Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 7, Issue. 12, December 2018, pg.277 – 290

An Improved Synthetic Minority Over Sampling Technique for Imbalanced Datasets Classification

A.Bhuvaneswari[1], Dr. R.Manicka Chezian[2]
[1]Department of Computer Science & Bharathiar University, India
[2]Department of Computer Science & Bharathiar University, India
[1] bhuvana.aruchsamy@gmail.com; [2] chezian_r@yahoo.co.in

Abstract— The fast growing of the world data availability in many large-scale, complex, and networked systems, such as surveillance, security, Internet, and finance, it becomes critical to advance the fundamental understanding of knowledge discovery and analysis from raw data to support decision-making processes. The problem of learning from imbalanced data is a relatively new challenge that has attracted growing attention from both academia and industry. The distribution between the samples of the majority and minority classes. The minority instance of the dataset has a smaller number of instances than other categories, such a dataset may imply a problem of category imbalance, which means that the trained classification model is likely to be more likely to be discovered because of a few category instances. The reason for the low, but the minority category instance error is judged as the majority category instance. It is not a solution to balance the distribution between artificial and minority data examples. Many algorithms have been designed based on this concept. The propose an improved algorithm ISMOTE to solve the category imbalance problem. ISMOTE differs from previous algorithms in that it differs from previous algorithms in that it does not consider only a few categories of distribution, but also measures the relative advantages of minority categories and multiple density distributions, and uses this as a basis for weight measurement. In addition, our method will choose to generate man-made with a few categories of instances and most recent references. This method can reduce the difficulty of classifier learning due to the generation of erroneous artificial data instances, and the artificial examples through this method can better help the classifier to learn

Keywords— Movie Recommendation System, Memory-Based Collaborative Filtering, Model-Based Collaborative Filtering, Stochastic Gradient Descent

I. INTRODUCTION

Information mining is the way toward separating designs from information. It is the analysis of observational data sets to find unsuspected associations and to sum up the data in new ways that are both clear and useful to the data owner. It is a prevailing technology which has great potential to help companies that focus on the most important information in their data warehouses.The classification techniques usually assume a balanced class distribution (i.e. their data in the class is equally distributed). More often than not, a classifier performs well when the order procedure is connected to a dataset equitably appropriated among various classes. Yet, numerous genuine applications confront the imbalanced class conveyance issue. In this circumstance, the grouping errand forces challenges when the classes present in the preparation information are imbalanced.

The imbalanced class dissemination issue happens when one class is spoken to by a substantial number of models (majority class) while the other is spoken to by just a couple (minority class). In this case, a classifier usually tends to predict that samples have the majority class and completely ignore the minority class.The class imbalance problem can appear either from between classes (inter class) or inside a solitary class (intra class). Inter-class imbalance refers to the case when one class has larger number of example than another class.One of the primary problems when learning with imbalanced data sets is the related absence of information where the quantity of tests is little. In a characterization errand, the span of informational index has a critical job in building a decent classifier.

The complexity is an important factor in a classifier ability to deal with imbalance problem. Idea intricacy alludes to the detachment level between classes inside the information. Straight detachment between classes implies the classifier not at risk to any measure of irregularity. On other hand, the high complexity refers to occurs high overlapping between the two classes that means the classifier susceptible to any amount of imbalance.Class imbalance learning refers to learning from data sets that exhibit significant imbalances among or within classes. Any data set with uneven data distributions can be considered imbalanced. The common understanding about "imbalance" in the literature is concerned with the between-class imbalance, in which case some classes of data are highly under-represented compared to other classes. By tradition, we call the classes having more models the lion's share classes, and the ones having less precedents the minority classes. Misclassifying an example from the minority class is usually costlier. For example, in a defect prediction problem from software engineering, codes with defects are much less likely to occur than codes without defects.

A learner's sensitivity to the class imbalance was found to depend on the data complexity and the overall size of the training set. Data complexity comprises issues such as and small disjuncts. The level of overlapping among classes and how the minority class examples disperse in information space exasperate the negative impact of class imbalance. The small disjuncts problem is also associated with the within-class imbalance. In terms of the size of the training data, a very large domain has a good chance that the minority class is represented by a reasonable number of examples, and thus may be less affected by imbalance than a small domain containing very few minority class examples. At the end of the day, the uncommonness of the minority class can be in a relative or total sense as far as the quantity of accessible models.

II. RELATED WORK

There few works handle the imbalance dataset learning algorithm to reduce information loss during feature space projection, this study proposes a novel oversampling algorithm, named minority oversampling in kernel adaptive subspaces (MOKAS), which exploits the invariant feature extraction capability of a kernel version of the adaptive subspace self-organizing maps.One approach is one-class classifiers, which tries to describe one class of objects (target class) and recognize it from every single other question (outliers).In this existing paper, the performance of One-Class SVM, adaptation of the popular SVM algorithm, will be analyzed. Another system is cost-delicate realizing, where the expense of a specific sort of error can be not quite the same as others, for instance by allotting a staggering expense to mislabeling a sample from the minority class.

Another existing algorithm is sophisticated online banking fraud involves multiple resources, including human wisdom, computing tools, web technology and online business systems. The instant and effective detection of such fraud challenges existing fraud detection techniques and systems. In this paper, we report our study and practices in the real world. A systematic online banking fraud detection approach is introduced. Its framework takes advantage of domain knowledge, mixed features, multiple data mining methods, and a multiple-layer structure for a systematic solution. It includes three algorithms: contrast pattern mining, neural network and decision forest, and their outcomes are integrated with an overall score measuring the risk of an online transaction being fraudulent or genuine.

