem. Sections 4 and 5 show the details of the performance evaluation and concludes the paper.

2. Related work

Different strategies like sampling, feature selection, cost sensitive classification, and ensembling methods have been proposed in the literature with a lot of variations to enhance the classification performance while data suffer from imbalanced classes' distribution. In this section, we are elaborating the different oversampling methods which are proposed in the literature to handle imbalanced data distribution.

Random oversampling or RO-sampling is one of the most popular techniques of sampling. RO-sampling makes rebalancing of the data distribution by selecting samples randomly and then replicating them. For example, the authors in [6] introduced two local graphs in an updated Random Walk Oversample (RWO). At first a proximity graph (proximity graph with k-nearest neighbors) is used to pick the instances for the minority class in high-density areas, then RWO sampling method is introduced. Without the effects of noises and outliers outside graph or at its boundaries, synthetic minority instances are produced. The second graph selects majority class instances in high-density regions and excludes the remaining instances. Their method performed well in most cases; this may be related to the nature of the datasets used as it showed high performance with continuous datasets only.

Synthetic Minority Oversampling Technique (SMOTE) was first introduced in [7]. It is the most common oversampling technique which addresses the imbalanced dataset problem by generating minority class samples to balance the class distribution. SMOTE gives more related minority class examples to learn from, consequently allowing a learner to carve broader decision regions which results in more coverage of the minority class. Despite this, SMOTE it has significant faults. New samples created in SMOTE firmly lie in the line segment between seed samples. So, the generated examples will not represent the distribution of original data. another problem is that SMOTE may introduce the problem of class overlapping [8]. The authors in [9] introduced a new oversampling technique called SNOCC to overcome SMOTE shortages. At SNOCC, the number of seed instances is increased, and the number of synthetic samples is not limited to two SMOTE seed instances. The nearest neighbors of instances are identified using a new algorithm. Their tests proved that SNOCC worked better than SMOTE and Cluster Based Synthetic Oversampling (CBSO). Another subsequent research proposed a novel over-sampling method explicitly to classify text data, which is provided by the hypothesis of the distribution. Their Random Distribution Process produces modern, random minority synthetic records, taking the advantage of the distributional features of the words in the dataset [10].

Weighted kernel-based SMOTE (WK-SMOTE) was introduced in [11] to resolve the SMOTE shortcomings on nonlinear problems through oversampling in the SVM feature space. WK-SMOTE modifies SMOTE for non-linear separable data by producing the synthetic instances in the classifier feature space rather than the input data space. In comparison with the other baseline approaches on several metrics, the suggested oversampling algorithm together with a cost sensitive SVM model have demonstrated an enhancement in performance. Therefore, a hierarchical structure for multiclass imbalanced issues with a progressive class order is created. The suggested WK-Smote and the hierarchic structure are tested on real world industrial fault detection system.

Two main variables are defined in SMOTE: N oversampling value and the k-neighbors. Nevertheless, in practical implementations, the two variables randomly selected by users cannot be optimized. However, the imbalance ratios of the data are completely distinct, making it more difficult to pick parameters in SMOTE. The authors in [12] proposed a new oversampling method relying on SMOTE to address the problem. This turns the problem of parameter selection into a multi-target optimization problem in SMOTE. To achieve their optimal solution, a new selection technique called absolute dominance-based selection was introduced to search best values for SMOTE parameters [10].

Also, the authors in [13] introduced a modified version of SMOTE to address another SMOTE shortcoming. While SMOTE algorithm generates new samples in the space of the minority data space, the Modified SMOTE algorithm generates new samples in the space separating minority and majority data. Those new samples were able to explain the difference between majority and minority points in a better way and consequently lead to better classifications results.

The study in [14] suggests a novel technique, called minority oversamples on adaptive kernel sub-spaces (MOKAS), which uses the kernel model adaptive subspace auto-organizing map's invariant feature extraction function. New instances are produced from trained subspaces and used in the input space. Such instances also define nonlinear frameworks found in the distribution of minority class data and enable learning models to balance in an accepted - way the distorted class distribution.

