EoT: Ensemble of Trees For Classifying High Dimensional Imbalanced Data

Md. Nurul Huda
ID: 012102015

A Thesis Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Science in Computer Science and Engineering

United International University

Dhaka, Bangladesh

February 2018

Approval Certificate

This thesis titled " **EoT : Ensemble of Trees For Classifying High Dimensional Imbalanced Data**" submitted by Md. Nurul Huda, Student ID: 012102015, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on 27th February, 2018.

Board of Examiners

1.

_____ Supervisor

Dr. Dewan Md. Farid
Associate Professor

2.

_____ Head Examiner

Dr. Mohammad Nurul Huda
Professor & Coordinator - MSCSE

3.

_____ Examiner-I

Suman Ahmmed
Asst. Professor, CSE
Director, CDIP & CESR

4.

_____ Examiner-II

Mohammad Moniruzzaman
Assistant Professor

Declaration

This is to certify that the work entitled "**EoT : Ensemble of Trees For Classifying High Dimensional Imbalanced Data** " is the outcome of the research carried out by me under the supervision of Dr. Dewan Md. Farid, Associate Professor, CSE.

_____

Md. Nurul Huda,
ID: 012102015
MSCSE

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

_____

Dr. Dewan Md. Farid
Associate Professor, CSE

Abstract

Class imbalance classification is a challenging research problem in data mining and machine learning, as most of the real-life datasets are often imbalanced in nature. Existing learning algorithms maximize the classification accuracy by correctly classifying the majority class, but misclassify the minority class. However, the minority class instances are representing the concept with more significant interest than the majority class instances in real-life applications.

Recently, several techniques based on sampling methods (under-sampling of the majority class and over-sampling the minority class), cost-sensitive learning methods, and ensemble learning have been used in the literature for classifying imbalanced datasets.

In this research, we introduce a new correlation-based feature grouping with under-sampling ensemble algorithm, called EoT for effective high dimensional imbalanced classification. EoT provides an competitive alternative to Bagging (sampling Bagging) and Random forest (sampling with Random forest) algorithms. We evaluated the performance of our proposed algorithm with the state-of-the-art methods based on ensemble like RUS-Bagging and RUS-Randomforest on 10 high dimensional imbalanced binary-class datasets with various imbalance ratios. The experimental results show that the EoT is a promising and effective approach for dealing with highly imbalanced datasets with large number of features.

Acknowledgement

This work would have not been possible without the help and support of some people over a significant time. I would like to express my gratitude to everyone who contributed to it in some way or other.

I would like to start by expressing my gratitude to Dr. Dewan Md. Farid, my respected academic supervisor. His continuous guidance, friendly attitude and inspiration throughout kept me motivated and optimistic in achieving results. His continuous guidance kept me on the right track and to finish my thesis successfully. My sincere gratitude also goes to United International University (UIU) for providing me with a wonderful environment of academics and fellow researchers. I am thankful to Dr. Swakkhar Shatabda for his useful advice in developing my ideas. Additionally, I thank Prof. Dr. Chowdhury Mofizur Rahman and Prof. Dr. Mohammad Nurul Huda for spending time in giving me a foundation on which to build my Research. I also thank the Soft Computing and Intelligent Information Systems (sci2s) Research Group of the University of Granada for providing us with access to many pioneering research materials.

Last but not the least, I owe to my family, especially my parents, for their immense emotional support which kept me going.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Huge developments in science and technology in modern times have led to the generation of data in all sectors such as the World Wide Web, healthcare, and similar scientific domains. Such large outbursts of data opens the gates for knowledge discovery and data research to take a crucial role in a wide range of applications from everyday activities to industrial decision-making applications. This has helped data mining technology to become one of the fastest growing areas of the information age of the 21st century.

Data mining and machine learning methods are capable of processing structured or unstructured data for identifying useful trends. Machine Learning methods learn through inference, so the data to be mined must be structured as a collection of examples of the target concept to be learned. Each sample, or data point, is described by a set of feature values, which typically are either numeric or categorical values. From this structured data, data mining algorithms can discover useful patterns through either supervised learning or unsupervised learning techniques.

Class imbalance problem is a fairly recent problem that has received attention in data mining applications. In real-world class imbalance data sets such as such as software prediction, oil spill detection, from satellite images, fraudulent transaction detection, diagnosis of rare diseases, the majority class instances dominate in quantity over the minority class instances. [1-3] However, the minority class instances are representing the concept with

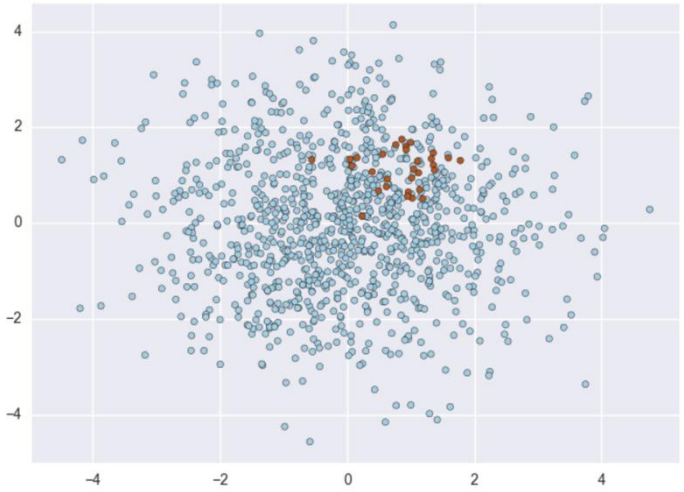

Figure 1: Imbalanced data.

more significant interest than the majority class instances. [4].

The most recommended methods for handling the problem of class imbalance are sampling techniques, ensemble, and cost-sensitive learning methods. The sampling techniques (under-sampling and over-sampling) either remove the majority class instances from the imbalanced dataset or add the minority class instances into the imbalanced dataset to get a comparatively balanced dataset. Ensemble techniques such as Bagging and Boosting are also widely used for the classification of imbalanced datasets. The ensemble methods use sampling technique in each iteration [5]. The cost-sensitive learning is also employed in solving the class imbalance problems, which assigns different costs concerning the misclassification error of classes. Typically high cost is allocated for the minority class instances and vice versa. [6] However, the classification results are not stable in cost-sensitive learning methods since it is tough to get the appropriate misclassification cost and various misclassification costs may result in multiple inductions.

The procedures handling imbalanced classification problems can be bifurcated into two categories: (i) external methods and (ii) internal methods. The external strategies are commonly referred to as balancing methods, which preprocess the imbalanced dataset to get a balanced data. The internal methods modify the existing learning algorithms for reducing their sensitiveness to the class imbalance when learning from the imbalanced data. Before delving into the different algorithms used for handling imbalanced dataset, let us first understand the basics of supervised and unsupervised learning algorithms from which we obtain predicted classifications.

1.1 Machine Learning

Machine learning is a term first coined by scientist Arthur Samuel as a branch of artificial intelligence that gives "computers the ability to learn without being explicitly programmed" [7]. Machine learning deals with the analysis and creation of algorithms that facilitate the making of data-driven decisions from previously provided data. The algorithms use the provided data, also known as training data, to "learn" by building a model and utilize it to generate predictions. This finds applications in domains where programming explicit algorithms is difficult or infeasible, such as email filtering, intrusion

detection and computer vision. Machine learning algorithms can be broadly classified into two categories: supervised learning and unsupervised learning.

