Feature Bagging for Outlier Detection

Aleksandar Lazarevic
United Technologies Research Center
411 Silver Lane, MS 129-15
East Hartford, CT 06108, USA
1-860-610-7560
aleks@cs.umn.edu

Vipin Kumar
Department of Computer Science,
University of Minnesota
200 Union Street SE
Minneapolis, MN 55455, USA
1-612-626-8704
kumar@cs.umn.edu

ABSTRACT
Outlier detection has recently become an important problem in many industrial and financial applications. In this paper, a novel feature bagging approach for detecting outliers in very large, high dimensional and noisy databases is proposed. It combines results from multiple outlier detection algorithms that are applied using different set of features. Every outlier detection algorithm uses a small subset of features that are randomly selected from the original feature set. As a result, each outlier detector identifies different outliers, and thus assigns to all data records *outlier scores* that correspond to their probability of being outliers. The outlier scores computed by the individual outlier detection algorithms are then combined in order to find the better quality outliers. Experiments performed on several synthetic and real life data sets show that the proposed methods for combining outputs from multiple outlier detection algorithms provide non-trivial improvements over the base algorithm.

Categories and Subject Descriptors
H.2.8 **[Database Management]**: Database Applications (data mining, scientific databases, spatial databases)

General Terms
Algorithms, Performance, Design, Experimentation.

Keywords
Outlier detection, bagging, feature subsets, integration, detection rate, false alarm.

1. INTRODUCTION
The explosion of very large databases and the World Wide Web has created extraordinary opportunities for monitoring, analyzing and predicting global economical, geographical, demographic, medical, political and other processes in the world. However, despite the enormous amount of data being available, particular events of interests are still quite rare. These rare events, very often called outliers or anomalies, are defined as events that occur very infrequently (their frequency ranges from 5% to less than 0.01% depending on the application). Detection of outliers (rare events) has recently gained a lot of attention in many domains, ranging from detecting fraudulent transactions and intrusion detection to direct marketing, and medical diagnostics. For example, in the network intrusion detection domain, the number of cyber attacks on the network is typically a very small fraction of the total network traffic. In medical databases, when classifying the pixels in mammogram images as cancerous or not, abnormal (cancerous) pixels represent only a very small fraction of the entire image. Among all users that visit an e-commerce web site, those that actually purchase are quite rare - for example less than 2% of all people who visit Amazon.com's website make a purchase, and this is much higher than the industry average. Although outliers (rare events) are by definition infrequent, in each of these examples, their importance is quite high compared to other events, making their detection extremely important.

The problem of detecting outliers (rare events) has been variously called in different research communities: novelty detection [23], chance discovery [24], outlier/anomaly detection [3, 5, 10, 19, 27, 36], exception mining [29], mining rare classes [11, 16-18], etc. Data mining techniques that have been developed for this problem are based on both supervised and unsupervised learning. Supervised learning methods typically build a prediction model for rare events based on labeled data (the training set), and use it to classify each event [11, 16, 18]. The major drawbacks of supervised data mining techniques include (1) necessity to have labeled data, which can be extremely time consuming for real life applications, and (2) inability to detect new types of rare events. On the other hand, unsupervised learning methods typically do not require labeled data and detect outliers (rare events) as data points that are very different from the normal (majority) data based on some measure [5]. These methods are typically called outlier/anomaly detection techniques, and their success depends on the choice of similarity measures, feature selection and weighting, etc. Outlier detection algorithms can detect new types of rare events as deviations from normal behavior, but on the other hand suffer from a possible high rate of false positives, primarily because previously unseen (yet normal) data are also recognized as outliers/anomalies, and hence flagged as interesting. In this paper, we focus on unsupervised methods for outlier detection.

Many outlier detection algorithms [3, 10, 19, 27, 31] attempt to detect outliers by computing the distances in full dimensional space. However, in very high dimensional spaces, the data is very sparse and the concept of similarity may not be meaningful any-

KDD'05, August 21–24, 2005, Chicago, Illinois, USA

more [3, 6]. In fact, due to the sparse nature of distance distributions in high dimensional spaces, the distances between any pair of data records may become quite similar [6]. Thus, by using the notion of similarity in high dimensional spaces, each data record may be considered as potential outlier. It has been shown recently that by examining the behavior of the data in subspaces, it is possible to develop more effective algorithms for cluster discovery [28] and similarity search in high dimensional spaces [1, 2, 4]. It has been shown that this is also true for the problem of outlier detection [3], since in many applications only the subset of attributes is useful for detecting anomalous behavior. In the example shown in Fig. 1, data records A and B can be seen as outliers only when certain two dimensions are selected (in Fig. 1b data record A is seen as outlier, in Figure 1c data record B is observed as outlier, in Figure 1d both data records A and B may be detected as outliers), while in other two-dimensional projections they show average behavior (Fig. 1a) [3]. In addition, when significant number of features in a database is considered *noisy*, finding outliers in all dimensions typically do not result in effective detection of outliers, while at the same time it is difficult to identify a few relevant dimensions where the outliers may be observed.