In [15], the authors are presenting a novel technique of Self-Organizing Map based Oversampling (SOMO), that generates two-dimensional representations of the input space by applying a Self-Organizing Map to enable the effectiveness of artificial data point generation. SOMO consists of three main steps: First a Self-Organizing Map gives the original, typically high-dimensional, space a two-dimensional representation. Then it produces artificial instances in a cluster and then produces synthetic instances between clusters. They also performed empirical experiments which enhanced the performance of methods, when SOMO produces artificial data.

When interpolating a synthetic example between a noisy data and one of its nearest neighbors, the generated example resides inside the majority class region. Such problem exists in SMOTE and ADASYN. Consequently, it is necessary to filter out the noisy samples. An efficient 3-stage fault diagnostic technique for imbalanced data is introduced in [16] to address this problem. First, a novel method of over-sampling called weighted minority oversampling (WMO) is introduced to make balancing for distribution of data. This adopts a modern method of data generation to avoid fault or excessive sampling. Furthermore, an improved deep auto-encoder (DA) solution is used to pick useful features dynamically. DA is enhanced in two perspectives: first, new costs based on the highest entropy and sparse cost are developed to acquire sparse durable features; second, a self-adjusting learning level to guarantee a good convergence output is achieved.

A new method for oversampling is proposed in [17] which uses the real value negative selection (RNS) method for generating synthetic minority instances with no real minority data necessarily available. The generated minority instances with rare actual minority data are merged with the majority data in order to provide a method for the binary classification learning. In their experiments, they show the efficacy of RNS to prevent the over-sampling problems faced by conventional approaches such as noise generation and redundant samples in the same clusters. However, we noticed from the results it performed well only for severely imbalanced datasets.

The authors in [18] are suggesting radial based oversampling methods (RBO), which can identify areas in which minority class artificial instances are to be produced on a radial base depending upon the imbalance distribution estimate. They take into consideration data got from all classes, in contrast to traditional multi-class over-sampling methods that use only minority class data. Experiment results conducted on a typical dataset indicate that the RBO artificial over-sampling technique offers a promising alternative to the current imbalanced dataset solutions.

A three-way decision model (CTD) is introduced in [19] where the costs of choosing main samples are taken into consideration. First, the CTD uses Constructive Covering Algorithm (CCA) to separate minority instances into multiple coverage. Each cover is then selected and divided into three regions based on the coverage density. Finally, according to the model of cover distribution on minority instances, the respective threshold α and β for CTD is reached, which allows to pick main instances for SMOTE oversampling.

The authors in [20] introduced generative adversarial minority oversampling (GAMO). The idea is that producing synthetic points near the borders of the minority class will let the classifier to learn class boundaries which are more robust to class imbalance. The convex generator produces synthetic points as convex combinations of the existing points from the minority class. They introduced also an additional discriminator which ensures that the generated points belong to the real distribution of the intended minority class.

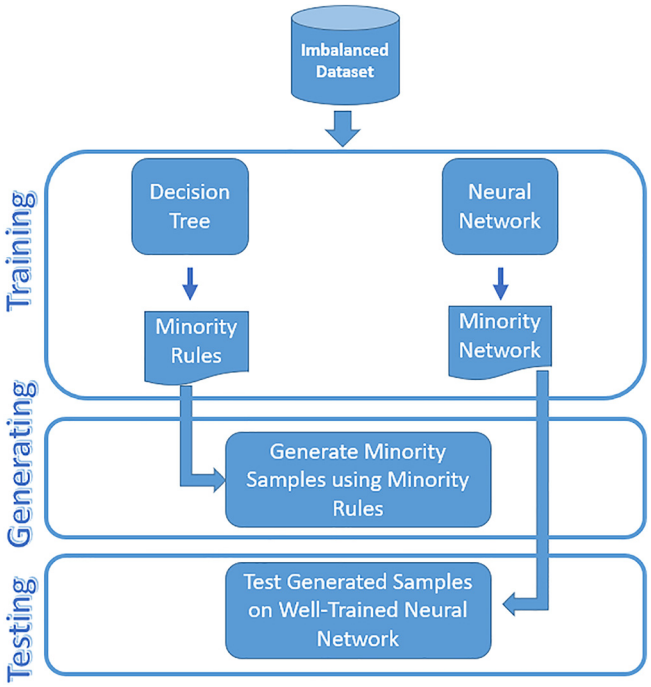

Fig. 1. An Architecture Diagram of TGT.