1.1.1 Supervised Learning

Supervised learning is the process of identifying new or unknown instances by training estimators (or classification algorithms) using a group of samples with known class values. Each example of the training data consists of a series of attributes in the form of a vector and labeled as belonging to a certain class. This training data is used to facilitate the creation of a predictive model. This, in turn, is used as a generalized definition to classify unknown instances or test data. In other words, supervised algorithms model relationships and dependencies between the input attributes and target prediction output in such a way that it allows for the prediction of output values against new data based on those relationships learned from the previous data. Supervised learning algorithms can be divided into two categories

1. Classification algorithms: Here the model is trained to predict class labels (a discrete value). Examples include:
    a. Decision trees [8]
    b. Random forests [9]
    c. Support vector machines (SVM) [10]
    d. Neural Networks [11]
    e. K-nearest-neighbors (KNN) [12]
    f. Naive Bayes [13]

2. Regression algorithms: Here the model is trained to predict new values for data wherein the data is continuous [14]. Examples include:
    a. Linear regression
    b. Multivariate regression
    c. Regression trees
    d. Lasso regression
    e. Logistic regression

1.1.2 Unsupervised Learning

Unsupervised learning consists of machine learning algorithms which use various techniques on the input data in order to derive meaningful patterns and create groups of data instances which are similar to each other. Here, there are no output labels or predefined categories for the given data to use in order to model the relationship between the features and its respective group. Unsupervised learning techniques can be divided into two subcategories:

1. Clustering algorithms: These algorithms use the features of the given unlabeled data to create groups or clusters of different classes. The variations of the algorithms stem from their different approaches in creating the clusters. The criteria for dividing the data include mean, medoids, hierarchies, and others. [15] Examples of clustering algorithms include:
    a. K-means [15]
    b. K-medoids
    c. Hierarchical clustering
    d. Fuzzy c-means
    e. Gaussian clustering
    f. Density based clustering

2. Association rule learning algorithms: These algorithms use features of the given data to mine for rules and patterns from the datasets that explain relationships between different attributes. These rules are mined based on some measures of interestingness. Examples of association rule learning algorithms include:
    a. Apriori algorithm
    b. FP-Growth algorithm
    c. Eclat algorithm

1.2 Data Mining Task

Data mining is the process of finding hidden information and patterns from the dataset. Alternatively, it has been called exploratory data analysis, data driven discovery and

deductive learning. Mainly in data mining we try to extract patterns or information from a dataset and build a logical structure by transforming the dataset. It is a challenge of data mining to choose the architecture for working with dataset because it varies with the dataset. Most of the data mining work follows a basic architecture as shown in figure 1.

1.3 Class Imbalance Problem & Motivation for Research

As mentioned previously, the class imbalanced datasets occur in many real-world applications. The aforementioned happens when the class distribution is imbalanced. Typically, the minority classes hold the most vital information and misclassifying them can lead to severe problems. As a result, class imbalanced has emerged as a very challenging problem in the field of data mining.

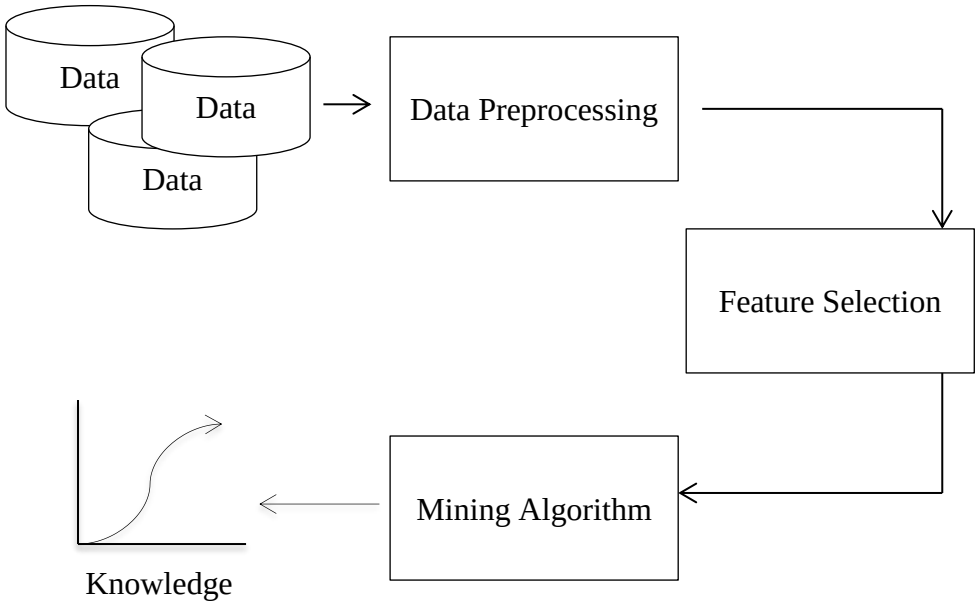

Figure 2: The process of extracting knowledge using Data Mining.

When class imbalance occurs, it usually occurs in a way that 1 in 100 instances is a minority class instance. Therefore, even if a classifier is unable to classify the minority class instances it will still have an accuracy of 99%. For that reason, accuracy of a classifier is not always sufficient to determine the performance of a classifier [16]. Problems arise with traditional classification algorithms in which two crucial assumptions are made [17] :

1. The goal of the classifiers is to maximize the accuracy (or minimize error rate)

2. The class distribution of training and test datasets are the same

Under these two assumptions, predicting everything as the majority class in a highly imbalanced dataset is the best approach to undertake. In other words, the machine learning algorithms become biased towards the majority class since its goal is to maximize accuracy and the majority class exists in high dominance over the minority class. Therefore, when presented with invloved imbalanced data sets, these algorithms fail to properly represent the distributive characteristics of the data and result in unfavorable accuracies across the classes of the data. This can be visually seen in figure 2, taking the example of Support Vector Machine (SVM) classifier [18]. The classifier produces a biased decision boundary since its goal is to minimize the classification error. To achieve this, the line must shift away from the majority class instances, represented by red color, to ensure that it is not misclassified.

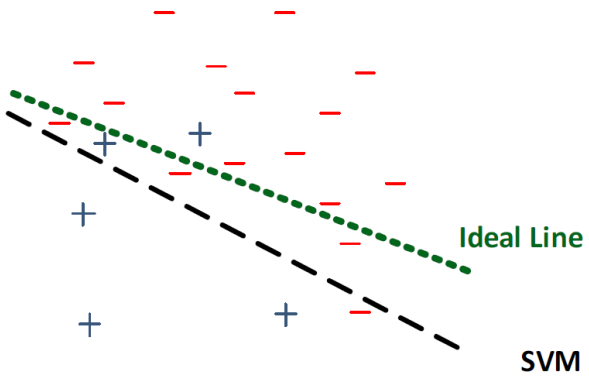

Figure 3: SVM classifier becomes biased towards majority class in imbalanced dataset.

There is also the case of multi-class imbalanced problems. A dataset is considered multiclass if it has k classes, where k is greater than 2. Data imbalance can also exist in a way such that there are multiple ways between classes to become imbalanced.

Such misclassification can have detrimental results in real life applications. Lets consider the case of cancer prediction based on a training dataset. If a person without cancer is diagnosed as having cancer, although this is not desirable it does not lead to any major damage. It may propel further investigation which ultimately will clarify that the person does not really have cancer. However, if a patient having cancer is diagnosed as not having

cancer, then this can have a dangerous impact. The person may not look into treatment thinking that they are not affected by the disease while the condition of the person grows worse without their awareness.

A number of ensemble classifiers with sampling techniques have been proposed for classifying binary-class low-dimensional imbalanced data in the last decade [19-21]. Ensemble classifiers use multiple machine learning algorithms to improve the performance of individual classifiers and combine multiple hypotheses to form an advance hypothesis [5]. The sampling methods use under-sampling (under-sampling of the majority class instances) and over-sampling (over-sampling the minority class instances) techniques to alter the original class distribution of imbalanced data. The under-sampling methods with random sampling of the majority class might suffer from the loss of potentially useful training instances. Furthermore, over-sampling with replacement proved to be ineffective in increasing the minority class recognition as it increases the likelihood of overfitting [20].

In light of this, we have found this domain to be a challenging research area for which we can make good contributions.

1.3 Thesis Contributions

Our thesis achieves the following goals:

- Study the application of different sampling techniques on imbalanced datasets
- Learn various ensemble learning methods for handling class imbalance at both data level and algorithm level
- Address the problems caused by high dimensionality in case of imbalanced datasets and mitigate them as much as possible
- Tackle the escalated information loss caused by feature selection and sampling in case of class imbalance high dimensional datasets
- Propose a new optimal ensemble classifier for classifying high dimensional imbalanced datasets

- Evaluate the proposed method against state-of-the-art techniques currently used

In particular, we present a new correlation-based feature grouping approach combined with under-sampling and Bagging, called the EoT algorithm. We first generate the correlation matrix among features. Then, we cluster the features into several clusters using Equal Size K-means clustering algorithm (with added hard constraint of equal size clusters) based on the aforementioned matrix and select the features from some cluster to form a sub-dataset to be used for model training. Clustering helps us to group the features in such a way that features in the same cluster are more related to each other than to those in other clusters. So, instead of randomly forming feature groups like Random Forest or FSE [53] we used clustering technique to form these groups. EoT combines the sampling and feature grouping methods to form an efficient and effective ensemble algorithm for class imbalance learning.

