Comparison of Approaches to Alleviate Problems with High-Dimensional and Class-Imbalanced Data

Ahmad Abu Shanab, Taghi M. Khoshgoftaar, Randall Wald, and Jason Van Hulse
Florida Atlantic University, Boca Raton, Florida, 33431
Email: aabusha@fau.edu, khoshgof@fau.edu, rwald1@fau.edu, jvanhulse@gmail.com

***Abstract*—Two of the most challenging problems in data mining are working with imbalanced datasets and with datasets which have a large number of attributes. In this study we compare three different approaches for handling both class imbalance and high dimensionality simultaneously. The first approach consists of sampling followed by feature selection, with the training data being built using the selected features and the original (unsampled) data. The second approach is similar, except that it uses the sampled data (and selected features) to build the training data. In the third approach, feature selection takes place before sampling, and the training data is based on the sampled data. To compare these three approaches, we use seven groups of datasets covering different application domains, employ nine feature rankers from three different families, and generate artificial class noise to better simulate real-world datasets. The results differ from an earlier work and show that the first and third approaches perform, on average, better than the second approach.**

I. Introduction

Class imbalance and high-dimensional datasets are two of the major problems in machine learning. Many important datasets are characterized by class imbalance, where there are few cases of the positive class, also called the class of interest (e.g., fraudulent network connections, patients with cancer, fault-prone software modules, etc.), and many more cases of the negative class (legitimate network connections, patients without cancer, non-fault-prone software modules, etc.). This causes the classifier to perform poorly, because many classifiers assume that the classes are equal in number and some performance metrics reach their maximum value without properly balancing the weight of each class. This is especially unfortunate because these imbalances tend to most hurt the minority class, which is the most important class. Before performing data mining on such datasets, measures must be taken to resolve the class imbalance.

High dimensionality is another problem encountered in many real-world datasets. This problem is found when a dataset has a large number of attributes, sometimes exceeding the number of instances/cases. This causes a problem because usually, most of these attributes are redundant (containing information already represented in other attributes) or useless (not having much correlation with the class) for building an inductive model. Better performance can be obtained if the redundant and useless attributes are removed. In the past few years, these two problems (class imbalance and high dimensionality) have received a lot of attention. Much work has been done towards understanding, handling, and alleviating each problem separately. Little work, however, has addressed datasets which are characterized by having the two problems simultaneously.

In this study, we investigate three approaches to combat class imbalance and high dimensionality. All approaches combine feature selection and sampling; the difference between one approach and another is the order (whether sampling takes place before or after feature selection) and the dataset (original or sampled) used for building a classifier.

In the first approach, sampling takes place before feature selection is performed, and then a classifier is built using the features selected and the original data (i.e., the sampled data is used when performing feature selection, but not for actually building the classifier). In the second approach, sampling also takes place first, then feature selection is performed; however, the classifier is built using the features selected and the sampled data. On the other hand, in the third approach, feature selection takes place first before sampling is performed, and then a classifier is built using the features selected and the sampled data.

Clearly, two other approaches can be considered, where only one technique (feature selection or sampling) is used alone. However, these options are not considered in this paper as all datasets investigated are both imbalanced and exhibit high dimensionality. Because of this, sampling should be performed to alleviate class imbalance and feature selection performed to cope with high dimensionality. Note that the "feature selection is followed by sampling, then a classifier is built using the original data" approach (which may seem to fit with the first three approaches discussed earlier) is equivalent to feature selection alone, hence its exclusion from this paper.

An additional problem exhibited by many real-world datasets is noise. Noise consists of incorrect or missing values in either the independent attributes or the class labels of instances in the dataset; these are called attribute noise and class noise, respectively. Noise can be caused by faulty sensors, problems with data entry, computer errors, and other sources. While many classifiers perform well on noise-free data, it is important to test them on more realistic, noisy datasets, since this better simulates how they would be used in the field. Thus, all experiments in this paper were performed on data which was first determined to be free of noise and then had artificial class noise added in a controlled fashion. This way, the results can be used to determine which of the three approaches outlined above are most effective when dealing with noisy data which also exhibits the problems of imbalance and high dimensionality.

Contributions: The primary contributions of this paper are as follows:

i. Compare three approaches to combining feature selection and data sampling and determine which one performs best across many real-world datasets from different application domains.
ii. Inject artificial class noise into all datasets to better see how the approaches perform on less-than-perfect data.
iii. Employ nine different feature rankers from three different families to ensure results will generalize.