Some existing methods like SMOTE that have been shown to be effective in addressing the class imbalance problem we also proposed a new under sampling technology, namely, instance-remove algorithm.The classifiers for testing are FLDA and linear SVM. The most commonly used technique is over/under sampling for handling the class imbalance problem (CIP) in various domains. In this examination, we review six surely understood testing strategies and look at the exhibitions of these key methods i.e., Mega-slant Diffusion Function (MTDF), Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling approach (ADASYN), Couples Top-N Reverse k-Nearest Neighbor (TR-KNN), Majority Weighted Minority Oversampling Technique (MWMOTE) and Immune centroids oversampling technique (ICOTE).Thus, it does not seem fair to directly relate class awkwardness to the loss of execution of learning algorithms.

Class imbalance is often reported as an obstacle to the induction of good classifiers by machine learning algorithms. In any case, for a few areas, machine learning algorithms can accomplish important outcomes even within the sight of exceedingly imbalanced datasets. The enhanced structure preserving oversampling (ESPO) technique and synergistically combine it with interpolation-based oversampling. ESPO is utilized to produce a vast level of the manufactured minority tests dependent on multivariate Gaussian dissemination, by evaluating the covariance structure of the minority-class tests and by regularizing the unreliable eigen spectrum.Although researchers have introduced many methods to deal with this problem, including resampling techniques and cost sensitive learning (CSL), most of them focus on either of these techniques.

In any case, similar to the ordinary SVM calculation, FSVMs can likewise experience the ill effects of the issue of class imbalance. The FSVMs for CIL (called FSVM-CIL), which can be used to handle the class imbalance problem in the presence of outliers and noise.FSVMCIL method is a very effective method for CIL, especially in the presence of outliers and noise in datasets.FSVM-CIL, for learning from imbalanced datasets in the presence of outliers and noise. In this method, we assign fuzzy-membership values for training examples in order to handle both the problems of class imbalance and outliers/noise. Imbalanced information order is normally assessed by measures, for example, G-mean and AUC rather than exactness. However, for many classifiers, the learning process is still largely driven by error based objective functions. We use the measure directly to train the classifier and discover the optimal parameter, ratio cost and feature subset based on different evaluation functions like the G-mean or AUC. Distinctive measurements can reflect diverse viewpoint execution of classifiers. The combination of SMOTE and under-sampling performs better than plain under-sampling. SMOTE was tried on an assortment of datasets, with fluctuating degrees of lopsidedness and shifting measures of information in the preparation set, hence giving a various testbed.

Majority part Weighted Minority Oversampling Technique (MWMOTE), is introduced for productively dealing with imbalanced learning issues. MWMOTE first identifies the hard-to-learn informative minority class samples and assigns them weights according to their Euclidean distance from the nearest majority class samples.Similarly, the samples of the small-sized groups are given higher weights for decreasing inside class imbalance. The synthetic sample generation technique of MWMOTE uses a clustering approach.over-sampling technique, called MDO (Mahalanobis Distance-based Over-sampling technique), generates synthetic samples which have the same Mahalanobis distance from the considered class mean as other minority class examples.MDO over-samples the minority classes by considering each candidate minority class sample and generating a new synthetic instance which has equal Mahalanobis distance from the considered class mean with the candidate sample. It generates synthetic samples toward the variation of the corresponding class and helps in reducing the overlapping between different class regions. In multiclass problems over-lapping between different class regions is a prevalent issue.two oversampling approaches, namely, RACOG and wRACOG, to generating and strategically selecting new minority class data points.

III.PROPOSE APPROACH

The k-NN based method judges what kind of processing action should be taken by measuring the information in the most K instances near a few category instances by measuring the minority class. The range will be between a few category instances and one of its most recent K instances. Using the interpolation method to generate an artificial instance requires two talents to be selected, and the clustering-based method first determines whether a few categories belong to the same group to obtain group information, so the clustering-based method can easily select two instances of the same group. Execution of the interpolation method, in this way, avoids the selection of two instances belonging to different groups to produce man-made.

- The first item is the lack of information caused by the small sample size of the training set. The insufficient number of data will increase the difficulty of the learning algorithm to find the correct classification rules. Increase the number of instances of the training set, which can reduce the error rate of the unbalanced classification model.
- The second item is Class overlapping. The minority category data instances are scattered among most categories. Increasing the complexity of the dispersion according to will increase the impact of the category imbalance on the effectiveness of the classification model. The proves that there is a certain degree of correlation between category overlap and category imbalance. When category overlap occurs, some data will become cumbersome or unhelpful classes learn the boundaries between categories (or rules).

- In the third category, the distribution disparity is small. When the clustering characteristics of a few category instances are not obvious, the minority data instances may have multiple clusters at the same time. The distribution of such category data instances in the entire feature space is often unbalanced, which will result in a few clusters of sub-category data instances that are not representative, which in turn affects the correctness of the classifier's decision-making. The stated that we can improve the performance of the classifier by changing the unrepresentative minority clusters by taking back the samples after the use of the data.

In terms of k-NN based, use Borderline-SMOTE and ADASYN as examples of how we describe k-NN based methods and how these two ineffective methods make these two ineffective.

The improved oversampling method ISMOTE for artificially generating a few class instances is propose Improve the mechanism for selecting reference samples and generate artificial instances. The propose ISMOTE contains three key steps. In the first phase, ISMOTE identifies all the noise instances from the original minority collection from the original minority collection, and establishes a minority collection that excludes the noise instances, and establishes exclusion. A small set of categories s_n of the instance, and then use this set to find the boundary majority category s_m^b. In the second stage, each member of s_n will be assigned a weight wi and a distance limit di, the size of wi is positively related to its importance in the data, and the size of di is related to the distribution density of the reference member. In the third stage, ISMOTE uses each member of s_n as a reference instance, and $x_i \in s_n$will generate artificial instances with most of the most recent class instances from the nearest class instance $y_i \in s_m^b$. The quantity is proportional to the weight of the $x_i \in s_n$ generated by the execution of the interpolation method. The method in which the ISMOTE method will be fully rendered will be fully rendered in algorithm.