We appraised the performance of our proposed method with Bagging and Random Forest algorithms on 10 imbalanced datasets. Based on our experiments, we can validate that integrating clustering-based feature grouping with sampling and Bagging algorithm is a promising technique for alleviating the problems of class imbalance and high dimensionality**.**

1.4 Organization of Thesis

The thesis is organized as follows:

Chapter 1 presents our main research problem briefly and explores its importance as well as our motivation to pursue in this track.
Chapter 2 discusses the various related work that have been conducted over the past with regards to class imbalance and the curse of dimensionality. We focus mainly on binary class high dimensional imbalanced problems and their proposed solutions.

Chapter 3 describes our proposed method in detail and compares it with similar bagging based methods.

Chapter 4 describes the datasets we used to evaluate our method as well as the experimental results.

Chapter 5 finishes with the conclusion and directions for future work

Chapter 2

Background and Literature Review

In recent years, the problem of class imbalance has garnered huge interest in the research community. Various methods have been proposed which can often be used as extensions to traditional machine learning approaches, thus making them compatible with the imbalance classification scenario. In the last decade, sampling methods, ensemble methods based on bagging, boosting and cost-sensitive learning methods have been amongst the most popular methods used to handle imbalanced binary classification problems [23]. In this chapter, we discuss the different methods proposed according to their classifications.

2.1 Sampling Methods

2.1.1 Under-sampling

Under-sampling balances the dataset by randomly removing instances from the majority class [22]. Although it has the benefit of decreasing the time required to train the model, this can cause the removal of informative samples from majority class especially if the number of instances of the minority class is very small . As a result, there will be a great loss of information from the majority instances, leading to a non-optimal classification. Figure 3 shows the random under-sampling process to select the majority class instances randomly. There have been many variations of under-sampling techniques proposed over the years by researchers and such methods are investigated in the following list.

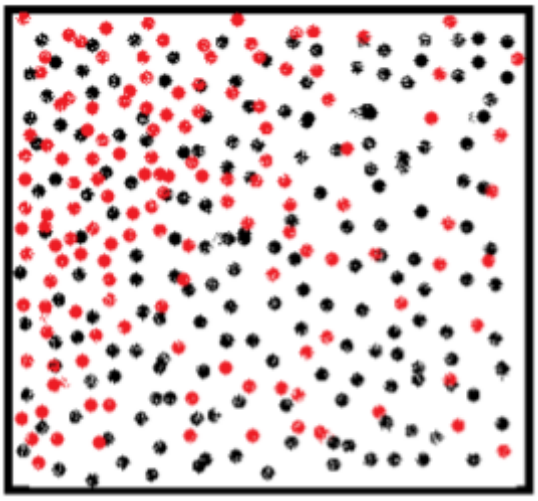

Figure 4: Random under-sampling (RUS) approach. Here the red dots are selected instances from the majority class, where black and red dots are representing all the majority class instances.

1. Neighborhood cleaning rule (NCL) proposed in [24] is a different form of under sampling, which considers the quality of the data before removing it. This method is very effective in building a model in such a way that it become easier for the classifier to identify the instances of minority class. It has been demonstrated through experiments that NCL helps to recuperate the representation of minority class instances and facilitate classifiers in identifying their presence in class imbalanced domains.

2. Cluster-based under-sampling approaches were studied by Show-jane Yen and Yue-Shi Lee [25]. This approach first groups all instances of the dataset into K clusters. In each i^{th} cluster, the number of majority class instances to be selected is found based on the imbalance ratio of the respective cluster and such number of instances is chosen randomly as representatives of the majority class. These instances are combined with minority class instances to obtain the training dataset. They also proposed five other under-sampling approaches for the majority class instances based on Euclidian distance between instances in the cluster. The other proposed methods select majority class instances having the lowest or highest average distance to the M (defined or all instances) nearest or furthest minority class instances. Experimental results showed that the first approach (SBC) performs better than the traditional random under-sampling method as it takes the most representative instances across the whole data space into account for generating the training data. Further insight of the experiment results showed that under-sampling based on instances having the highest average distance from M closest minority class instances (SBCMD) showed stable performance on datasets having high anomalies while the other proposed under-sampling techniques based on clustering did not show stable performance.

3. When under sampling is done on a dataset, a lot of important instances are likely to get ignored. Two algorithms were proposed in [26] to deal with this deficiency. Easy-Ensemble combines ensemble (will be discussed in Section 4) method with under sampling, where several subsets are created using the majority class by under-sampling, which are all separately trained. The results are later combined to get a single output. Balance-Cascade uses AdaBoost as it's base classifier, which is an

ensemble method. It trains the data using the ensemble method but the difference is that, in each iteration the majority class examples that are correctly classified are completely discarded from further consideration.

2.1.2 Oversampling

Oversampling balances the data by creating a super set by adding instances to minority class. No information is lost during oversampling. However, it increases the runtime to classify the data because oversampling increases the size of the dataset in order to balance it. Oversampling increases the number of minority class instances, and this can be done in different ways. The most familiar of these procedures is random oversampling. In the method above, minority samples are replicated till samples of both concepts are balanced. The difficulty with this procedure, however, is that there is a high probability of overfitting due to the same instances occurring multiple times. Overfitting occurs when the classifier models the training data too well and cannot generalize to classify new instances beyond those with features as found in the training data.

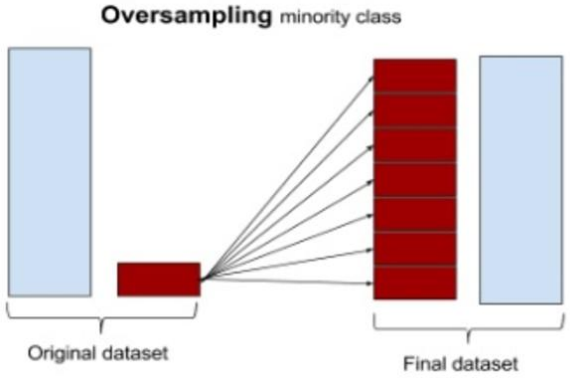

Figure 5: Oversampling technique.

1. SMOTE (Synthetic Minority Over-sampling Technique) approach proposed in [27] has been able to achieve better classifier performance than random over-sampling. The main idea of SMOTE is to learn using minority classes by operating in the "feature space" instead of the "data space". SMOTE combines the under-sampling of the majority class instances with over-sampling of minority class instances through the creation of synthetic data instances. It creates new instances not quite randomly because it takes k closes instances into account before

generating another instance. This is done by by interpolating several minority class instances that lie together. SMOTE is illustrated in Figure 4. Here, x_i is the selected point, x_{i1} to x_{i4} are some selected nearest neighbors and r_1 to r_4 the synthetic data points created by the randomized interpolation. It has been successful in reducing loss ratio caused by under-sampling. However, overfitting can be present up to some degree because the newly created instances could be similar to the existing minority samples.

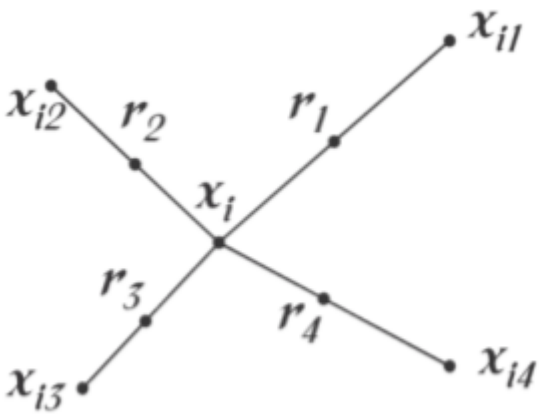

Figure 6: An illustration of how to create synthetic data points from the SMOTE algorithm.

2. Borderline-SMOTE was proposed by Huin Han and Wen-Yuan in [28]. It is a variation of SMOTE oversampling. However it only considers the borderline instances to create a new instance. It has a better True Positive rate and F-value compared to SMOTE.

3. SMOTE-ENN [29] is yet another variation of SMOTE where an instance is selected. It is classified by its closest examples. If it is not correctly classified by 3 of them, it is discarded from the training set.

4. Safe-level-SMOTE introduces a concept called safe level which is computed using nearest neighbor minority instances. It synthesizes minority instances around safe level. Experimental results from paper [30] prove that this method has a better accuracy than SMOTE and Borderline-SMOTE.

5. SMOTE-RSB [31] uses SMOTE together with editing technique based on Rough Set Theory (RST). Initially SMOTE is applied to balance the dataset then rough set theory is applied. The latter gets rid of any inconsistency by joining together instances that resembles each other.