The remainder of this paper is organized as follows. Section II presents related work. Section III introduces the methodology for our experiments, including the feature selection techniques, the sampling technique, the classifiers, the datasets and noise injection mechanism, and the classification performance metrics. Section IV presents our experimental results along with statistical analysis of the results. Finally, conclusions and future work are presented in Section V.

II. Related work

Class imbalance is one of the major problems in data mining. It is very important to alleviate the skewed class distribution in datasets that suffer from class imbalance, because doing so will help improve classifier accuracy. The primary technique used to deal with this problem is known as sampling, where the dataset is transformed into a more balanced one by adding or removing instances. Another technique used in dealing with class imbalance is the use of cost-sensitive learners [3]. In these learners, costs are assigned to misclassification types (false positive and false negative), and a higher cost is assigned to misclassification of the class of interest (false negative). However, it is difficult to determine in advance the appropriate costs of misclassification.

A comprehensive study on different sampling techniques was performed by Kotsiantis [9], Guo [5], and Van Hulse [14], including both oversampling and undersampling techniques (which add instances to the minority class and remove instances from the majority class, respectively), and both random and directed forms of sampling. In our paper, we use random undersampling (RUS), where instances from the majority class are deleted randomly until the number of majority-class instances is equal to the number of minority-class instances (that is, the final ratio is 50:50). Despite its simplicity, RUS has been shown to be a very effective sampling technique [8].

High dimensionality is another problem hindering the data mining process, requiring extensive computation and delaying the learning process. Feature selection, whereby only a subset of the original features are used for building a classifier, is the most popular technique for handling high-dimensional data. The process of feature selection involves choosing the best features for performing classification. There are two broad categories of feature selection: filter-based techniques and wrapper-based techniques. Filter-based techniques use only the dataset itself to pick the best features; often, this involves using different statistical measures to determine which features best predict the class. On the other hand, wrapper-based techniques use a classifier to directly find the subset of features which performs best. This classifier is usually the same one which will be used for building the final model. Much research has been done on feature selection, comparing different techniques and finding the best percentage of features to include in datasets. A survey on the concepts and algorithms of feature selection can be found in the work of Liu and Yu [11].

These two problems (class imbalance and high dimensionality) have received a lot of attention in the past few years. However, most of the work has focused on dealing with each problem separately. Few studies focused on real-world datasets suffering from both problems simultaneously. Some [12] restrict themselves to a small domain of feature selection, such as text mining, where only binary features (e.g., presence or absence of a word within a document) are found. Others [1] only consider one possible order of feature selection and sampling, without examining the importance of this order.

In their recent work Khoshgoftaar et al [7] investigated both feature selection and data sampling together on datasets from the software engineering field. They proposed four data processing approaches based on two main questions, whether feature selection or sampling should come first and whether to use original or sampled data. In this preliminary study employing six commonly-used feature rankers, they came to the conclusion that performing sampling before feature selection performs better, on average, than data sampling after feature selection. The present study differs from this earlier work in several key ways: (1) this study examines datasets from a number of different application domains, all of which exhibit high dimensionality and class imbalance; (2) this study injects class noise into the data, to more accurately represent realistic data; (3) this study examines nine different feature rankers from three different families, making the results more generalizable; and (4) this study comes to different conclusions than the previous work, possibly as a result of the above.

III. Methodology

A. Feature Ranking Techniques

In this paper, we examine filter-based feature rankers, since wrapper-based techniques can be very computationally expensive. In particular, three families of filter-based feature rankers are used: three "commonly-used" rankers found throughout the literature (chi-square (CS), information gain (IG), ReliefF (RF)), five "threshold-based" feature rankers recently proposed by our group [15] (Mutual Information (MI), Kolmogorov-Smirnov Statistic (KS), Deviance, Area Under the ROC Curve (AUC), Area Under the Precision-Recall Curve (PRC)), and a technique not often used for ranking, signal to noise (S2N). These specific rankers were chosen because previous research by our team has shown them to be the best-performing members of each family [2]. An overview of each ranker family is provided below.