A. Collection S_{min} and Collection S^b_{maj}

The noise area is usually located outside of a cluster of far fewer category instances. The splitter will determine that these instances are in most categories because they are surrounded by these. Security zones are typically located within a cluster of a few category instances. The splitter is easier to identify the security zone because it has a sufficient number of instances. However, for a category imbalance problem, the security zone with fewer instances is likely to not contain enough classifiers to learn the minority. Therefore, the method of creating artificial instances should have the ability to detect the noise region, only for the operation of a few categories of instances in the security zone and to avoid the effects of minority instances on the noise zone. In addition, for the artificial instance method, most of the categories are not useful. It is closer to the boundary of a few category instances. If there is such information, it can help the algorithm save a lot of time. At ISMOTE, it builds a set, and it provide enough information to create artificial instances by constructing a collection of $\boldsymbol{S^b_{maj}}$(a few categories that remove the noise instances) and a collection of $\boldsymbol{S^b_{maj}}$ (boundary majority instances)

This entire $\boldsymbol{S_{min}}$, $\boldsymbol{S^b_{maj}}$ is described as follows:

ISMOTE first filters out the noise from the original collection of minority instances (here the noise is defined as the most recent K-bit neighbor instances of a few class instances, and then the minority class $\boldsymbol{S_{min}}$ has been

removed. In order to do it Distance, in order to find the nearest K-bit neighbors of a few category instances, and then determine whether the K-bit neighbor instances are all major categories, and then remove the less-category instances. This removal action represents an impossible event in the process of generating noise into the artificial data. In this way, ISMOTE will be able to remove the noise from the training set and will be able to remove the noise from the training set.

ISMOTE will establish a nearest K-bit majority class instance neighbor set $\hat{N}_{maj}(x_i)$ for each of the minority

class instances$x_i \in S_{min}$. When the K parameter is set to be relatively small, each of the majority class instances

in $\widehat{N}_{maj}(x_i)$ will be able to be assumed that the position of its distribution is closer to the decision boundary region. Then we will perform the union action of all$\widehat{N}_{maj}(x_i)$, in this way we can get the boundary instance set S^b_{maj} of most categories. When a small number of class instances are very close to each other or are far away from most class instances, then all minority class instances are likely to have no majority neighbor class instances. This situation caused Borderline-SMOTE or The ADASYN algorithm fails to achieve the capabilities that its designers want them to achieve. However, ISMOTE still can get S_{min} and S^b_{maj} in good working order, because it have considered this possibility from the beginning, so we can't create artificial data examples because of the above situation.

B. Get weight d and Distance d

ISMOTE obtains the information necessary to produce an artificial instance and obtains the information necessary to produce an artificial instance. The valuable minority instances and most of the category boundary instances are grouped into S_{min} and S^b_{maj}. However, even members of the S_{min} group of minority instances that we consider valuable are not necessarily the same in value, and some instances may provide more useful information than other provided messages. Therefore, it is necessary to assign rights to each of the minority category instances based on their importance. An example with a relatively large weight means that it needs to produce multiple artificial loops around it. This is due to the fact that it provides less information on the minority categories. Some oversampling methods use most instances of recent neighbors to judge the importance of their minority instances; this mechanism is inadequate and appropriate in many cases. So ISMOTE uses a new machine to use the new mechanism to assign the appropriate parameters to a few class instances. The fifth to seventh steps in algorithm show that our ISMOTE will calculate the importance $w_i$and the distance limit di for each minority instance $x_i \in s_{min}$ for each minority instance. This weight$w_i$calculation concept is based on the following three important observations:

The first observation: A data instance with a distribution location closer to the decision boundary provides more information than the distribution location is farther away from the impact of the data instance on the decision boundary. This observation implies that higher weight coefficients should be given for data instances whose distribution locations are closer to the decision boundary than to distant data instances. If a few category instances A and B are distributed in the outer adjacent majority category instances of the minority group distribution, the minority class instances C and D are located outside the minority group distribution away from the majority category instance. This represents the location of instances A and B relative to instance C and the location of instance D is closer to the decision boundary. Therefore, instance A and instance B should be more informative than instance C and instance D. Similarly, instance C is more informative than instance D. This situation shows that instance A and instance B should be given a higher importance weight than instance C and instance D. Of course, the weight of instance C should also be higher than the weight of instance D.

The second observation: The distribution of a few category instances to the surrounding minority instances is sparse and more important than the more densely distributed minority instances. The distribution of a few categories of instances in the training set may be uneven. From the point of view of producing artificial instances, a few instances with sparse surrounding distributions will be more important than the denser instances of the minority. This is because the more densely distributed instances contain more information than the more sparsely distributed instances. Therefore, we need to create more artificial instances for a few sparsely distributed instances of the surrounding area to increase the density of its surrounding distribution to reduce the imbalance within the category.

The third observation: If a minority instance is faced with a more densely distributed majority instance, its importance will be more important than a small number of class instances that are sparsely distributed to most instances. The minority group instance A and the minority category instance B belong to the same group size

and the same distance from the most recent category instances. However, the minority category distribution B faces the majority distribution density. A few category instances A are still higher. This imbalance in distribution density will make the classifier difficult to learn for a small number of class instances B. Therefore, the minority class instance B should be more important than the minority class instance A in the generation of the artificial instance, so the weight of the instance B should be higher than the weight of the instance A.