Furthermore, it is well known in machine learning that ensembles of classifiers can be effective in improving overall prediction performance. These combining techniques typically manipulate the training data patterns single classifiers use (e.g. bagging [9], boosting [14]) or the class labels (e.g. ECOC [20]). In general, an ensemble of classifiers must be both diverse and accurate in order to improve prediction of the whole. In addition to classifiers' accuracy, diversity is also required to ensure that all the classifiers do not make the same errors. However, it has been shown that standard combining methods (e.g. bagging) do not improve the prediction performance of simple local classifiers (e.g. k-Nearest Neighbor) due to correlated predictions across the outputs from multiple combined classifiers [9, 20] and their low sensitivity to data perturbation. Nevertheless, local classifiers are extremely sensible to the selection of features that are used in the learning process, and prediction of their ensembles can be decorrelated by selecting different feature representations (e.g different set of features) [6, 25]. Since many outlier detection techniques that compute full dimensional distances are also local in their nature, they are also sensitive to the selection of features used in distance computation. In addition, presence of noisy and irrelevant features can significantly degrade the performance of outlier detection.

In this paper, we propose a novel feature bagging framework of combining predictions from multiple outlier detection algorithms for detecting outliers in high-dimensional and noisy data sets. Unlike standard bagging approach where the classification/regression models that are combined use randomly sampled data distributions, in this approach outlier detection algorithms are combined and their diversity is improved by sampling random subsets of features from the original feature set. Due to aforementioned sensitivity of outlier detection algorithms to the selection of features used in distance computation, each outlier detector identifies different outliers and assigns different outlier scores to data records. The outlier scores are then combined in order to find the better quality outliers than the outliers identified by single outlier detection algorithms.

It is important to note that the proposed combining framework can be applied to the set of any outlier detection algorithms or even to the set of different outlier detection algorithms. Our experimental results performed on synthetic and real life data sets have shown that the combining outlier detection algorithms provide non-trivial improvement over the base algorithm.

2. BACKGROUND AND RELATED WORK

Outlier detection algorithms are typically evaluated using the detection rate, the false alarm rate, and the ROC curves [26]. In order to define these metrics, let's look at a confusion matrix, shown in Table 1. In the outlier detection problem, assuming class "C" as the outlier or the rare class of the interest, and "NC" as a normal (majority) class, there are four possible outcomes when detecting outliers (class "C"), namely true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN).

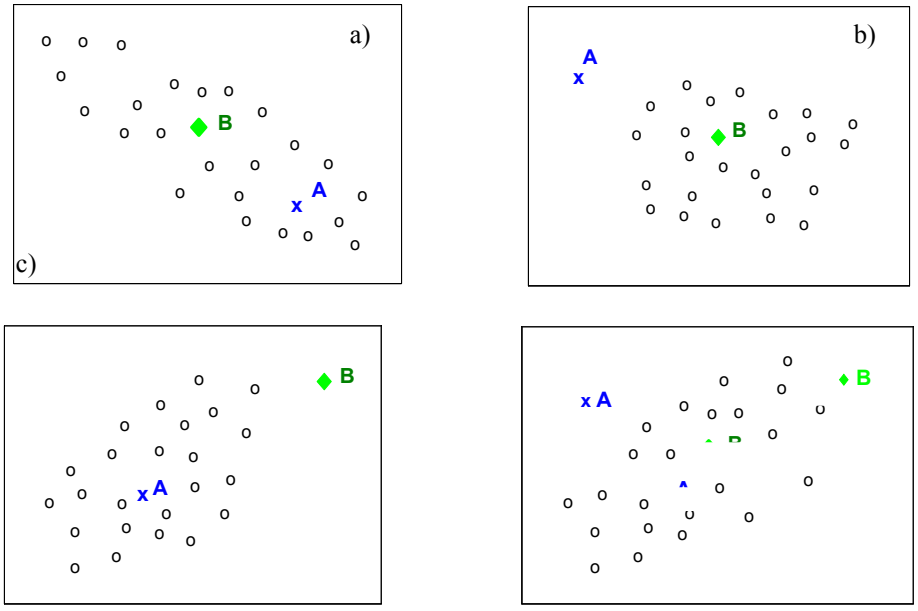

Figure 1. Different two-dimensional projections of data space reveal different set of outliers or may not reveal outliers at all.

Table 1. Confusion matrix defines four possible scenarios when classifying class "C"

	Predicted Outliers - Class C	Predicted Normal class NC
Actual Outliers - Class C	True Positives (TP)	False Negatives (FN)
Actual Normal class NC	False Positives (FP)	True Negatives (TN)

From Table 1, *detection rate* and *false alarm rate* may be defined as follows:

$$\textit{Detection rate} = TP / (TP + FN)$$

$$\textit{False alarm rate} = FP / (FP + TN)$$

Detection rate gives information about the number of correctly identified outliers, while the false alarm rate reports the number of outliers misclassified as normal data records (class NC). The ROC curve represents the trade-off between the detection rate and the false alarm rate and is typically shown on a 2-D graph (Fig. 2), where false alarm rate and detection rate are plotted on x-axis, and y-axis respectively. The ideal ROC curve has 0% false alarm rate, while having 100% detection rate (Figure 2). However, the ideal ROC curve is hardly achieved in practice, and therefore researchers typically compute detection rate for different false alarm rates and present results on ROC curves. Very often, the area under the curve (AUC) is also used to measure the performance of outlier detection algorithm. The AUC of specific algorithm is defined as the surface area under its ROC curve. The AUC for the ideal ROC curve is typically set to be 1, while AUCs of "*less than perfect*" outlier detection algorithms are less than 1. In Figure 2, the shaded area corresponds to the AUC for the lowest ROC curve.