1) Commonly-Used Rankers: Since these rankers are commonly used throughout the literature and for space considerations, only the general outline of these rankers is provided; the interested reader may consult the references for full details.

i. Chi-Squared (CS) is based on the χ^2 statistic, and it evaluates features independently with respect to the class labels. The larger the chi-squared, the more relevant the feature is with respect to the class. The values of the features must first be discretized into a number of intervals using some discretization method [10]. The chi-squared value of each feature is computed as:

$$\chi^2 = \sum_{i=1}^{I} \sum_{j=1}^{B} \frac{\left[A_{ij} - \frac{R_i \times B_j}{N}\right]^2}{\frac{R_i \times B_j}{N}}$$

where I denotes the number of intervals, B the number of classes, N the total number of instances, R_i the number of instances in the ith interval, B_j the number of instances in the jth class, and A_{ij} the number of instances in the ith interval and jth class. Note that for the chi-square approximation to be valid, the test requires a sufficient sample size.

ii. Information Gain (IG) is a commonly used measure in the fields of information theory and machine learning. IG measures the number of bits of information gained about the class prediction by knowing a feature's value when predicting the class. To calculate the information gain of a given feature X with respect to the class attribute Y, one must know the uncertainty about the value of Y both on its own and when considered in conjunction with the value of X. These are measured by the entropy of Y, $H(Y)$, and by the conditional entropy of Y given X, $H(Y|X)$, respectively. The entropy of Y (which consists of classes Y_1 and Y_2) is given by:

$$H(Y) = -\sum_{i=1}^{k} P(Y = Y_i) \log_2(P(Y = Y_i))$$

The conditional entropy of Y given X (consisting of values $X_1, X_2, \ldots, X_r$) is

$$H(Y|X) = -\sum_{j=1}^{r} P(X = X_j) H(Y|X = X_j)$$

The information gain of feature X is defined as:

$$IG(X) = H(Y) - H(Y|X)$$

Thus, the level of a feature's significance is determined by how great is the decrease in entropy of the class after it is considered with the corresponding feature.

iii. ReliefF (RF) is a feature selection method which estimates the importance of features by considering how much their values change when comparing a randomly-chosen instance I_0 with its nearest hit, H (an instance from the same class) and its nearest miss, M (one from a different class). Relief's estimate $W[X]$ of attribute X is obtained by:

$$\begin{aligned} W[X] \quad = \quad & P(\text{different value of } X \mid \\ & \quad \text{nearest instance from different class}) \\ & -P(\text{different value of } X \mid \\ & \quad \text{nearest instance from same class}) \end{aligned}$$

ReliefF is an extension of the Relief algorithm to handle noise and multiclass datasets which finds one near miss, $M(B)$, for each different class instead of one miss M from a single different class and finds their weighted average to compute their contribution to updating $W[X]$.

2) Threshold-Based Feature Rankers: The family of threshold-based feature rankers consists of a novel approach to permit the use of a classification performance metric as a feature ranker [15]. Note that while none of these feature rankers use a classifier, they do use the feature values (normalized to lie between 0 and 1) as a posterior probability, choosing a threshold and "classifying" instances based directly on the values of the feature being examined. Classifier performance metrics are then used to evaluate the quality of the feature. In effect, this allows the use of the performance metrics to describe how well the feature correlates with the class; since no actual classifiers are built, this still qualifies as filter-based feature selection.

i. Mutual Information (MI) computes the mutual information criterion with respect to the number of times a feature value and a class co-occur, the feature value occurs without the class, and the class occurs without the feature value. Mutual information is defined as:

$$MI = \max_{t \in [0,1]} \sum_{\hat{y}^t \in \{P,N\}} \sum_{y \in \{P,N\}} p\left(\hat{y}^t, y\right) \log \frac{p\left(\hat{y}^t, y\right)}{p\left(\hat{y}^t\right) p(y)}$$

where $y(x)$ is the actual class of instance x, $\hat{y}^t(x)$ is the predicted class based on the value of the attribute X_j at a threshold t,

$$p\left(\hat{y}^t = \alpha, y = \beta\right) = \frac{\left|\left\{\left(x \left| \hat{X}^j(x) = \alpha \right.\right) \cap \left(y(x) = \beta\right)\right\}\right|}{|P| + |N|}$$

$$p\left(\hat{y}^t = \alpha\right) = \frac{|\{(x|y(x) = \alpha)\}|}{|P| + |N|}$$

$$\alpha, \beta \in \{P, N\}$$

Note that the class (actual or predicted) can be either positive (P) or negative (N).