The concept of distance limits $d_i$is based on the following two important observations, but before describing

these observations, we must first describe that our ISMOTE will use a few category instances and multiple examples with a few category instances and multiple reference examples. Generate an interpolation method. It assumes that the most difficult to learn for each of the few categories is the direction of the most recent category instance. A very intuitive concept is that if artificial objects are placed between a few category instances and most of the most recent category instances, these artificial instances will easily change the decision boundary of the classifier for minority cognition, and the artificial instance distribution is located far from most categories. The closer the instances are, the more decision boundaries are likely to broaden a few categories.

Two important observations of d_i.

The first observation: It can find out based on the description of the second observation of the above weights. One thing that we want to produce artificial instances should be limited. The artificial instance generated based on a small number of category instances B as a reference example, the distance between it and the artificial instance should be compared with the distance between the minority class instance A and the artificial instance generated based on the minority class instance A as a reference instance. I have to come far. The reason is simple, because the density of the location of instance B is actually looser than the density of the location of instance A, so we should not be as arbitrary as the algorithms of other artificial instances. It is assumed that the areas in which instance A and instance B are located are evenly distributed.

The second observation: it can find out from the description of the third observation of the above weights. One thing is that when we configure the artificial instance to be between a few category instances and most recent category instances, then if it is a minority When the density of most category instances faced by category instances is high, in order to balance the learning pressure faced by minority instances, it may be appropriate to give a wider decision boundary for a few category instances. The states of instance A and instance B are the same, and the difference is only the density of the majority of the instances facing each other. We can clearly realize that the difficulty of learning in instance B will be much more difficult than that of example A. Thus, the artificial instance is placed between instance B and its most recent category instances and the artificial instance generated according to instance B should be closer to the most recent category instance than the artificial instance generated from instance A and the nearest most category instance.

The ISMOTE we propose for the weights for the weight w_i and the distance limit d are specifically designed

to take into account the above observations. The calculation of the weight w_i and the distance limit d_i.

Calculate each of the minority class instances $x_i \in S_{min}$ and most of the class instances closest to it the distance r between $y_i \in S_{maj}^b$.The majority of the class instances y_i closest to the minority class instance $x_i \in S_{min}$ are guaranteed to be one of the members of the set S_{maj}^b, since the set S_{maj}^b is the set of most class instances where all the distribution locations are closer to the decision boundary.

Take a few category instances $x_i \in S_{min}$ as the center of the circle and distance r as the radius. Calculate the number of all S_{min}members C_{min} (without the center of the circle) within this circle. For the same reason, the

most class instance $y_i \in S_{maj}^b$ is taken as the center of the circle, and the distance r is taken as the radius. Calculate the number of all major class instances within the circle, C_{maj} (remove the center of the circle). The number of C_{min} can be used to measure the density of the distribution of a few category instances. If the number of C_{min} is larger, it represents the area where the minority category instances are densely distributed in the $x_i \in S_{min}$ position. At the same time, the number of C_{maj} can be used to measure the density of the distribution of most category instances. If the number is larger, it represents the majority of the class instances. $y_i \in S_{maj}^b$ is a region with a dense distribution of most category instances.

After obtaining the C_{min} and C_{maj} of each minority class instance $x_i \in S_{min}$ in step 2, the weight wi of this minority class instance $x_i \in S_{min}$ is calculated. The greater the weight wi, the more important it is for this minority class instance $x_i \in S_{min}$. The calculation of the weight w_i is as follows.

$$w_i = \frac{C_{maj} + 1}{C_{min} + 1} \qquad (1)$$

According to observations on the weight w_i, a few category instances with a smaller group instance group distribution than the outer ones should have a higher weight than the inner ones. And our ISMOTE can do this very well. Taking the C_{maj} of instance A and instance C should be similar, but the number of C_{min} instance A should be smaller than the number of C_{min} instance C, because it is closer to most recent class instances, so the distance r will be smaller, so fewer instances of the category that can be considered will be less. Therefore, the weight w_i of the instance A will be larger than the weight w_i of the instance C.

The weight w_i, the minority class instances belonging to the sparsely distributed minority group should have a higher weight w_i than the weight w_i of the minority class instances belonging to the more densely distributed minority group. And our ISMOTE can do this very well. The distances between instance A and instance B are approximately the same from their respective most recent class instances, but it can be clearly seen that the number of C_{min} in instance B will be smaller than the number of C_{min} in instance A. C_{maj} of instance B The number will be similar to the number of C_{maj} in instance A, so instance B will have a larger weight than instance A. weight w_i, a few category instances located near the most densely populated majority category instances should have a weight w_i that is smaller than a few category instances located near most of the category instances that are sparsely distributed in most categories. It is important to come. And our ISMOTE can do this very well. The number of C_{min} in instance A and instance B will be similar, but the C_{maj} of instance B will be

larger than the C_{maj} of instance A of C_{maj}, so the weight of instance B will be better than that of instance A. Still bigger.

After obtaining the C_{min} and C_{maj} of each minority class instance $x_i \in S_{min}$ in step 2, the distance limit di of the minority class instance $x_i \in S_{min}$ is calculated, and the distance limit di value is larger, the position representing the artificial instance may be farther from the reference minority class instance. Far, and vice versa. The calculation of the distance limit d_i is as follows.

$$d_i = \frac{C_{maj} + 1}{C_{maj} + 1 + C_{min} + 1} \qquad (2)$$

According to observation of the distance limit d_i, if the distribution density of the location of the reference instance of a few categories is higher, the closer the artificial instance should be to the reference instance of this minority category. And our ISMOTE can do this very well. It can clearly see the minority category instances in the area where instance A is located the density of the cloth is higher, and the number ofC_{min} in the example A is larger than the number of C_{min} in the example B. Therefore, under the condition that the number of C_{maj} is similar, the distance limit di of the instance A is smaller than the distance limit of the instance B. Still small.