6. Santos et al. [46] implemented a cluster-based (k-means) over-sampling approach where SMOTE was adapted to oversample clusters with reduced sizes. This work considered merging the minority class instances from the multiple over-sampled datasets.

7. ADASYN algorithm from paper [32] was motivated by the success of SMOTE and it improves learning in two ways:
    a. It decreases the bias created by the class imbalance and
    b. It adaptively adjusting the boundary of classification towards the tough examples.

    It which creates synthetic minority instances using k nearest neighbors [47]. However, in contrast to SMOTE, this method assigns weight to the minority class instances based on how many neighbors are of the majority class. This weight is used to determine how many synthetic instances to generate from which minority instance.

2.2 Cost Sensitive Measures

Cost sensitive learning method does not focus on balancing the data by sampling or creating synthetic instances; rather it is works by weighting the existing instances using a cost matrix. The weight of each misclassified instances is increased and the weight of each correctly classified instances is decreased. This is a weak method because the weight of an instance cannot be determined before the entire dataset has been trained.

2.2.1 Cost-sensitive Tree

Cost-sensitive Tree induction via instance weighting was proposed in paper [33] written by Kai Ming Ting. The idea of information gain is used to construct a tree. The attribute with highest entropy is assigned as the root, the one with lower gain becomes the child of the node and so on. After the tree is constructed, pruning is carried out to reduce the size of the tree. This results in a tree with least error and therefore this method has a higher accuracy.

2.2.2 Cost-sensitive Neural Network

Zhi-Hua Zhou and Xu-Ying Liu came up with another approach [34] that involves neural network. Sampling, threshold moving, hard ensemble and soft ensemble were all considered while training the Cost Sensitive Neural Network. Overall, threshold-moving and soft ensembles were figured out to be a good choice for training cost sensitive neural networks. However, this method became more difficult with increase in the degree of class imbalance.

2.2.3 GSVM-RU

SVM models are usually more accurate on moderately imbalanced data but they are not always very efficient. Granular SVMs-repetitive undersampling algorithm (GSVM-RU) suggested in [35] uses the idea of SVM to minimize the negative effect of information loss and maximize the positive effect of data cleaning in the under-sampling process. It has also been demonstrated to be is very efficient because it modifies the model in such a way that is becomes easier for SVM to classify the instances and it also requires less time to compute.

2.3 Ensemble method

Ensemble method is the idea of combining several traditional classifiers to build a model, which has a much higher accuracy. **Error! Reference source not found.** shows how a generic ensemble learning model uses the dataset to generate a prediction.

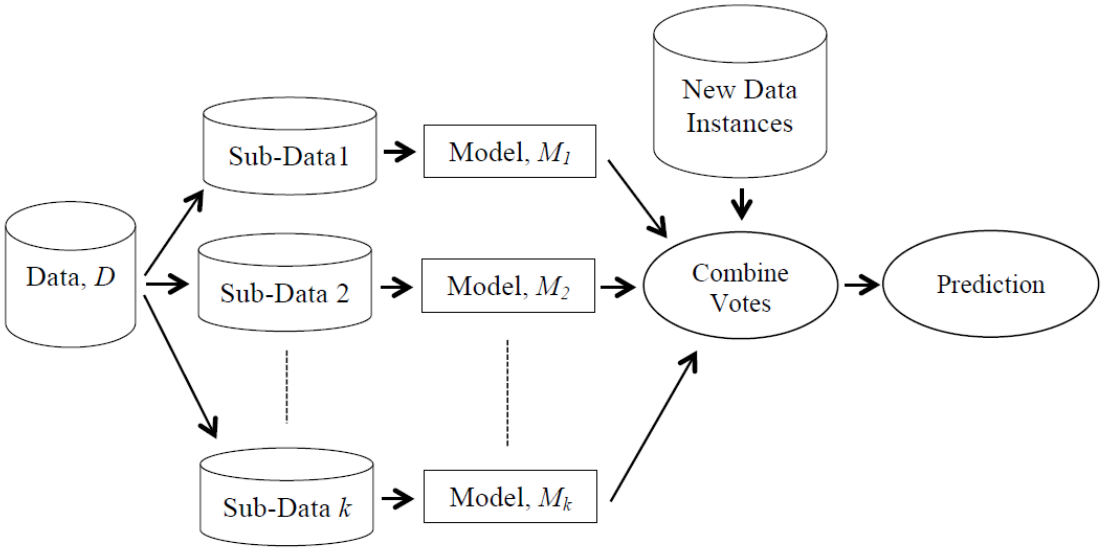

Figure 7: Ensemble Learning Model.

2.3.1 Bagging

Bagging [36] creates subsets from the original dataset using sampling with replacement method. It then derives a model for each subset using a learning scheme (which can be any traditional classifier). Any new instance is classified using majority voting from the derived models. All the models return a class label and the class label that receives the most votes is assigned to the instance. The bagged classifier shows better predictive accuracy compared to the individual base classifiers in the case of balanced classification tasks. The standard bagging algorithm is shown below:

Algorithm 1 Bagging Algorithm

Input: Training data, D, number of iterations, k, and a learning scheme.
Output: Ensemble model, M^*
Method:
1: **for** i = 1 to k **do**
2: create bootstrap sample D_i, by sampling D with replacement;
3: use D_i, and learning scheme to derive a model, M_i;
4: **end for**
To use M^* to classify a new instance, x_{New}:
Each $M_i \in M^*$ classify x_{New} and return the majority vote;

1. Shuo Wang and Xin Yao carried out a diversity analysis on imbalanced dataset in their paper [37]. They combined under sampling, over sampling and SMOTE into

their ensemble models. Their study lead to the proposal of three hybrid bagging based methods:

a. Under-Over-Bagging
b. Over-Bagging and
c. SMOTE-Bagging

Over-Bagging is quite similar to Under-Bagging in that Random over-sampling is done instead of Random under-sampling in each iteration. Under-Over-Bagging trains the base classifiers with varying resampling rates (multiples of 10) which determine the number of instances coming from each of the classes, thus using Random under-sampling and Random over-sampling according to need. SMOTE-Bagging uses SMOTE as a pre-processing step for the base classifiers. These hybrid bagging based ensembles have shown improved predictive accuracy in imbalanced learning domains.

2. Lu et al. proposed HS-Bagging [48] which uses both Random under-sampling and SMOTE at every bagging iteration as a pre-processing step. This method chooses an optimal sampling rate (majority: minority ratio) for SMOTE and Random under-sampling based on predictive accuracy on OUT-OF-BAG instances. This method showed better performance compared to Under-Bagging and SMOTE-Bagging. However in this method, time complexity has been increased due to sampling rate optimization.

3. Barandela et al. proposed Under-Bagging [49] which uses Random under-sampling as a pre-processing step for each of the base classifiers in bagging.

2.3.2 Boosting

Boosting is an ensemble method of generating strong hypothesis through the combination of multiple weak classifiers. Boosting is similar to bagging in that it combines multiple base learners to obtain a result based on voting technique. However, it's differences lie in that boosting assigns weight to instances according to how hard they are to classify, thus

assigning a higher weight to samples that are harder to recognize. A base model uses the weights appointed by the previous base models. The base models are also assigned weights according to their predictive accuracy, and these weights are multiplied with the predictions of the respective learners when majority voting is done. Figure 8: Sampling with boosting for classifying imbalanced data shows the process of classifying imbalanced data using sampling with boosting approach.