ii. The Kolmogorov-Smirnov Statistic (KS) measures a feature's relevance by dividing the data into clusters based on the class and comparing the distribution of that particular attribute among the clusters. It is effectively the maximum difference between the curves generated by the true positive and false positive rates ($TPR(t)$ and $FPR(t)$) of the ersatz "classifier" as the decision threshold changes from 0 to 1, and its formula is given as follows:

$$KS = \max_{t \in [0,1]} |TPR(t) - FPR(t)|$$

iii. Deviance (Dev) is the minimum residual sum of squares based on a threshold t. It measures the sum of the squared errors from the mean class given a partitioning of the space based on the threshold t. As it represents to total error found in the partitioning, lower values are preferred.

iv. Area Under the ROC Curve (AUC), the area under the receiver operating characteristic (ROC) curve, is a single-value measure based on statistical decision theory and was developed for the analysis of electronic signal detection. It is the result of plotting $FPR(t)$ against $TPR(t)$. In this study, ROC is used to determine each feature's predictive power. ROC curves are generated by varying the decision threshold t used to transform the normalized attribute values into a predicted class. That is, as the threshold for the normalized attribute varies from 0 to 1, the true positive and false positive rates are calculated.

v. Area Under the PRC Curve (PRC), the area under the precision-recall characteristic curve, is a single-value measure depicting the trade-off between precision and recall. It is the result of plotting $TPR(t)$ against precision, $Pre(t)$. Its value ranges from 0 to 1, with 1 denoting a feature with highest predictive power. The PRC curve is generated by varying the decision threshold t from 0 to 1 and plotting the recall (x-axis) and precision (y-axis) at each point in a similar manner to the ROC curve.

3) Signal-To-Noise: Signal to Noise (S2N) is a simple univariate ranking technique which defines how well a feature discriminates two classes. S2N, for a given feature, separates the means of the two classes relative to the sum of their standard deviation. S2N is obtained for each feature using this formula:

$$S2N = \frac{\mu_+ - \mu_-}{\sigma_+ + \sigma_-}$$

Where μ_+ and μ_- are the mean values for the feature from the positive class and negative class, respectively, and σ_+ and σ_- are the corresponding standard deviations.

B. Sampling Techniques

Sampling is a family of preprocessing techniques used for modifying a dataset to improve its balance, to help resolve the problem of class imbalance. There are four major classes of sampling techniques, depending on two choices: whether the sampling will be under-sampling (removing samples from the majority) or oversampling (adding samples to the minority), and whether the sampling will be random (removing/adding arbitrary samples) or focused/algorithmic (e.g., removing majority samples near the class boarder, or adding artificially-generated minority samples). In this paper, due to space considerations (and prior research showing its effectiveness), we used random undersampling [7], which deleted instances from the majority

TABLE I
DATA SETS

Data set	# attributes	# instances	% positive	% negative
Lung cancer	12534	181	17.1	82.9
ALL	12559	327	24.2	75.8
Lung clean	12601	132	17.4	82.6
Internet Ad	1559	3279	14.0	86.0
Musk	167	6598	15.4	84.6
Satimage-4	37	6435	9.7	90.3
Optdigits-8	65	5620	9.9	90.1

class until the class ratio was 50:50 majority:minority. Future research will consider a wider range of sampling techniques and balance levels.

C. Classifiers

Five classifiers were used in this study: Naïve Bayes, Multilayer Perceptron, 5-Nearest Neighbor, Support Vector Machines, and Logistic Regression. These were selected due to their prevalence in the literature and their absence of built-in feature ranking which might interfere with the feature ranking performed in the experiments. All classifiers were built using the Weka machine learning software [6], using the default parameters unless noted otherwise.