The distance limit d_i, the reference example of a few category instances is the most If the distribution density of most of the category instance groups is higher, it means that this reference instance has it has a large learning difficulty, so it needs to adjust more decision boundaries, which means should allow the distribution of artificial instances to be able to be compared to a few categories of instances far and our ISMOTE can do this very well. It can clearly see that the distribution density of most category instances faced by instance B is more realistic. Example A is high, and the number of C_{maj}in instance B is larger than the number of C_{maj} in instance A. Therefore, in the case where the number of C_{min} is similar, the distance limit d_i of the instance B is larger than the distance limit d_i of the example A.

C. Generation of artificial instances

The problems that arise with many of the KNN-based and clustering-based man-made instances. To solve these problems, we have taken an improved approach, using two different categories of examples as a reference to perform the interpolation method. At first glance, the interpolation of two different categories seems to be unreasonable. At present, most of the artificial instance generation methods hope that the instance is very likely to exist, so intuitively select two minority instances as Refer to the examples to create man-made (although not necessarily in line with expectations). However, we believe that the most important issue of the category imbalance problem should be to solve the problem that the classifier is not efficient in learning a small number of instances. Therefore, ISMOTE does not produce the most likely target of real existence. Whether the artificial instance can really improve the learning efficiency of the minority category is the highest criterion.

It assumes that for each minority category instance, the most difficult direction of learning is near most of the category instances are in the same orientation, so we use a few category instances x_i∈S_min and most of the most recent category instances y_i∈S_maj^b as reference examples and generate artifacts through interpolation. This approach is significantly better than the KNN-based approach in that the user's method is

better because the user's nearest neighbor parameter K does not affect the action of the artificial instance. A better practice of this approach than the clustering-based approach is that the location of the artificial instance is absolutely unlikely to be located in a cluster of most category instances.

The KNN-based and clustering-based artificial instance generation methods. The artificial instance generation method is even better. The KNN-based artificial instance method does not consider the location distribution of the nearest K neighbor instances. The nearest K neighbor instances do not necessarily belong to the same group as the reference. Case A chooses to perform the interpolation method with instance B, which does not belong to the same group. It can be seen that the artificial instance generated by the artificial instance method of the KNN-based artificial instance method may not be helpful for a few categories. However, our method will configure the artificial instance in the orientation that is most difficult for a few categories. The artificial instance of the artificial instance C will be located in the instance will be located in the instance A and the distance instance and the distance instance and distance in this way, the most recent category instances of most recent category instances of instance A are used to help the classifier identify minority instance instances.

The KNN-based method is that another problem is that when the user-defined parameter K is set too small, it is likely to produce artificial instances that overlap with the reference instance. It can be seen that the artificial example C almost overlaps with the artificial example A, however, the artificial example C produced by our method will be able to better protect the minority instance. Adding User-Defined Parameters Increasing the number of user-defined parameters K can avoid the problem of the KNN-based method. The external problem, the KNN-based method may produce the wrong artificial instance, an example (assuming K=20), instance A may be executed with an instance of a different group and possibly with an instance B of a different group. In the case of a man-made instance, this execution of the interpolation method produces a man-made instance that will fall in most category areas. However, our method cannot have such a problem. It is impossible for Real A to choose an interpolation method from a minority instance across its majority category, because we only allow instance A and Most recent class instances perform interpolation methods. In general, a generic K value is very difficult to find. But our approach avoids the problem of choosing the correct K value by separating the K value parameter from the processing that produces the artificial.

D. Algorithm implementation

Algorithm. ISMOTE (trainset, K)

Input:1. Train Set:

training set T∈K: The number of nearest neighbors.

Procedure Begin

1. Create an empty set S_{min}

2. Find the nearest K-bit class neighbor instance $\tilde{N}(x_i)$ for each minority instance $x_i \in T$. If the members in $\tilde{N}(x_i)$ are not all of the categories, add xi to the set S_{min}, otherwise treat it as noise (Noise).

3. Find each instance of a few categories $x_i \in S_{min}$ nearest K-bit majority category neighbor $\hat{N}_{maj}(x_i)$.

4. Do a union of $\hat{N}_{maj}(x_i)$. for all $x_i \in S_{min}$ and find the majority category set S_{maj} in the boundary area.

$$S_{maj}^{b} = \bigcup_{x_i \in \hat{N}_{maj}(x_i)} x_i$$

5. Taking the set S_{min} as the reference instance set (SEED SET), within the set S_{maj}^{b}, look for the most recent category instance $y_i \in S_{maj}^{b}$ of each instance $x_i \in S_{min}$ in S_{min}, when the most recent category instance of $x_i \in S_{min}, y_i \in S_{maj}^{b}$ is found At that time, the Euclidean distance of the two points is taken as the radius r, x_i and y_i are respectively used as the center point and the circle is drawn by the distance of the radius r, and the number of minority instances in the circle with xi as the center point is calculated (required in the set Within S_{min}), this number is indicated by C_{min}. Calculate the number of most category instances in the circle with yi as the center point, this number is indicated by C_{maj}.