3. Proposed technique

Train Generate Test (TGT) is based on using adversarial guided oversampling to make some sort of double checking the generated samples to assure they belong to minority class. In TGT, the generation process is guided by minority class rules acquired by training decision tree on the dataset. Decision tree is not used to do classification but rather to extract knowledge about the minority class only where this extracted knowledge will be used to generate synthetic minority samples. Then, it comes to verification process, in which we use a well-trained neural network to make a second check for the generated samples to assure they are all similar to minority class samples given in the dataset. An architecture diagram of TGT is shown in Fig. 1. TGT has three steps: training, generating, and testing. Below is an explanation of each step.

3.1. Training step

The first step is training step which is utilized to train two classifiers on the given data: a decision tree and a neural network. The decision tree is trained to get the minority class classification rules which will later be used to generate the synthetic samples. Then, a neural network is trained to classify the minority and majority classes data. The neural network will be later used for testing the generated samples. Despite the training on imbalanced data, the two classifiers are expected to extract and use all the available knowledge about the minority class in the data. This would guarantee that the new generated samples follow the distribution of the minority class data even though its scarcity.

Training step details is described in Algorithm1. It contains two functions. First one is BuildDT, which takes the training data along with class label, uses Gini index for selecting the attribute that best classifies the instances and outputs arrays containing the upper and lower value for each attribute. The second function is NN which takes the majority and minority sets to be trained and then used later in verifying generated samples.

Algorithm 1: Training

```
Function BuildDT (D,class,atts)
Input: D: instances of training, class: class
label, atts: array of attributes
Output: arrays for lower and upper limits for each
attribute
Create a new root node.
If all instances belong to the same class (have the
same class label) then
Return tree with single root node having label cl
Else
  Selected_Att ← CalcGiniIndex(D, atts) //
select the attribute in atts which best
classifies instances based on its Gini index
  Root_decision_attribute ← Selected_Att
  Foreach value v_i of Selected_Att do
    Add a new branch for the test Selected Att = v_i
    Let Val_inst_vi be the subset of D which have the
value v_i for the Selected_Att
    If Val_inst_vi is empty then
      Add leaf node with its class label
    Else
      BuildDT(Val_inst_vi, class,{atts -
Selected_Att}) //recursion after excluding the
selected attribute
    Endif
  End for
Endif

Minority_Weights = NN(A,B,epochs)
//Inputs of NN: A, B, epochs where A: Majority class
samples, B: Minority class samples, epochs:
#iterations
//Outputs of NN: Minority class weights
End
```

3.2. Generation Step

The second step is to generate synthetic samples. To generate a new sample, the attributes of data are looped and a new value for each attribute is generated based on the lower and upper values that are obtained from minority class classification rules of step1. The decision tree obtained through step1 is considered to be an explainable classifier that we can utilize the understanding for its logic to generate new samples that follow the rules identified by the classifier. Generation step details is described in Algorithm2.

Algorithm 2: Generation

```
   Function Generate (N, t, D, class, atts)
   //Generate samples based on the acquired rules
   Input: N: #samples, t: #attributes, D: instances
   of training, class: class label, atts: array of
   attributes
   Output: Array of generated samples
1  For p = 1 to N do// loop all samples
2      lower[p] ← BuildDT (D, class, atts) // an
   array of lower limits of all attributes
3      upper[p] ← BuildDT (D, class, atts) // an
   array of upper limits of all attributes
4      Sample[p] = rand(lower[p], upper[p]) //
   generate sample based on values in the upper and
   lower arrays
5  End for
6  End
```

3.3. Testing step

The last step is the testing step. In this step, the generated samples are tested or verified using the trained neural network. If classifier shows that the sample belongs to minority class, then it will be kept in new synthetic samples array. Otherwise, it will be discarded. The neural network is considered to be an unexplainable classifier that identify the class based on the finding the right weights of network neurons in order to find the right classification. Using such classifier would ensure that the generated sample of minority class through explainable classifier is verified by another classifier that works in a completely different and unexplained method.

The proposed technique is expected to search till it generates the number of required verified samples. Since the generation process is guided by the minority data classification rules extracted using the decision tree classifier, the verification step is expected to succeed most of the time. During performance evaluation, generation of 1778 verified sample required 2305 trials with a success rate of 77%. Testing step details is described in Algorithm3.