Naive Bayes (NB) is a simple Bayesian classifier which uses Bayes's Theorem to find the posterior probability of an instance being in each class based on its feature values. Although mathematically it depends on all features being independent of (unrelated to) one another, in practice it is a very effective classifier on a wide variety of datasets. Multilayer Perceptron (MLP) is a neural-net-based classifier, using a simple feedforward network with distinct layers. Each layer is fully connected to the ones before and after it, and neurons compute their outputs by finding the weighted sum of their inputs and passing this to a sigmoid function. In these experiments, the `hiddenLayers` parameter was set to 3 to build a network with one hidden layer containing three nodes, and the `validationSetSize` parameter was set to 10 so that the classifier would leave 10% of the instances out to determine when to stop training. 5-Nearest Neighbor (5NN) is an instance-based learner which finds the five instances in the training set closest to the test instance (using the Euclidean distance metric), and then the classes and weights of these nearest neighbors are used to predict the class of the test instance (i.e., using Weka's kNN classifier, the `weightByDistance` parameter was set to 1, and `k` was set to 5). Support Vector Machines (SVM) work by finding the optimal hyperplane which best divides the two classes in N-dimensional feature space (where N is the number of features). In Weka, the complexity parameter `c` was set to 5.0 and `buildLogisticModels` was set to true. Logistic Regression (LR) is a very simple classifier; it shares much in common with a linear regression, with the output run through a logistic function.

D. Data sets

Table I lists the seven datasets used in this study, including their characteristics in terms of the total number of attributes, number of instances, percentage of positive instances, and percentage of negative instances. They are all binary class datasets. That is, for all the datasets, each instance is assigned one of two class labels.

Three cancer gene expression datasets are considered: lung cancer, Acute Lymphoblastic Leukemia (ALL), and lung clean. The Lung Cancer dataset is a classification of malignant pleural mesothelioma (MPM) vs. adenocarcinoma (ADCA) of the lung, and consists of 181 tissue samples (31 MPM, 150 ADCA) [16]. The acute lymphoblastic leukemia dataset consists of 327 tumor samples of which 79 are positive (24.2%). The Lung Clean dataset was derived from a noisy lung cancer dataset containing 203 instances, including 64 (31.53%) minority instances and 139 (68.47%) majority instances. To produce a dataset that both was imbalanced and could be considered 'clean' (as defined by many classifiers having relatively near perfect classification on the dataset), a supervised cleansing process was used to reduce the original lung dataset. 5-fold cross-validation was performed on the original lung dataset using a 5NN classifier, and any instances which produced a probability of membership in the opposite class that was greater than 0.1 were removed.

In addition to the gene expression sets, four datasets from the UCI Machine Learning Repository [4] were used. The first of these was the Internet Advertisements dataset, which contains 3279 instances representing images along with some keywords embedded in web pages. The independent features consist of image dimensions, phrases in the URL of the document or the image, and text occurring in or near the image's anchor tag, while the dependent feature determines whether an image is an advertisement ("ad") or not ("noad"). Musk data is a set of different conformations of various molecules for predicting drug activity, in particular whether new molecules will be musks or non-musks. Also used in the experiments for this study are two datasets from the domain of image recognition, Optidigits-8 and Satimage-4.

We selected these datasets for this study not only because they are from different application domains and show different class distributions, but also because these datasets are relatively clean. Due to the fact that the datasets are relatively clean, we avoid any problems associated with injecting noise into datasets that are already noisy. Moreover, to make the study more comprehensive, 63 datasets were generated by injecting noise into the mentioned 7 datasets. We used the same noise injection procedure reported by Van Hulseet al [13] where two parameters, α and β, control noise injection. The first parameter controls the overall noise level (in this study we used $\alpha \in \{40\%, 50\%\}$), and the second parameter β controls the level of noise affecting the positive class (this study used $\beta \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$). Note that the case with α = 50% and β = 100% was excluded, because this would result in a dataset where no instances in the minority class exist. Thus, nine different noisy versions of each of the seven initial datasets were generated, with varying levels of noise and class imbalance depending on parameters α and β. Results from these nine different views on each dataset were averaged together in our experiments.

E. Classifier Performance Metrics

We used two performance metrics to evaluate the performance of learners: the area under the receiver operating characteristic curve (AUC), and the area under the precision-recall characteristic curve (PRC). These two performance metrics were chosen because from past experiments it was observed that while two learners might show similar performance using AUC, one of them might show better performance using PRC. Moreover, they have been proven to be statistically consistent. Both metrics are based on TPR and FPR.

Note that while AUC and PRC are used as both feature rankers and as classifier performance metrics, these uses are disconnected from each other. When used as a feature ranker, these techniques are applied to normalized feature values, examining only one feature at a time to determine how well that feature predicts the class. When used as a classifier performance metric, however, the output of the classifier (built using all of the selected features) is used and evaluated.