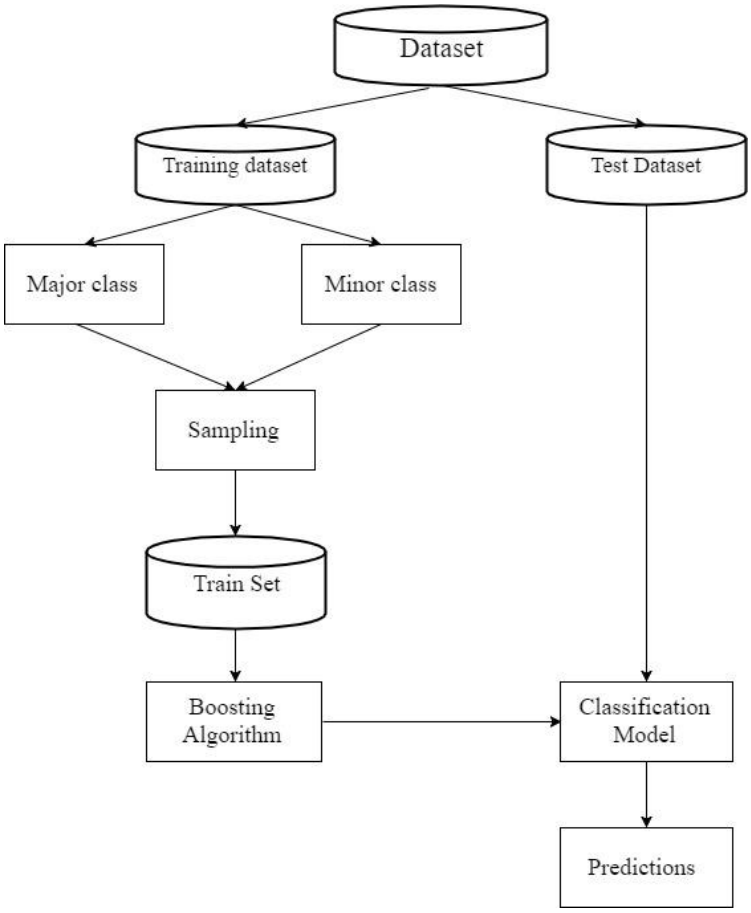

Figure 8: Sampling with boosting for classifying imbalanced data.

Although boosting was not designed explicitly for imbalanced classification, the way it assigns weights to difficult samples makes it consummate for the purpose. It can be regarded as an implicit cost-sensitive procedure which deals with the problem of class imbalance by assigning more weight to the minority instances since they will be harder to classify in an imbalanced dataset.

1. AdaBoost [38] is an accuracy-oriented algorithm where all the instances are initialized with the same weight. In each iteration, a model is created by sampling instances with replacement. The instances with higher weight are taken into account. The model is classified and the weight of the misclassified instances is increased. A series of iterations are carried out and eventually the most misclassified instances will have the highest weight at the end. The ensemble models are combined together to output a single class label. Majority voting is used in order to determine the class label for a new instance. These hybrid approaches

have a higher accuracy when compared to a single classifier. Moreover, the base classifier does not even need to be changed.

Algorithm 2 AdaBoost

Input: Training set $S = \{\mathbf{x}_i, y_i\}$, $i = 1, \ldots, N$; and $y_i \in \{-1, +1\}$; T: Number of iterations; I: Weak learner

Output: Boosted classifier: $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$ where h_t, α_t are the induced classifiers (with $h_t(x) \in \{-1, 1\}$) and their assigned weights, respectively

$D_1(i) \leftarrow 1/N$ for $i = 1, \ldots, N$
for $t = 1$ to T **do**
  $h_t \leftarrow I(S, D_t)$
  $\varepsilon_t \leftarrow \sum_{i, y_i \neq h_t(\mathbf{x}_i)} D_t(i)$
  if $\varepsilon_t > 0.5$ **then**
    $T \leftarrow t - 1$
    return
  end if
  $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
  $D_{t+1}(i) = D_t(i) \cdot e^{(-\alpha_t h_t(\mathbf{x}_i) y_i)}$ for $i = 1, \ldots, N$
  Normalize D_{t+1} to be a proper distribution
end for

2. Two new extensions of AdaBoost,AdaBoostM1 and AdaBoostM2 were proposed in [39] [40]. These two algorithms mainly concentrate on improving the dataset in such a way that it is easier to classify the minority class examples therefore reducing training error.

Algorithm 3 AdaBoost.M1

Input: Training set $S = \{\mathbf{x}_i, y_i\}$, $i = 1, \ldots, N$; and $y_i \in \mathbb{C}, \mathbb{C} = \{c_1, \ldots, c_m\}$; T: Number of iterations; I: Weak learner

Output: Boosted classifier:

$$H(x) = \arg\max_{y \in \mathbb{C}} \sum_{t=1}^{T} \ln\left(\frac{1}{\beta_t}\right) [h_t(x) = y]$$ where h_t, β_t are the induced classifiers (with $h_t(x) \in \mathbb{C}$) and their assigned weights, respectively

$D_1(i) \leftarrow 1/N$ for $i = 1, \ldots, N$
for $t = 1$ to T **do**
  $h_t \leftarrow I(S, D_t)$
  $\varepsilon_t \leftarrow \sum_{i=1}^{N} D_t(i)[h_t(\mathbf{x}_i) \neq y_i]$
  if $\varepsilon_t > 0.5$ **then**
    $T \leftarrow t - 1$
    return
  end if
  $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
  $D_{t+1}(i) = D_t(i) \cdot \beta_t^{1-[h_t(\mathbf{x}_i) \neq y_i]}$ for $i = 1, \ldots, N$
  Normalize D_{t+1} to be a proper distribution
end for

Algorithm 4 AdaBoost.M2

Input: Training set $S = \{\mathbf{x}_i, y_i\}$, $i = 1, \ldots, N$; and $y_i \in \mathbb{C}, \mathbb{C} = \{c_1, \ldots, c_m\}$; T: Number of iterations; I: Weak learner

Output: Boosted classifier:

$$H(x) = \arg\max_{y \in \mathbb{C}} \sum_{t=1}^{T} \ln\left(\frac{1}{\beta_t}\right) h_t(x, y)$$

where h_t, β_t (with $h_t(x, y) \in [0, 1]$) are the classifiers and their assigned weights, respectively

$D_1(i) \leftarrow 1/N$ for $i = 1, \ldots, N$

$w_{i,y}^1 \leftarrow D(i)/(m-1)$ for $i = 1, \ldots, N,\ y \in \mathbb{C} - \{y_i\}$

for $t = 1$ to T **do**

$W_i^t \leftarrow \sum_{y \neq y_i} w_{i,y}^t$

$q_t(i, y) \leftarrow \frac{w_{i,y}^t}{W_i^t}$ for $y \neq y_i$

$D_t(i) \leftarrow \frac{W_i^t}{\sum_{i=i}^{N} W_i^t}$

$h_t \leftarrow I(S, D_t)$

$$\epsilon_t \leftarrow \frac{1}{2} \sum_{i=1}^{N} D_t(i) \left(1 - h_t(\mathbf{x}_i, y_i) + \sum_{i, y \neq y_i} q_t(i, y) h_t(\mathbf{x}_i, y)\right)$$

$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$

$$w_{i,y}^{t+1} = w_{i,y}^t \cdot \beta_t^{\frac{1}{2}(1 + h_t(\mathbf{x}_i, y_i) - h_t(\mathbf{x}_i, y))}$$

for $i = 1, \ldots, N,\ y \in \mathbb{C} - \{y_i\}$

end for

3. SMOTE-Boost [41] adds the idea of SMOTE into the boosting algorithm. Like the other boosting algorithms, this method also assigns weight to the instances and with every iteration the weights are updated the only difference is that in SMOTE-Boost, SMOTE is applied at the end of every iteration to generate synthetic examples for minority class.

4. Data-Boost-IM [42] combines boosting, an ensemble based learning algorithm together with data generation to ensure that the resultant predictive accuracies for both majority and minority class. It focuses on hard example of both the classes to generate synthetic examples.

5. MSMOTE is another algorithm, which divides the minority class examples into 3 distinct categories (safe sample, border sample and latent noise sample) and pays special attention to the groups while generating synthetic instances. Safe samples are created in the exact same way as SMOTE. Border sample also works the same way as SMOTE but instead of considering k neighbors, only the nearest neighbor is considered. Lastly, nothing is considered while generating instances for latent sample. The combination of MSMOTE and AdaBoost was suggested in [43].

6. Seiffert et al. [44] presented a novel hybrid sampling/boosting algorithm, called RUSBoost, which applied random under-sampling (RUS) with AdaBoost algorithm. RUS randomly removes the majority class instances to form a balanced data. RUSBoost was built based on the SMOTE-Boost (synthetic minority over-sampling with AdaBoost) algorithm [41]. SMOTE-Boost was built upon over-sampling approach with AdaBoost algorithm. It is much simpler and much less time consuming compared to SMOTE.

7. Galar et al. [45] presented an ensemble algorithm by evolutionary under-sampling (EUA) approach, called EUSBoost, to classify highly imbalanced datasets. EUS generated several sub-datasets using randomly under-sampling technique and obtained a best under-sampled dataset of the original dataset that cannot be improved further. RUSBoost, SMOTE-Boost, and EUSBoost applied data sampling techniques into the AdaBoost.M2 algorithm by considering the minority class instances and the majority class instances.