6. Give each $x_i \in S_{min}$ an importance weight w_i

$$w_i = \frac{C_{maj} + 1}{C_{min} + 1}$$

7. Give each $x_i \in S_{min}$ a distance limit d_i

$$d_i = \frac{C_{maj} + 1}{C_{maj} + 1 + C_{min} + 1}$$

8. For each $x_i \in S_{min}$

Calculate the number of artificial instances that x_i needs to generate C_i = (number of most class instances - number of minority class instances) $* \frac{w_i}{\Sigma w_i}$

The interpolation method is performed by two points of the most recent class neighbor instance y_i of x_i and x_i, and an artificial minority class instance new Node C_i is generated. Let n be the number of features of the data set

For i = 1 to C_i

Initialize d if to n-dimensional

For attr = 1 to n

$$\mathrm{dif[attr]} = \mathrm{rand(1)} * \mathrm{d_i} * (\mathrm{y_i[attr]} - \mathrm{x_i[attr]}$$

End LOOP

$$newNode[i] = x_i + dif$$

End LOOP

Add the generated artificial instance to the training set

End LOOP

End

The clustering-based approach is that the problem with the method is that the clustering algorithm's clustering results may be wrong. The clustering algorithm is an unsupervised algorithm, an unsupervised algorithm that does not have a category label. For the clustering algorithm, whether to classify instances in the same group, whether to classify instances in the same group is based on whether the distance between the instances is close enough. The group to which instance A belongs and the group to which instance B belongs are grouped because they are too close to each other, so it is possible that the group is judged to be the same by the clustering algorithm. So, the clustering-based method might choose an instance. The method might choose instance A and instance B as reference instances, as a reference instance, and produce a human instance C that might fall in most category areas. However, our method is unlikely to have this problem. Because of reference to example A, it is only possible to perform interpolation methods with most of the class instances D closest to it, so artificial instance C cannot fall within most categories. In addition, the clustering algorithm will also be a problem for the distance between instances. Its distance between instances will also be a problem for the distance between instances, because this setting involves how close the two groups are to each other. Into a group.

The clustering-based approach is that even if the clustering results are correct, another problem with the method is that even if the clustering results are correct, the same group the instance distribution does not necessarily present a normal distribution (elliptical distribution), which may result in artificial instances falling into most category areas. The clustering-based method may choose an instance method. It may select instance A and instance B as reference instances, and generate artificial instances that may fall into most category regions to produce artificial instances C that may fall into most category regions. However, our method can completely avoid this problem, because our method is based on a few category instances and the nearest multi-category instance, so it is absolutely impossible for artificial instances to fall in most category areas

IV. RESULT AND DISCUSSION

To measure the efficacy of the proposed ISMOTE and compare its performance and compare its differences with SMOTE, Borderline-SMOTE, ADASYN, MSMOTE, MOKAS. The difference between the two. We collected twenty real-world data sets from the UCI website and the KEEL website for two statistical tests. Wilcoxon signed rank test and paired t-tests.

For the assessment of imbalances in two categories, we often refer to the lesser categories as Positive class and the larger categories to Negative classes. The confusion matrix is a very typical evaluation method. We show it in Table 1, the column represents the real category label, the real category label, and the row represents the other label predicted by the classifier. TP (True Positive) is a small number of categories that are correctly classified by the classifier. FN ((False Negative) is a few categories that are incorrectly misclassified by the classifier. FP (False Positive) is the most misclassified classifier by the classifier. TN (True Negative) is correctly classified by the classifier Most of the other places. In addition to using fusion matrix, there are several composite performance indicators calculated.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is the correct proportion of classifiers in all instances. In general, the higher the Accuracy, the better the performance of the measured algorithm. But it does not apply to category imbalances because the number of Positive class instances is less than the number of Negative class instances.

$$TP_{rate} = \frac{TP}{TP + FN} \text{ (also known as Recall)}$$

Recall is the correct proportion of the classifier in all Positive class instances. I.e. A small class of Accuracy.

$$FP_{rate} = \frac{FP}{TN + FP}$$

$FP_{rate}$is the proportion of classifier errors in all Negative class instances? A common example is a false alarm. The higher the FP_{rate}, the higher the number of false alarms may occur.

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

Precision is the correct proportion of the classifier class in all instances where the classifier is judged to be positives. The choice between Recall and Recall is that the positives instance is very expensive to be judged by the classifier. If so, it is better to use Recall.

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2}$$

AUC (the Area Under the Curve) is the area between the ROC curve and the coordinate axis. AUC stands for randomly selecting a positive instance and randomly selecting a negative instance, and then the classifier will then use this classifier to predict the correct ratio of this positive instance will be higher than the rate at which the classifier will predict the wrong instance prediction error. AUC is an indicator that is often used to measure the performance of classifiers. The larger the AUC value, the representative points

$$G - mean = \sqrt{PositiveAccuracy * NegativeAccuracy} = \sqrt{\frac{TP}{Tp + FN} * \frac{TN}{Tn + FP}}$$

G-mean reflects the ability of the classifier to balance the two, reflecting the ability of the classifier to balance the two. G-mean is a more comprehensive indicator of the performance of the classifier, because it considers both the classifier is Accuracy for the positive class instance and Accuracy for the Negative class instance. Therefore, the larger the G-mean indicator, the larger the indicator for the classifier, and the better the ability of the classifier to correctly judge the two types of instances.

$$F-measure = \frac{(1+\beta^2) * Recall * Precision}{\beta^2 * Recall + Precision}$$

F-measure parameters β is a user-adjustable parameter used to weigh Recall And the importance of the Precision two indicators, but often set to 1, representing Recall is as important as Precision (our experiment also sets β to 1). F-measure is a simultaneous consideration. It is a value that considers both Precision and Recall. If both Precision and Recall are high, F-measure will be very high. Therefore F- measure can be used as a measure of the strength of the classifier in dealing with the problem of unevenness. Can be used as a measure of the strength of the classifier in dealing with the problem of unevenness.