Algorithm 3: Testing

```
   Function Test (N, A, B)
   //Test the generated samples on the well-
   trained neural network
   Input: N: Amount of oversampling, A: Majority
   class samples, B: Minority class samples
   Output: Original Data + (N/100) * Minority class
   samples
1  For j = 1 to N do
2      w(2,1)_{k,j} ←NN (A, B, epochs)
3      w(1,0)_{j,i} ←NN (A, B, epochs)
4      Calculate hidden node inputs (Net(1)_{p,j})
         Net(1)_j = Σ_{i=0}^{n} w(1,0)_{j,i} x_{p,i}
5      calculate hidden node outputs (X(1)_{p,j})
6        x^1_{p,j} = 1/(1 − e^{Net(1)_j})
7      calculate inputs to the output nodes (Net(2)_{p,k})
8        Net(2)_k = Σ_{j=0}^{n} w(2,1)_{k,j} x(1)_{p,j}
9      Compute the network outputs (O_{p,k})
10       O_{p,k} = 1/(1 − e^{Net(2)_k})
11     If O_{p,k} ∈ B
12     Then synth ← concatenate sample with verified
   samples array // add the verified sample to the
   array of verified synthetic samples
13     End if
14 End for
15 Write file
16 End
```

4. Performance evaluation

The objective of the evaluation is to prove the effectiveness of the proposed adversarial guided oversampling technique. The proposed technique aims to balance the class distribution of data by making guided oversampling though using minority class rules driven from training the decision tree on the dataset. And then checking the generated samples using well-trained neural network to assure they all belong to minority class. Towards this goal, the proposed technique is evaluated on different datasets over different classifiers against SMOTE method which is the baseline oversampling approach in the literature and also against one of its very recent smote variations which is Modified SMOTE. The following subsections describes the evaluation details.

4.1. Datasets

For our experiment, we used five numerical datasets which can be downloaded from [21,22] . The first dataset is Poker. It has 11 attributes and 1485 records for 2 classes, the first class is 25 records (value 1) and the second class is 1460 records (value 0). The second dataset is Yeast. It has 9 attributes and 514 records for 2 classes, the first class is 51 records (value 1) and the second class is 463 records (value 0). The third dataset is Cleveland. It has 14 attributes and 173 records for 2 classes, the first class is 13 records (value 1) and the second class is 163 records (value 0). The fourth dataset is Indian diabetes. It has 9 attributes and 768 records for 2 classes, the first class is 268 records (value 1), the second class is 500 records (value 0). The fifth dataset is kc2 software fault prediction. It has 22 attributes and 522 records for 2 classes. The first class is 107 records (value 0). The second class is 415 records (value 1). For each dataset, cross validation procedure was used to split the data into training and testing sets. The number of cross validation folds was set to 10 folds. Details of the datasets is summarized in Table 1.

4.2. Evaluation method

Three evaluation matrices were used for performance evaluation, which are, accuracy, sensitivity, specificity as recommended in literature for evaluating performance of binary classifications

Table 1
Number of Samples in each Dataset Before and After Oversampling.

	Poker		Yeast		Cleveland		Indian Diabetes		Kc2 Software Fault Prediction	
	Before	After	Before	After	Before	After	Before	After	Before	After
Minority	25	650	51	459	13	169	268	536	107	428
Majority	1460	1460	463	463	160	160	500	500	415	415

[23,24]. Accuracy is the most common performance metric in practice, particularly for binary and multi-class classification issues, as seen in various studies [25,26]. Sensitivity determines the amount of real positive that is correctly classified as such, while specificity determines the amount of real negative that is correctly identified. In other words, Specificity metric is used to measure the fraction of negative patterns that are classified correctly. Consequently, sensitivity takes into account the prevention of false negatives, and for false positive specificity does likewise [27].

Three different classifiers, K-Nearest, Fuzzy K-Nearest, and Support Vector Machines classifications, were used for performance evaluation. The KNN classifier takes only one parameter and the best results got when k = 10 while the FKNN classifier takes two parameters k and m where k = 10 and m = 0.5. Classifiers were trained on different settings to compare the proposed oversampling technique against SMOTE and modified SMOTE. They were trained once on data without oversampling, once with data after oversampling it with SMOTE, once with data after oversampling it with modified SMOTE and finally with the data after oversampling it with TGT.