Chapter 3

Proposed Method

From our evaluation of different methods, we have explored ensemble approaches in detail, particularly that of Bagging and Random Forest. From our testing, we propose a new algorithm named EoT.

EoT is based on the combination of random under sampling, feature grouping and Bagging algorithm. It is similar to Bagging and Randomforest with the critical difference occurring in the feature bagging technique. Bagging uses the full feature-set for training the base models, while Randomforest elects from random feature subsets at each split. In comparison, our proposed EoT uses cluster-based feature grouping performed in the correlation space and trains each of the base models using one of these groups. EoT separates the majority and the minority class instances from the original dataset and performs under-sampling on the majority class instances for alleviating class-imbalance and divides the features into k equal size clusters using equal size K-means algorithm. Equal Size K-means algorithm is described in the pseudo-code of figure 11 [52]. Here, the parameter k is determined by hyper-parameter tuning. After that, one of the groups is selected randomly at the time of training each of the base models. The intuition behind this procedure is to build each tree using features that are related to each other and work in cooperation. We have tried to preserve this cooperation by grouping features based on the correlation matrix.

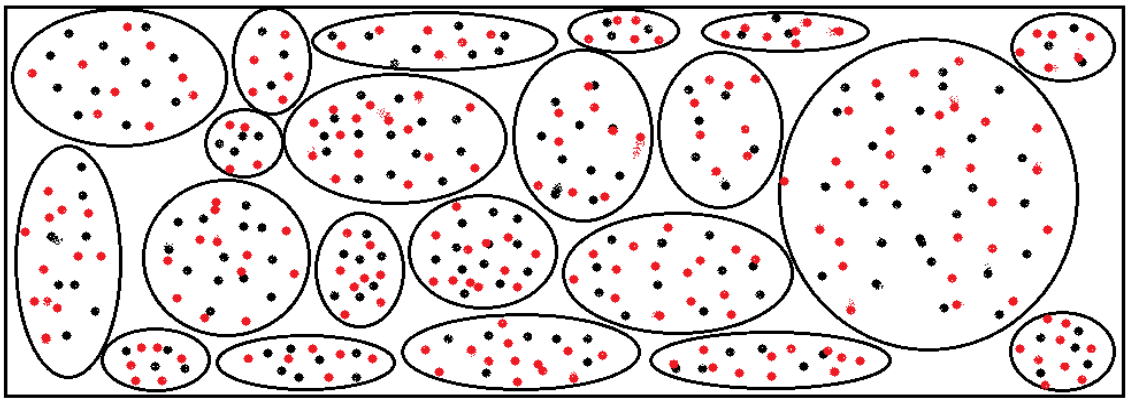

Figure 9: Feature clustering using equal size K-means.

However, features in the groups formulated by our proposed grouping method may be correlated with each other and they will represent the same concept in such a case.
We have taken into account such possibilities and introduced a probability value which is used inside the splitter method of each of the decision trees. The algorithm for the proposed EoT procedure can be found in below.

Algorithm 5: Proposed EoT procedure

Input: Training_data D, number_of_iterations m and number_of_groups k,

Output: Ensemble Model M*

Method:

1. remove all features with variance not greater than zero
2. find the correlation matrix for features of D
3. divide features into k groups employing equal_size_kmeans on correlation space
4. **for** i = 1 to number_of_estimators **do**
    a. pick features from a Cluster;
    b. balance the dataset and generate D_i ;
    c. train base model M_i using D_i;
5. **end for**

To use M* to classify a new instance X_{new} :

classify X_{new} using each $M_i \in M^*$ and return the majority vote ;

Algorithm 6: Equal Size K-means

while termination_flag no false do:

1. estimate centroids for each cluster
2. **for** every instance do :
    a. estimate the distances to the cluster centroids
3. **end for**
4. short instances upon improvement of the best possible substitute cluster over the current cluster
5. **for** each instance extracted from a max heap :
    a. **for** every cluster :
        i. **if** an instance is waiting to leave the cluster and this swap generates improvement, swap instances
        ii. **if** the instance is swapped without breaching size limit, swap instance
    b. **if** no moved :
        i. add instance to list_for_transfer
6. **end for**
7. **if** transfer == 0 or iteration == max_iter :
    a. termination_flag = **true**
8. **end if**

end while

EoT combines each vote of an ensemble of decision trees using C4.5 algorithm [8] to classify new instances. All instances are loaded with identical weights and the weights remain so till the end of the training procedure (EoT is not cost sensitive for dealing with class imbalance). However, the effect of class imbalance is mitigated through under sampling inside each iteration of the method.

3.2 Hypothetical Comparison with Bagging

Advantages of EoT over Bagging :

- Does not consider the full feature set for each split like Bagging which ensures diversity of the base models which is essential for satisfactory performance of ensembles [51].
- Aforementioned diversity enables our proposed method to reduce the variance of the final bagged model.

Disadvantages of EoT compared to Bagging :

- If only a few features contain most of the information, using those features can be decisive in case of the accuracy of the base models which is

instrumental to the success of ensembles [51] and those features may not be selected for each base model in EoT.

3.3 Comparison with RandomForest

Advantages of EoT over Random Forest:

- Random Forest selects feature groups randomly for each split among which the best feature is chosen. However, in cases where the number of features is extremely large, the selected feature may not have the level of association required for the accuracy of the base models which will ultimately effect the accuracy of the final model [51].
- Proposed EoT does not suffer from the aforementioned issue due to the way it ensures that each tree is trained with closely associated features even in case of extremely high dimensionality.

Disadvantage of EoT compared to Random Forest:

- EoT may, not achieve the amount of diversity that Random Forest pulls off due to the way it selects splitting features from the full dataset.

Chapter 4

Results

4.1 Performance Evaluation

To understand the different evaluation metrics we use and how they are derived, first we must understand the confusion matrix.

4.1.1 Confusion Matrix

Confusion Matrix is a two by two matrix containing four results produced by a classifier in case of a two-class problem. In the case of such a problem, an estimator predicts two class values or labels, such as Yes/No and 1/0, for given input data. The comparatively more important class is generally denoted as “positive” and the other as “negative”. These four outcomes are represented as the following:

- True Positive (TP) – This means that the predicted class is positive and in actuality, the class is indeed positive.
- True Negative (TN) - This means that the predicted class is negative and in actuality, the class is indeed negative.
- False Positive (FP) - This means that the predicted class is positive and in actuality, the class is really negative. This is an error made by the classifier is referred to as Type I error.
- False Negative (FN) - This means that the predicted class is negative and in actuality, the class is really positive. This is an error made by the classifier is referred to as Type II error.

Using these preliminary measures, we can now find accuracy and error rate, two of the most commonly used evaluation metrics to assess the performance of a classifier.

Table 1: Confusion Matrix

		Predicted	
Actual		**Positive**	**Negative**
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Accuracy is calculated as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1.0, whereas the worst is 0.0.

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

Similarly, Error rate is calculated as the number of all incorrect predictions divided by the total number of the dataset. The best error rate is 0.0, whereas the worst is 1.0. It can also be found by 1 – Accuracy.

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN}$$

4.1.2 Problem of accuracy as performance metric in class imbalance

Many researchers have noted [16] that accuracy is not a proper measure for evaluating classifiers on imbalanced datasets. When class imbalance occurs, it usually occurs in a way that 1 in 100 instances is a minority class instance. Therefore, even if a classifier is unable to classify the minority class instances it will still have an accuracy of 99%. For that reason, accuracy of a classifier is not always sufficient to determine the performance of a classifier.

Let us take the example of intrusion detection in computer networks. For malicious attacks in networking, let's suppose that:

yes = malicious connection

no = normal connection

Here, FP represents that it was predicted that the connection is malicious but after using practical evaluation, the connection was found to be normal. Similarly, FN represents that

it was predicted that the connection is normal but later found that the connection was malicious. Hence we can clearly understand that the second type of error (FN) is more costly than the first (FN). Detecting a normal connection as malicious did not pose a serious threat to security as proper defensive measures were taken. However, not detecting an actual threat would compromise the entire security of the system. This shows us that Type II error is more costly than Type I error.