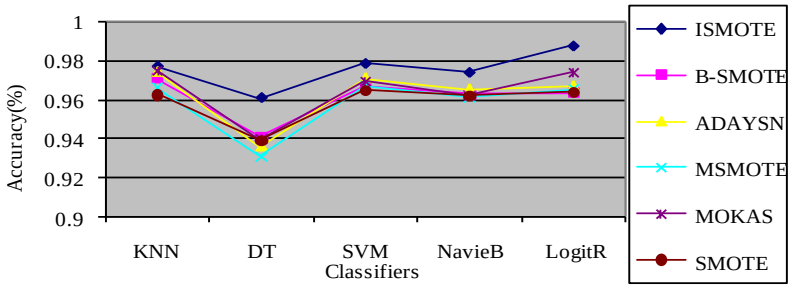

Figure 1 Comparison of different classifier with imbalance accuracy.

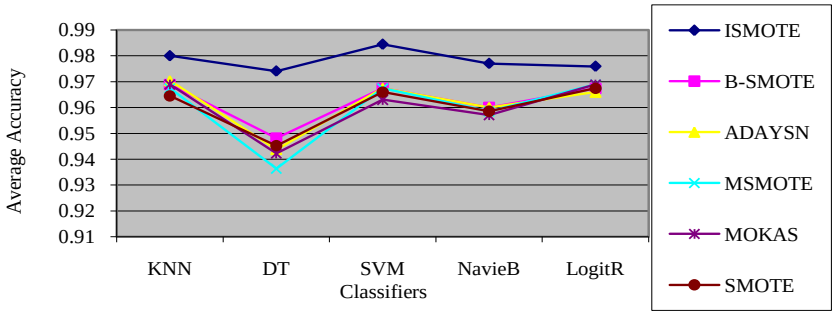

Figure 2 Comparison of different classifier with imbalance average accuracy.

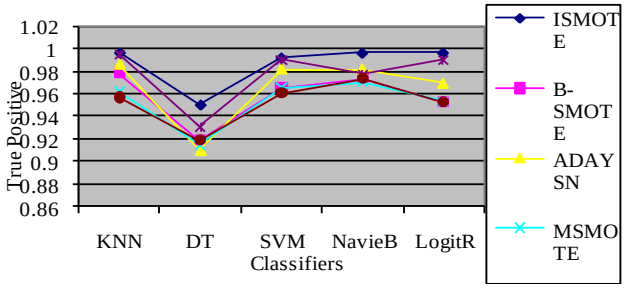

Figure 3 Comparison of different classifier with imbalance true positive.

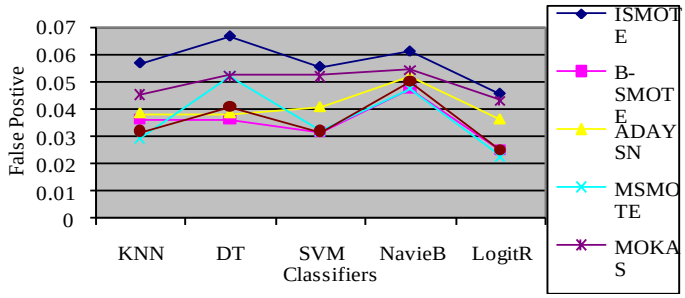

Figure 4 Comparison of different classifier with imbalance false positive.

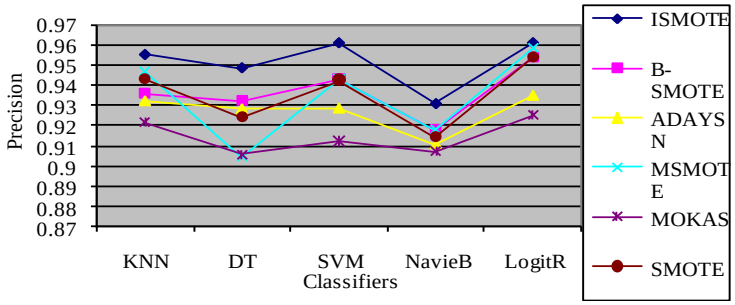

Figure 5 Comparison of different classifier with imbalance precision.

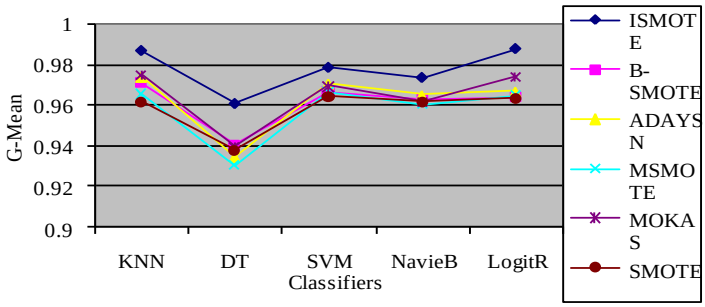

Figure 6 Comparison of different classifier with imbalance G-mean.

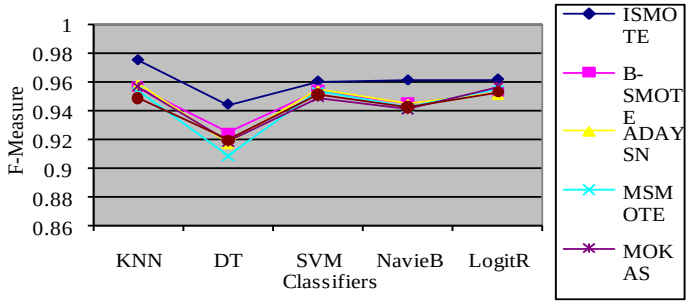

Figure 7 Comparison of different classifier with imbalance F-Measure.