To evaluate the performance of each classifier on each dataset across its different settings of oversampling techniques, 10-folds cross validation was used to split the data into the training and testing sets. The classifier is trained the training set before collecting the performance metrics resulting from testing the classifier on the testing set.

4.3. Evaluation results

TGT is evaluated against data in its raw form without applying any kind of oversampling, against traditional SMOTE [7] and against modified SMOTE [13]. Results of the proposed technique TGT against data without oversampling, SMOTE [7], and Modified SMOTE [13] are summarized in Figs. 2, 3, 4, 5 and 6 using three evaluation metrics as mentioned earlier, accuracy, sensitivity, and specificity.

We can observe that the proposed technique TGT outperformed in all five datasets with different classifiers. We may notice that before oversampling, the classifiers showed high value in the accuracy metric as seen in Fig. 2,3,4,5, and 6. In general, the accuracy metric measures the fraction of all instances that are correctly categorized. But here, this metric value is fake if used alone for evaluation [28]. As due to the imbalance of data, the classifiers tend to get all majority samples as correct and all minority samples as incorrect. It's also noticed that the performance of the

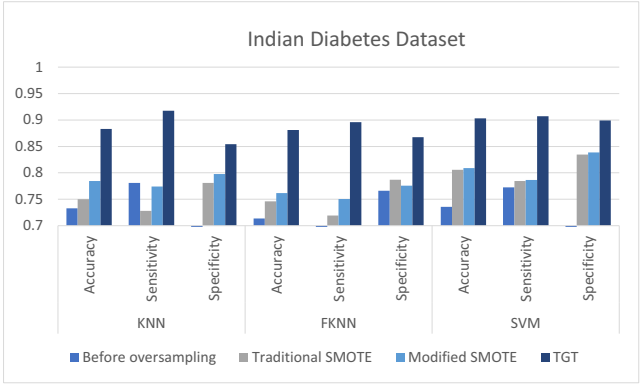

Fig. 2. The Evaluation Metrics of Indian Diabetes Dataset with the Three Classifiers.

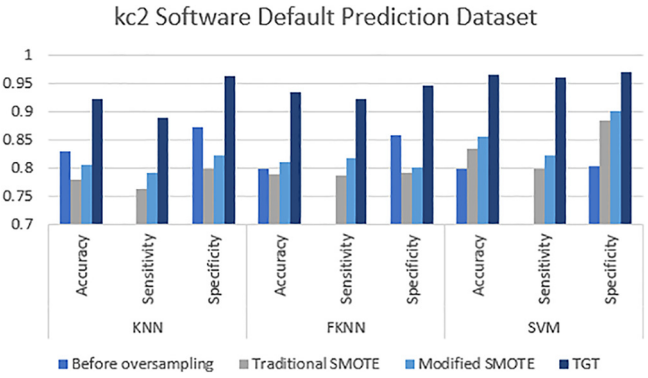

Fig. 3. The Evaluation Metrics of kc2 Software Default Prediction Dataset with the Three Classifiers.

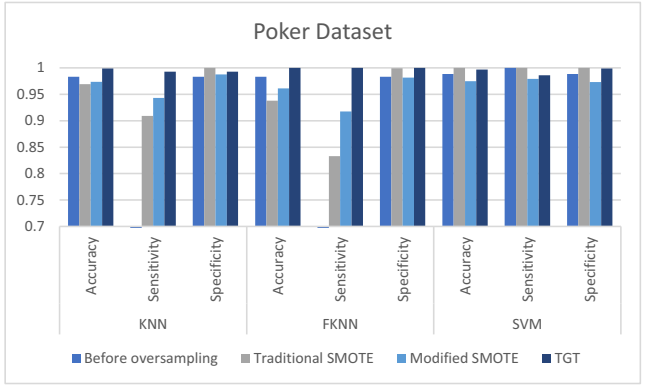

Fig. 4. The Evaluation Metrics of Poker Dataset with the Three Classifiers.