TABLE II
SAMPLING PLACEMENT APPROACHES

1	Sampling then Feature Selection, training set uses original data
2	Sampling then Feature Selection, training set uses sampled data
3	Feature Selection then Sampling, training set uses sampled data

IV. RESULTS AND ANALYSIS

As mentioned earlier, seven datasets were used in this experiment. These datasets are relatively clean to avoid validity problems caused by injecting noise into a dataset that already has noise. Nine filter-based feature ranking techniques are applied, and five learners are used (NB, MLP, 5NN, SVM, and LR). The selection of these five learners was based on the fact that they are commonly used in the literature. We investigated three approaches that are used to deal with both class imbalance and high dimensionality. All approaches combine feature selection and sampling; the difference between one approach and another is the order (whether sampling takes place before or after feature selection) and the dataset (original or sampled) used for classification.

In the first approach – assigned code 1 – sampling takes place first before feature selection is performed, and then a classifier is built using the selected features and the original (unsampled) data. In the second approach – assigned code 2 – sampling also takes place first before feature selection is performed, however, a classifier is built using the selected features and the sampled data. On the other hand, in the third approach – assigned code 3 – feature selection takes place first before sampling is performed, and then a classifier is built using the selected features and the sampled data. We excluded two other approaches, where only one technique (feature selection or sampling) is used alone, because all datasets investigated in this paper are imbalanced and exhibit high dimensionality. Both feature selection and sampling are necessary to help alleviate class imbalance and cope with high dimensionality.

We used AUC and PRC to evaluate the performance of the classifiers. In the experiments, four runs of five-fold cross-validation were performed. Only the combined results of this cross-validation are presented in the tables; these results show the averages of all nine noise injection patterns for each dataset, as well as the averages of all nine feature rankers. Further discussion of the breakdown based on the different noise injection patterns could not be included due to space considerations.

Table III represents the average AUC for every classification model constructed over the four runs of five-fold cross-validation, across all nine levels of injected noise. We also present (1) the average performance (last column of the tables) of each of the three approaches for each learner over the seven datasets, and (2) the average performance (last row of the tables) of each dataset over the five learners and across three approaches. Table IV represents the average PRC for every classification model constructed over the four runs of five-fold cross-validation, using the same structure as Table III. In both tables "Sample Placement" is abbreviated as "SP" for space considerations, and bold values represent the best performance for that combination of learner and dataset across all three sampling placement approaches.

We also performed an ANalysis Of VAriance (ANOVA) test to find statistically significant patterns in these data. The results are presented in Table V. In this analysis, factor A is the five learners, factor B is the nine feature rankers, and factor C is the choice of approach. Since a significance factor of 5% was chosen, the "Pr $>$ F" value must be less than this value (e.g., 0.05) for the result to be significant.

TABLE III
CLASSIFICATION PERFORMANCE IN TERMS OF AUC

Learner	SP	Lung cancer	ALL	Internet Ad	Musk	Lung clean	Sat image-4	optdigits-8	Average
NB	1	.93641	.96365	**.93166**	**.81440**	.86697	**.87574**	.95271	.90594
	2	.93737	.95528	.93046	.80719	.86191	.86950	.94989	.90166
	3	**.96700**	**.96442**	.93009	.81020	**.88999**	.86603	**.95285**	**.91151**
MLP	1	.88110	**.91230**	.89101	**.90627**	**.84734**	**.89465**	**.95174**	**.89777**
	2	.89076	.89294	.89481	.88271	.83647	.87871	.93777	.88774
	3	**.91227**	.91049	**.89687**	.88249	.85261	.88044	.93843	.89623
5-NN	1	.92916	.91754	.87159	.84678	.88669	.79460	.93901	.88363
	2	.93160	.93227	.88025	**.86946**	.87803	**.85831**	.96650	.90235
	3	**.94519**	**.93797**	**.88258**	.86857	**.89495**	.85755	**.96656**	**.90763**
SVM	1	.89173	**.92069**	.75321	.73748	**.84357**	.60273	.82973	.79702
	2	.87492	.90035	.88621	**.88718**	.81722	.72671	**.95525**	.86398
	3	**.89403**	.91845	**.88799**	.88596	.83784	**.72796**	.95520	**.87249**
LR	1	**.79500**	**.89272**	**.90262**	**.89864**	**.75620**	**.73187**	**.96056**	**.84823**
	2	.77562	.85586	.87721	.88795	.75220	.72360	.95306	.83222
	3	.75854	.86331	.88009	.88603	.72825	.72418	.95289	.82761
Average		.88805	.91588	.88644	.85809	.83668	.80084	.94414	.87573

To further examine interactions among and within the known-meaningful factors, we used Tukey's honestly significantly difference test, which lists those values within a factor or group of factors which are significantly different from one another. These results are presented in Tables VI and VII, with values listed in order from highest to lowest performance and values with the same subscript being statistically indistinguishable from one another.