V. CONCLUSION

ISMOTE chose to use a few category instances and most of the most recent category examples as a reference. This method guarantees that man-made has a certain degree of help in helping the classifier identify the minority. In this way, ISMOTE also solves the problem that the user needs to also solve the user's need to find the appropriate K value setting, because ISMOTE's dependence on the nearest neighbor parameter K for the nearest neighbor parameter is not as serious as the KNN-based artificial instance method. serious. The problem with the artificial instance method of the clustering-based artificial instance method is that the benchmark for selecting the reference instance is very dependent on the result of the clustering algorithm. Based on our experimental results, ISMOTE has better AUC values, F-measure values, and values, as well as G-means values relative to other artificial example methods. This is a good overall performance because of the value relative to other artificial instance methods. This is a good overall performance because ISMOTE improves the mechanism for selecting reference samples and improves the generation of artificial instances. These two mechanisms can ensure that ISMOTE can improve the correct rate of identifying the minority instances by the classifier, and as far as possible, it does not affect the correct rate of identifying the minority instances for the lifting classifier, and does not affect the classifier for most instances as much as possible. Judge. This statistical analysis will support our arguments. Although have successfully improved the classifier's judgment rate for a few, ISMOTE still has some areas for improvement. To improve the value of FP rate while maintaining the current advantage, or to combine the clustering algorithm to increase the information that can be referenced, the algorithm to increase the information that can be referenced, such as the group to which the instance belongs.

REFERENCES

[1] H. Parvin, B. Minaei-Bidgoli, H. Alizadeh, "Detection of cancer patients using an innovative method for learning at imbalanced datasets", Proc. Int. Conf. Rough Sets Knowl. Technol., pp. 376-381, 2011.

[2] M. D. Martino, F. Decia, J. Molinelli, A. Fernandez, "Improving electric fraud detection using class imbalance strategies", Proc. Int. Conf. Pattern Recognit. Appl. Methods, pp. 135-141, 2012.

[3] W. Wei, J. Li, L. Cao, Y. Ou, J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data", World Wide Web, vol. 16, no. 4, pp. 449-475, 2013.

[4] G. Xu, F. Shen, J. Zhao, "The effect of methods addressing the class imbalance problem on P300 detection", Proc. Int. Joint Conf. Neural Netw., pp. 1-5, 2013.

[5] A. Amin et al., "Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study", IEEE Access, vol. 4, pp. 7940-7957, Oct. 2016.

[6] H. He, E. A. Garcia, "Learning from Imbalanced Data", IEEE Trans. Knowle. Data Eng., vol. 21, no. 9, pp. 1263-1284, Jun. 2009.

[7] R. Prati, G. A. P. A. Batista, M. Monard, "Class imbalances versus class overlapping: An analysis of a learning system behavior", Proc. Mexican Int. Conf. Artif. Intell. Adv. Artif. Intell., pp. 312-321, 2004.

[8] H. Cao, X.-L. Li, D.-K. Woon, S.-K. Ng, "Integrated oversampling for imbalanced time series classification", IEEE Trans. Knowl. Data Eng., vol. 25, no. 12, pp. 2809-2822, Dec. 2013.

[9] N. Thai-Nghe, Z. Gantner, L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data", Proc. Int. Joint Conf. Neural Netw., pp. 1-8, 2010.

[10] R. Batuwita, V. Palade, "FSVM-CIL: Fuzzy support vector machines for class imbalance learning", IEEE Trans. Fuzzy Syst., vol. 18, no. 3, pp. 558-571, Jun. 2010.

[11] P. Cao, D. Zhao, O. Zaiane, "An optimized cost-sensitive SVM for imbalanced data learning", Proc. Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining, pp. 280-292, 2013.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique", J. Artif. Intell. Res., vol. 16, pp. 321-357, 2002.

[13] H. He, Y. Bai, E. A. Garcia, S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning", Proc. IEEE Int. Joint Conf. Neural Netw., pp. 1322-1328, 2008.

[14] S. Barua, M. M. Islam, Y. Xin, K. Murase, "MWMOTE–Majority weighted minority oversampling technique for imbalanced data set learning", IEEE Trans. Knowl. Data Eng., vol. 26, no. 2, pp. 405-425, Feb. 2014.

[15] L. Abdi, S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques", IEEE Trans. Knowl. Data Eng., vol. 28, no. 1, pp. 238-251, Jan. 2016.

[16] B. Das, N. C. Krishnan, D. J. Cook, "RACOG and wRACOG: Two probabilistic oversampling techniques", IEEE Trans. Knowl. Data Eng., vol. 27, no. 1, pp. 222-234, Jan. 2015.

[17] R. C. Prati, G. E. Batista, M. C. Monard, "A study with class imbalance and random sampling for a decision tree learning system", Proc. Artif. Intell. Theory Practice II, pp. 131-140, 2008.

[18] X. Zhang, D. Ma, L. Gan, S. Jiang, G. Agam, "CGMOS: Certainty guided minority oversampling", 2016.

[19] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, "Handling imbalanced datasets: A review", GESTS Int. Trans. Comput. Sci. Eng., vol. 30, pp. 25-36, 2006.

[20] A. Pourhabib, B. K. Mallick, Y. Ding, "Absent data generating classifier for imbalanced class sizes", J. Mach. Learn. Res., vol. 16, pp. 2695-2724, Jan. 2015.