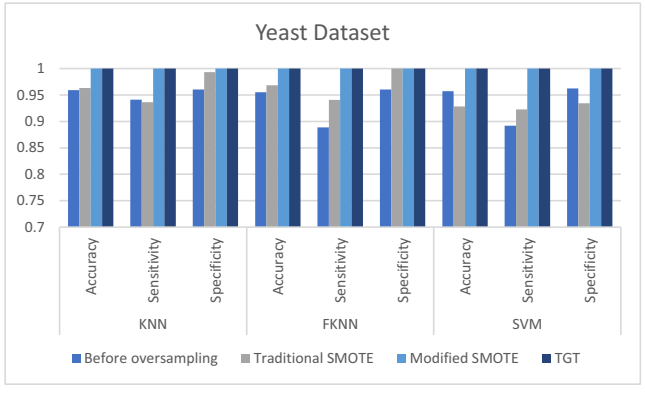

Fig. 5. The Evaluation Metrics of Yeast Dataset with the TThree Classifiers.

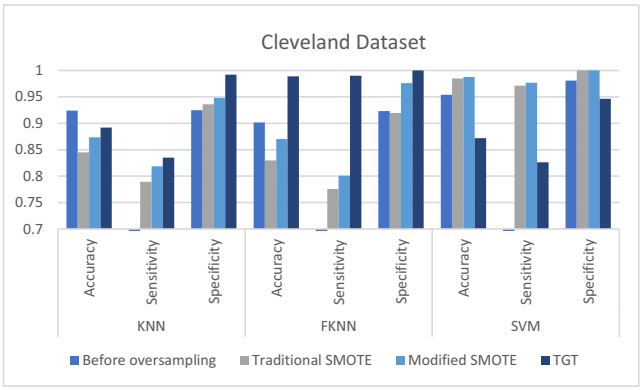

Fig. 6. The Evaluation Metrics of Cleveland Dataset with the Three Classifiers.

algorithm is variant on the five datasets. This may be related to the ratio between the majority and minority classes in the original data. The results show that the proposed technique performs extremely well when the ratio between the majority and minority classes in the original data is high as in Indian Diabetes and Kc2 Software Fault Prediction datasets.

The outperformance of the proposed technique against SMOTE algorithm [7] and modified SMOTE [13] may be related to the difference between how the three methods work. Traditional SMOTE algorithm generates synthetic samples in the space of the minority data space, the Modified SMOTE algorithm generates synthetic samples in the space between minority and majority data. Both methods generate a new synthetic sample by analyzing a randomly chosen sample of minority class. On the other hand, TGT generates the synthetic samples guided by the classification rules of the minority class derived by training the dataset of the decision tree. After the guided generation, these generated samples will be verified using the well-trained neural network to assure that all synthetic samples belong to minority class. Otherwise, they will be discarded. Those double-checked new samples generated by the guided adversarial oversampling technique have led to better classifications which proves our initial argument. In addition, the interaction between explainable and non-explainable classifier and the analysis of the whole dataset has shown to be effective enough to generate new synthetic samples better than those generated by SMOTE and modified SMOTE which depend on the analysis of individual samples to generate new samples.

5. Conclusion

This paper presented an adversarial guided oversampling technique (TGT) for handling the imbalanced datasets. The proposed technique utilizes two classifiers to extract and model the knowledge about the minority class data. A decision tree is trained on the given data to model the minority class data as set of classification rules where those rules are used to generate new samples of minority class. Then, a neural network is trained on the given data and used to verify that all the generated samples belong to the minority class data distribution. The proposed technique showed a higher performance when was evaluated against standard and recent data oversampling techniques over different datasets and using different classifiers.