The results demonstrate that

i. Using the AUC performance metric, approach 3 (feature selection then sampling) performs best for most learners and datasets, with approach 1 (sampling then feature selection, with the training data built on the unsampled data) a close second and approach 2 a distant third.
ii. Using the PRC performance metric, approach 1 performs best, with approach 3 right behind it and approach 2 performing worst.
iii. The ANOVA results show that different learners had significantly different performances, as did different approaches. The interaction term between learner and approach was also significant (meaning the patterns of results found within one would change when the data was grouped according to the other). However, none of the terms pertaining to choice of ranker were

TABLE IV
CLASSIFICATION PERFORMANCE IN TERMS OF PRC

Learner	SP	Lung cancer	ALL	Internet Ad	Musk	Lung clean	Sat image-4	optdigits-8	Average
NB	1	**.85472**	**.91588**	**.85254**	**.48826**	.73527	**.40195**	**.75310**	**.71453**
	2	.75037	.86474	.84636	.46496	.63829	.37850	.73407	.66818
	3	.84709	.89242	.84348	.46993	**.73858**	.37053	.74588	.70113
MLP	1	.75305	**.83083**	**.80597**	**.74996**	**.68377**	**.50324**	**.84227**	**.73844**
	2	.71111	.76799	.78538	.63765	.61845	.41774	.73106	.66705
	3	**.78297**	.81679	.78847	.63879	.67109	.42077	.73555	.69349
5-NN	1	.83107	.82593	**.73612**	**.67796**	.75756	.44207	.81488	.72651
	2	.80355	.83369	.71768	.67383	.70865	**.45847**	.83334	.71846
	3	**.86075**	**.85238**	.72307	.67563	**.78255**	.45562	**.83594**	**.74085**
SVM	1	**.77590**	**.83355**	.60578	.42345	**.67506**	.12855	.51738	.56567
	2	.67634	.78551	.78251	.62631	.55726	.17801	.78016	.62659
	3	.74063	.82987	**.79182**	**.62711**	.63250	**.17960**	**.78071**	**.65461**
LR	1	**.59001**	**.78185**	**.82856**	**.69167**	**.46809**	**.17710**	**.83209**	**.62420**
	2	.45360	.71643	.74398	.64387	.40966	.17401	.78960	.56159
	3	.45699	.73997	.76125	.64372	.40474	.17494	.79155	.56759
Average		.72588	.81919	.77420	.60887	.63210	.32407	.76784	.66459

TABLE V
ANOVA RESULTS

Factor	DF	Type I SS	Mean Square	F Value	Pr > F
A	4	7.33422	1.83355	260.25	< .0001
B	8	0.09723	0.01215	1.73	0.0873
A x B	32	0.06892	0.00215	0.31	0.9999
C	2	0.40413	0.20206	28.68	< .0001
A x C	8	1.90752	0.23844	33.84	< .0001
B x C	16	0.04704	0.00294	0.42	0.9789
A x B x C	64	0.01207	0.00019	0.03	1.0000

TABLE VI
TUKEY'S HONESTLY SIGNIFICANTLY DIFFERENT RESULTS, BY LEARNER

Learner	Performance Metric	Rank of Approaches
NB	AUC	$3_A, 1_{AB}, 2_B$
	PRC	$1_A, 3_A, 2_B$
MLP	AUC	$1_A, 3_A, 2_B$
	PRC	$1_A, 3_B, 2_C$
5-NN	AUC	$3_A, 2_A, 1_B$
	PRC	$3_A, 1_A, 2_A$
SVM	AUC	$3_A, 2_A, 1_B$
	PRC	$3_A, 2_A, 1_B$
LR	AUC	$1_A, 2_B, 3_B$
	PRC	$1_A, 3_B, 2_B$

statistically significant at 5%, so further consideration of these is unnecessary.

iv. Looking more closely at this interaction in terms of the Tukey's honestly significantly different criterion (using both AUC and PRC values), it can be seen that approach 3 was best or tied for best when using 5-NN and SVM; with MPL and LR, 1 was best or tied for best; and with NB, they were both statistically insignificant for both performance metrics.

v. The best average AUC for all datasets was obtained when the learner NB was used with strategy 3 (0.96700, on the Lung Cancer dataset), and the worst average performance for all data sets was obtained when the learner SVM was used with strategy 1 (0.60273, on the Sat image-4 dataset).