4.1.3 Evaluation Metrics

1. True positive rate: It is the percentage of positive examples correctly classified within positive class, also referred to as recall or sensitivity.

$$TPR = \frac{TP}{TP + FN}$$

2. True negative rate: It is the percentage of negative examples correctly classified within negative class, also referred to as specificity.

$$TNR = \frac{TN}{FP + TN}$$

3. False positive rate: It is the percentage of negative examples misclassified as belonging to the positive class.

$$FPR = \frac{FP}{FP + TN}$$

4. False negative rate: It is the percentage of positive examples misclassified as belonging to the negative class.

$$FNR = \frac{FN}{TP + FN}$$

5. Precision: It is the proportion of positive examples that are actually positive, representing how accurate the learning model is.

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

6. F-Measure: As another one class metric, F-measure is defined as:

$$F - Measure = \frac{(1 + \beta^2) * recall * precision}{\beta^2 * recall * precision}$$

where β corresponds to relative importance of precision and recall, and it is usually set to 1. Value of F-measure (or F-value) incorporates both precision and recall, in order to measure the "goodness" of a learning algorithm for the class.

7. G-mean: G-mean is an overall performance metric and defined as:

$$G - mean = \sqrt{positiveAcc * negativeAcc}$$

where $positiveAcc$ and $negativeAcc$ are TP rate and TN rate respectively. This measure relates to a point on the ROC curve and the idea is to maximize the accuracy on each of the two classes in order to balance both classes at the same time.

8. ROC Curve: ROC curve is a two-dimensional graph to select possibly optimal models based on the TP rate and FP rate. It also represents trade-offs between benefits (TP) and costs (FP). In the ROC curve, the TP rate is represented on the Y-axis and the FP rate on the X-axis. Each prediction result or one instance of a confusion matrix represents one point in the ROC space. Several points on a ROC graph should be noted. The lower left point (0, 0) represents that the classifier labeled all examples as negative. The upper right point (1, 1) is the case where all examples are classified as positive. The point (0, 1) represents perfect classification, and the line y = x defines the strategy of randomly guessing the class. In order for assessing the overall performance of a classifier, one can measure the fraction of the total area that falls under the ROC curve (AUC). AUC varies between 0 and +1. Larger AUC values indicate generally better classifier performance.

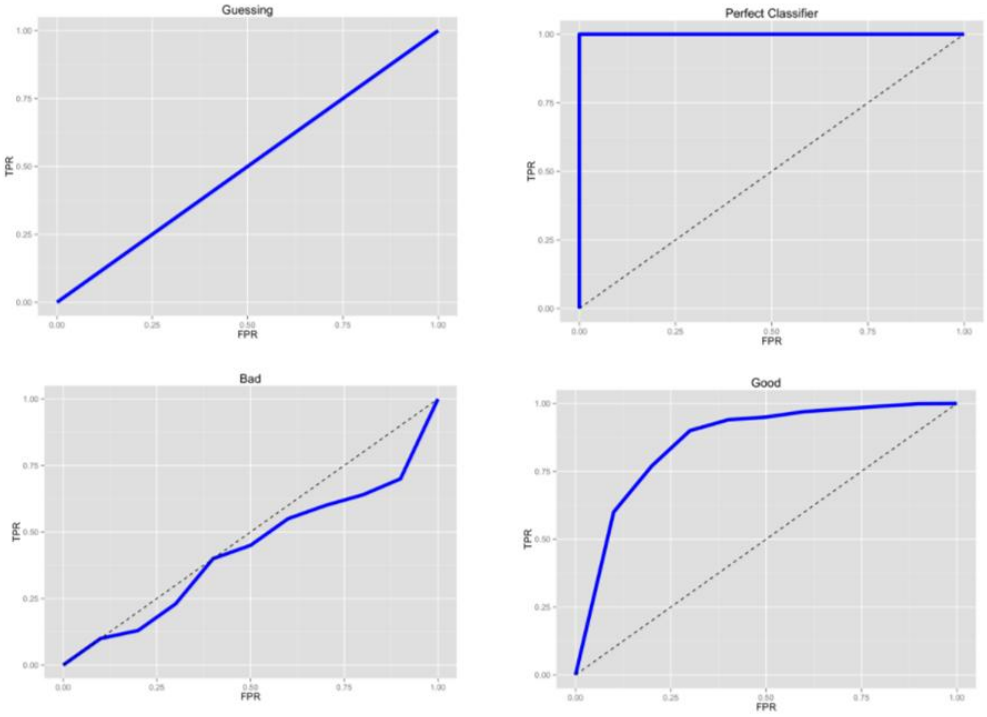

Figure 10: ROC Curve performance comparisons.

4.2 Dataset Characteristics

We have used datasets from KEEL-dataset repository [50] with varying imbalance ratio. Table 2: Description of Datasets Used shows the datasets details.

Table 2: Description of Datasets Used.

Dataset	**Attributes**	**Instances**	**Imbalance Ratio**
segment0	19	2308	6.02
vehicle1	18	846	2.9
cleveland-0_vs_4	13	177	12.62
dermatology-6	34	358	16.9
vowel0	13	988	9.98

page-blocks0	10	5472	8.7
winequality-red-4	11	1599	29.17
poker-9_vs_7	10	244	29.5
poker-8-9_vs_6	10	1485	58.4
page-blocks-1-3_vs_4	10	472	15.86

4.3 Experimental Results

In our experiments, we have used 10 high dimensional imbalanced datasets from KEEL Dataset and have compared the proposed EoT method with Bagging and RandomForest methods. Each dataset was validated using Area Under the ROC Curve (AUC) and AUPR(area under precision recall). As for the base learner, we used C4.5 decision tree induction.

Each of these experiments have been run 10 times with 10 fold cross validation and their average AUROC scores are shown in Table 3 while average AUPR scores are shown in Table 4.

Table 3: Average AUROC Comparison.

Dataset	**Bagging**	**Random Forest**	**Proposed Method**
segment0	0.964	0.979	**0.986**
vehicle1	0.990	**0.995**	0.993
cleveland-0_vs_4	0.900	**0.968**	0.926
dermatology-6	0.998	**1.0**	0.999
vowel0	0.985	**0.988**	**0.988**

page-blocks0	0.986	**0.990**	0.989
winequality-red-4	0.762	**0.795**	0.739
poker-9_vs_7	0.881	0.958	**0.990**
poker-8-9_vs_6	0.719	0.965	**0.999**
page-blocks-1-3_vs_4	0.998	**0.999**	**0.999**

From the results we can see that the EoT algorithm performs best both with respect to AUROC and AUPR most of the times when the imbalance ratio is high and the number of features is large.

Table 4: Average AUPR Comparison

Dataset	**Bagging**	**Random Forest**	**Proposed Method**
segment0	0.988	**0.994**	0.987
vehicle1	0.966	**0.982**	0.977
cleveland-0_vs_4	0.548	**0.784**	0.640
dermatology-6	0.96	**1.0**	0.999
vowel0	0.957	**0.994**	0.989
page-blocks0	0.905	**0.918**	0.916
winequality-red-4	0.142	**0.157**	0.137

poker-9_vs_7	0.458	0.839	**0.893**
poker-8-9_vs_6	0.183	0.631	**0.983**
page-blocks-1-3_vs_4	0.978	**0.988**	**0.988**

Chapter 5

Conclusion

Existing classification algorithms generally focus on majority class instances and ignore the minority class instances in case of datasets with class imbalance while a large number of features make it even more difficult for the classifier to extract useful patterns without over-fitting on the majority class. So, it is a challenging task to construct an effective classifier that can correctly classify the instances a high dimensional imbalanced dataset. This is ever so pertinent as class imbalance problems affect a vast range of domains. Recently, computational intelligence researchers have proposed several hybrid techniques by combining sampling with ensemble classifiers along with several feature selection and feature grouping methods for dealing with high dimensional class imbalance problems. This paper sought to present a new algorithm called EoT, or Ensemble of Trees, in order to deal with high dimensional class imbalanced datasets. We have compared the performance of EoT algorithm with that of Bagging and Random Forest. From our experimental results, we could come to the conclusion that that EoT is competitive in case of high dimensional imbalanced datasets.

However, our current set of experiments is limited to decision trees as base models only. In the future, we intend on experimenting with other base estimators, which are not negatively affected by multi-collinearity. We have used a single value of splitting probability for all the base models currently. In future, we would like to experiment with different probabilities for different base models and test for their synergy.

References

[1] N. V. Chawla and N. Japkowicz, “Editorial: Special Issue on Learning from Imbalanced Data Sets," SIGKDD Explorations, 2004.

[2] G. M. Weiss, \Mining with rarity: a unifying framework," SIGKDD Explor. Newsl., pp. 7-19, 2004.