Declaration of Competing Interest

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

References

[1] Fernández A, García S, Galar M, Prati RC, Krawczyk B, Herrera F. Learning from imbalanced data sets. Springer; 2018.
[2] Fernández A, del Río S, Chawla NV, Herrera F. An insight into imbalanced big data classification: outcomes and challenges. Complex & Intelligent Systems 2017;3(2):105–20.
[3] Rout N, Mishra D, Mallick MK. Handling imbalanced data: a survey. In: International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications. Springer; 2018. p. 431–43.
[4] Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A. A comparative study of data sampling and cost sensitive learning. In: 2008 IEEE International Conference on Data Mining Workshops. IEEE; 2008. p. 46–52.
[5] S. J. Dattagupta, "A performance comparison of oversampling methods for data generation in imbalanced learning tasks," 2018.
[6] S. Roshanfekr, S. Esmaeili, H. Ataeian, N. Maleki Khas, and A. J. a. Amiri, "UGRWO-Sampling: A modified random walk under-sampling approach based on graphs to imbalanced data classification," p. arXiv: 2002.03521, 2020.
[7] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer W P J J o a i r. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 2002;16:321–57.
[8] Barua S, Islam MM, Murase K. A novel synthetic minority oversampling technique for imbalanced data set learning. In: International Conference on Neural Information Processing. Springer; 2011. p. 735–44.
[9] Zheng Z, Cai Y, Li YJC. Informatics, Oversampling method for imbalanced classification. Computing and Informatics 2016;34(5):1017–37.
[10] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional random oversampling for imbalanced text classification," in Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016, 805-808.
[11] J. Mathew, C. K. Pang, M. Luo, W. H. J. I. t. o. n. n. Leong, and l. systems, "Classification of imbalanced data by oversampling in kernel space of support vector machines," IEEE transactions on neural networks and learning systems, vol. 29, no. 9, 4065-4076, 2017.
[12] Huang Z, Yang C, Chen X, Huang K, Xie YJNC, Applications. Adaptive oversampling method for classification with application to imbalanced datasets in aluminum electrolysis. Neural Computing and Applications 2019:1–17.
[13] A. Mahmoud, El-Kilany, A., Ali, F. , Mazen, S., "A Novel Oversampling Technique to Handle Imbalanced Datasets " in The 34th ECMS international conference on modeling and simulation, United Kingdom, 2020, vol. 34, no. 1, 177-182
[14] Lin W-C, Tsai C-F, Hu Y-H, Jhang J-S. Clustering-based undersampling in class-imbalanced data. Inf Sci 2017;409:17–26.
[15] Douzas G, Bacao F. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Syst Appl 2018;91:464–71.
[16] Zhang Y, Li X, Gao L, Wang L, Wen LJJoms. Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. J Manuf Syst 2018;48:34–50.
[17] Tao X et al. Real-value negative selection over-sampling for imbalanced data set learning. Expert Syst Appl 2019;129:118–34.
[18] B. Krawczyk, M. Koziarski, M. J. I. t. o. n. n. Woźniak, and l. systems, "Radial-based oversampling for multiclass imbalanced data classification," IEEE transactions on neural networks and learning systems, 2019.
[19] Yan YT, Wu ZB, Du XQ, Chen J, Zhao S, Zhang YPJIJoAR. A three-way decision ensemble method for imbalanced data oversampling. Int J Approximate Reasoning 2019;107:1–16.
[20] S. S. Mullick, S. Datta, and S. Das, "Generative adversarial minority oversampling," in Proceedings of the IEEE International Conference on Computer Vision, 2019, 1695-1704.
[21] Vanschoren J, Van Rijn JN, Bischl B, Torgo LJASEN. OpenML: networked science in machine learning. ACM SIGKDD Explorations Newsletter 2014;15 (2):49–60.
[22] Alcalá-Fdez J et al. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing 2011;17.
[23] Ranawana R, Palade V. Optimized precision-a new measure for classifier performance evaluation. In: 2006 IEEE International Conference on Evolutionary Computation. IEEE; 2006. p. 2254–61.
[24] Hossin M, Sulaiman MJIJoDM, Process KM. A review on evaluation metrics for data classification evaluations. International Journal of Data Mining & Knowledge Management 2015;5(2):1.
[25] Gu Q, Zhu L, Cai Z. Evaluation measures of the classification performance of imbalanced data sets. In: International symposium on intelligence computation and applications. Springer; 2009. p. 461–71.
[26] Hossin M, Sulaiman M, Mustapha A, Mustapha N, Rahmat R. A hybrid evaluation metric for optimizing classifier. In: 2011 3rd Conference on Data Mining and Optimization (DMO). IEEE; 2011. p. 165–70.
[27] N. J. I. l. F. Japkowicz, algorithms, and applications, "Assessment metrics for imbalanced learning," Imbalanced learning: Foundations, algorithms, and applications, 187-206, 2013.
[28] J. Akosa, "Predictive accuracy: a misleading performance measure for highly imbalanced data," in Proceedings of the SAS Global Forum, 2017, 2-5.