V. CONCLUSION

In this paper, we compared three approaches to deal with highly imbalanced datasets that also exhibit high dimensionality. The experiment was carried out on seven relatively clean datasets, from different application fields. We injected noise into these datasets, and used nine feature ranking techniques from three families and one sampling technique. We applied the three approaches mentioned earlier to evaluate their effectiveness in dealing with high dimensionality and class imbalance in the presence of class noise. The evaluation was carried out using both area under the ROC curve and area under the PRC curve classifier performance metrics.

The experimental results suggest that either approach 1 (performing sampling first followed by feature selection and building a model based on the original data) or approach 3 (performing feature selection first followed by sampling) will give the best performance, with the best choice among these two depending on the learner and performance metric chosen. With the PRC metric or the MLP or LR learners, approach 1 was generally better, while with the AUC metric or the 5-NN or SVM learners, approach 3 was generally better. The performance of these two approaches was often statistically indistinguishable, however, so there is not a large difference.

Future research may involve conducting more experiments, using other levels of overall noise, examining more datasets from other application fields, and considering additional data sampling techniques (including undersampling and oversampling) and balance levels (e.g., both 50:50 and 65:35).

TABLE VII
TUKEY'S HONESTLY SIGNIFICANTLY DIFFERENT RESULTS, FOR ALL LEARNERS TOGETHER

Performance Metric	Rank of Approaches
AUC	$3_A, 2_B, 1_C$
PRC	$1_A, 3_A, 2_B$

REFERENCES

[1] A. Al-Shahib, R. Breitling, and D. Gilbert, "Feature selection and the class imbalance problem in predicting protein function from sequence," *Appl. Bioinformatics*, vol. 4, pp. 195–203, 2005.

[2] W. Altidor, T. M. Khoshgoftaar, and J. Van Hulse, "Exploring solutions to the combined problem of class imbalance and high dimensionality on noisy data," Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Tech. Rep., 2010.

[3] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 973–978. [Online]. Available: http://portal.acm.org/citation.cfm?id=1642194.1642224

[4] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[5] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou, "On the class imbalance problem," in *Fourth International Conference on Natural Computation, 2008. ICNC '08.*, vol. 4, Oct. 2008, pp. 192–201.

[6] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[7] T. M. Khoshgoftaar, K. Gao, and N. Seliya, "Attribute selection and imbalanced data: Problems in software defect prediction," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, vol. 1, Oct. 2010, pp. 137–144.

[8] T. M. Khoshgoftaar, C. Seiffert, J. Van Hulse, A. Napolitano, and A. Folleco, "Learning with limited minority class data," in *Sixth International Conference on Machine Learning and Applications, 2007. ICMLA 2007.*, Dec. 2007, pp. 348–353.

[9] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25–36, 2006.

[10] H. Liu and R. Setiono, "Chi2: feature selection and discretization of numeric attributes," in *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence, 1995.*, Nov. 1995, pp. 388–391.

[11] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, April 2005.

[12] L. Tang and H. Liu, "Bias analysis in text classification for highly skewed data," in *ICDM '05: Proc. Fifth IEEE Int'l Conf. Data Mining*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 781–784.

[13] J. Van Hulse and T. M. Khoshgoftaar, "Knowledge discovery from imbalanced and noisy data," *Data & Knowledge Engineering*, vol. 68, no. 12, pp. 1513–1542, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYX-4X2JSST-1/2/a426620768a8f9aeb2c196c78db95134

[14] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 935–942. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273614

[15] H. Wang, T. M. Khoshgoftaar, and K. Gao, "Ensemble feature selection technique for software quality classification," in *SEKE*, 2010, pp. 215–220.

[16] X. Wang and O. Gotoh, "Accurate molecular classification of cancer using simple rules," *BMC Medical Genomics*, vol. 2, no. 1, p. 64, 2009. [Online]. Available: http://www.biomedcentral.com/1755-8794/2/64