[3] N. Japkowicz and R. Holte, Workshop report: AAAI-2000 workshop on learning from imbalanced data sets," AI Magazine, vol. 22, pp. 127-136, 2001.

[4] D. M. Farid, A. Now´e, and B. Manderick, “A new data balancing method for classifying multi-class imbalanced genomic data,” 25th Belgian-Dutch Conference on Machine Learning (Benelearn), pp. 1–2, 12-13 September 2016.

[5] D. M. Farid, L. Zhang, A. Hossain, C. M. Rahman, R. Strachan, G. Sexton, and K. Dahal, “An adaptive ensemble classifier for mining concept drifting data streams,” Expert Systems with Applications, vol. 40, no. 15, pp. 5895–5906, November 2013.

[6] Ling, C. X., and V. S. Sheng. "Cost-Sensitive Learning and the Class Imbalance Problem. 2011." Encyclopedia of Machine Learning: Springer.

[7] Munoz, Andres. "Machine Learning and Optimization." URL: https://www. cims. nyu. edu/~ munoz/files/ml_optimization. pdf [accessed 2016-03-02][WebCite Cache ID 6fiLfZvnG] (2014).

[8] Quinlan, J. Ross. C4. 5: programs for machine learning. Elsevier, 2014.

[9] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.

[10] Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." Neural processing letters 9.3 (1999): 293-300.

[11] Haykin, Simon, and Neural Network. "A comprehensive foundation." Neural Networks 2.2004 (2004): 41.

[12] Peterson, Leif E. "K-nearest neighbor." Scholarpedia 4.2 (2009): 1883.

[13] D. M. Farid, L. Zhang, C. M. Rahman, M. Hossain, and R. Strachan, “Hybrid decision tree and na¨ıve bayes classifiers for multi-class classification tasks,” Expert Systems with Applications, vol. 41, no. 4, pp.1937–1946, March 2014.

[14] Meer, Peter, et al. "Robust regression methods for computer vision: A review." International journal of computer vision 6.1 (1991): 59-70.

[15] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108.

[16] R. Barandela, J. S. Sanchez, V. Garcia, and E. Rangel. Strategies for learning in class imbalance problems. Pattern Recognition, 36(3):849-851, 2002.

[17] Provost, Foster. "Machine learning from imbalanced data sets 101." Proceedings of the AAAI'2000 workshop on imbalanced data sets. 2000.

[18] Phoungphol, Piyaphol. "A classification framework for imbalanced data." (2013).

[19] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp.1263–1284, June 2009.

[20] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalance data: A review," International Journal of Pattern Recognition and Artificial Intelligence, vol. 23, no. 4, pp. 687–719, June 2009.

[21] M. Wasikowski and X. wen Chen, "Combating the small sample class imbalance problem using feature selection," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1388–1400, October 2009.

[22] D. M. Farid, A. Now´e, and B. Manderick, "A new data balancing method for classifying multi-class imbalanced genomic data," 25th Belgian-Dutch Conference on Machine Learning (Benelearn), pp. 1–2, 12-13 September 2016.

[23] C. Beyan and R. Fisher, "Classifying imbalanced data sets using similarity based hierarchical decomposition," Pattern Recognition, vol. 48, no. 5, pp. 1653–1672, May 2015.

[24] Jorma Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution," in 8th Conference on AI in Medicine in Europe. Cascais, Portugal: Springer Berlin / Heidelberg, 2001, vol. 2001, pp. 63-66

[25] Show-Jane Yen and Yue-Shin Lee, "Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset," in International Conference on Intelligent Computing. Kunming, China, 2006, pp. 731-740.

[26] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou, "Exploratory undersampling for class-imbalance learning," IEEE Transactions on Systems and Man and and Cybernetics and Part B, vol. 39, pp. 539-550, 2009

[27] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling TEchnique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[28] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in 2005 International

Conference on Intelligent Computing. Hefei, China: Springer-Verlag, 2005, pp. 878-887.

[29] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard, "A study of the behavior of several methods for balancing machine learning training data," SIGKDD Exploration, vol. 6, pp. 20-29, 2004.

[30] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok 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. Bangkok, Thailand: Springer-Verlag, 2009, vol. 5476, pp. 475-482.

[31] Enislay Ramentol, Yaile Caballero, Rafael Bello, and Francisco Herrera, "SMOTE-RSB*: A Hybrid Preprocessing Approach based on Oversampling and Undersampling for High Imbalanced Data-Sets using SMOTE and Rough Sets Theory," Knowledge and Information Systems, vol. 33, pp. 245-265, 2012.

[32] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," in 2008 International Joint Conference on Neural Networks. Hong Kong, Hong Kong Special Administrative Region of the Peoples Republic of China, 2008, pp. 1322-1328

[33] Kai Ming Ting, "An instance-weighting method to induce cost-sensitive trees," IEEE Transactions on Knowledge and Data Engineering, vol. 14, pp. 659-665, 2002.

[34] Zhi-Hua Zhou and Xu-Ying Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," IEEE Transactions on Knowledge and Data Engineering, vol. 18, pp. 63-77, 2006.

[35] Yuchun Tang, Yan-Qing Zhang, Nitesh V. Chawla, and Sven Krasser, "SVMs modeling for highly imbalanced classification," IEEE Transactions on Systems and Man and and Cybernetics and Part B: Cybernetics, vol. 39, 2009.

[36] Leo Breiman, "Bagging predictors," Machine Learning, vol. 24, pp. 123-140, 1996.

[37] Shuo Wang and Xin Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in IEEE Symposium Series on Computational Intelligence and Data Mining. Nashville TN, USA, 2009, pp. 324-331.

[38] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging- and Boosting- and and Hybrid-Based Approaches," IEEE Transactions on Systems and Man and and Cybernetics and Part C: Applications and Reviews, 2011.

[39] Robert E. Schapire and Yoram Singer, "Improved boosting algorithms using confidence-rated predictions," Machine Learning, vol. 37, pp. 297-336, 1999.

[40] Yoav Freund and Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol. 55, pp. 119-139, 1997.

[41] Nitesh V. Chawla, Aleksander Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in 7th European Conference on Principles and Practice of Knowledge Discovery in Databases. Cavtat Dubrovnik, Croatia, 2003, pp. 107-119

[42] Hongyu Guo and Herna L Viktor, "Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach," SIGKDD Explorations, vol. 6, pp. 30-39, 2004.

[43] Shengguo Hu, Yanfeng Liang, and Lintao Ma, "MSMOTE: Improving classification performance when training data is imbalanced," in 2nd International Workshop on Computer Science and Engineering. Qingdao, China, 2009, pp. 13-17.

[44] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," IEEE Transactions on Systems and Man and Cybernetics and Part A, vol. 40, pp. 185-197, 2010.

[45] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, and Francisco Herrera, "EUSBoost: Enhancing Ensembles for Highly Imbalanced Data-sets by Evolutionary Undersampling," Pattern Recognition, vol. 46, pp. 3460-3471, 2013.

[46] M. S. Santos, P. H. Abreu, P. J. Garc´ıa-Laencina, A. Sim˜ao, and A. Carvalho, “A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients,” Journal of Biomedical Informatics, vol. 58, pp. 49–59, September 2015.

[47] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip et al., “Top 10 algorithms in data mining,” Knowledge and information systems, vol. 14, no. 1, pp. 1–37, 2008.

[48] Y. Lu, Y.-m. Cheung, and Y. Y. Tang, “Hybrid sampling with bagging for class imbalance learning,” in Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 2016, pp. 14–26.

[49] R. Barandela, R. M. Valdovinos, and J. S. S´anchez, “New applications of ensembles of classifiers,” Pattern Analysis & Applications, vol. 6, no. 3, pp. 245–256, 2003.

[50] J. Alcal´a-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garc´ıa, L. S´anchez, and F. Herrera, “Keel data-mining software tool: Dataset repository, integration of algorithms and experimental analysis framework,” Journal of Multiple-Valued Logic and Soft Computing, vol. 17, no. 2-3, pp. 255–287, 2011.

[51] Breiman, Leo. "Bagging predictors." Machine learning 24.2 (1996): 123-140.

[52] Erich Schubert, A Framework for Clustering Uncertain Data. Proceedings of the VLDB Endowment, 8(12): 1976–1979, 2015.

[53] Aly, Mohamed A., and Amir F. Atiya. "Novel methods for the feature subset ensembles approach." International Journal of Artificial Intelligence and Machine Learning 6.4 (2006): 1-7.