Most of outlier detection techniques can be categorized into four groups: (1) statistical approaches, (2) distance based approaches, (3) profiling methods and (4) model-based approaches. In statistical techniques [5, 7, 12], the data points are typically modeled using a stochastic distribution, and points are determined to be outliers depending on their relationship with this model. However, most statistical approaches have limitation with higher dimensionality, since it becomes increasingly difficult and inaccurate to estimate the multidimensional distributions of the data points [3]. Distance based approaches [3, 10, 19, 27, 35, 37] attempt to overcome limitations of statistical techniques and they detect outliers by computing distances among points. Several recently proposed distance based outlier detection algorithms are based on (1) computing the full dimensional distances of points from one another using all the available features [19, 27] or only feature projections [3], and (2) on computing the densities of local neighborhoods [10]. In addition, a few clustering-based techniques have also been used to detect outliers either as side products of the clustering algorithms (points that do not belong to clusters) [2, 31] or as clusters that are significantly smaller than others [13]. In profiling methods, profiles of normal behavior are built using different data mining techniques or heuristic-based approaches, and deviations from them are considered as intrusions. Finally, model-based approaches usually first characterize the normal behavior using some predictive models (e.g. replicator neural networks [15] or unsupervised support vector machines [13, 21]), and then detect outliers as the deviations from the learned model.

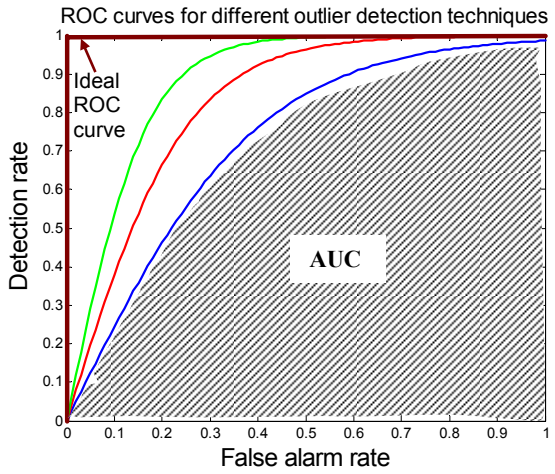

Figure 2. The ROC Curves for different detection algorithms

On the other hand, extensive research was devoted to classifier ensembles in recent years. There were numerous techniques proposed in literature for combining classification algorithms [9, 11, 14, 17, 20]. However, it is important to note here that the problem of combining outlier detection algorithms is not exactly the same to the problem of classifier ensembles due to several reasons. First, in classifier ensembles, classification algorithms deal with combining discrete outputs (class labels) typically using different types of voting techniques. In combining outlier detection algorithms, the outlier scores or rankings of the algorithms are combined instead of class labels, although some classifier ensembles also combine rankings (or class probability estimates) from single classifiers through averaging. Second, classifiers that are combined typically have complete knowledge of training data records and their labels (supervised learning) while outlier detection algorithms typically deal only with data records without any labels (unsupervised learning). However, some classifier ensembles that do not use class labels effectively (e.g. bagging) are very similar to combining outlier detection algorithms. Finally, certain classifier ensembles (e.g. boosting [14]) can control the combining process by observing the error rate, which is not possible in combining outlier detection algorithms since the label is not given and it is not known in advance what data records are really outliers.

3. OUTLIER DETECTION TECHNIQUES

Outlier detection algorithms that we utilize in this study are based on computing the full dimensional distances of the points from one another as well as on computing the densities of local neighborhoods. In our previous work [21]], we have experimented with numerous outlier detection algorithms in the problem of network intrusion detection, and we have concluded that the density based outlier detection approach (e.g. LOF) typically achieved the best prediction performance. Therefore, in this study, we have chosen the LOF approach to illustrate our findings.

3.1 Density Based Local Outlier Factor (LOF) Detection Approach

The main idea of this method [10] is to assign to each data example a degree of being outlier. This degree is called the local outlier factor (LOF) of a data example. Data points with high LOF have

more sparse neighborhoods and typically represent stronger outliers, unlike data points belonging to dense clusters that usually tend to have lower LOF values.

To illustrate advantages of the LOF approach over the simple nearest neighbor approach, consider a simple two-dimensional data set given in Figure 3. It is apparent that the density of the cluster C_2 is significantly higher than the density of the cluster C_1. Due to the low density of the cluster C_1 it is apparent that for every example p_3 inside the cluster C_1, the distance between the example p_3 and its nearest neighbor is similar to the distance between the example p_2 and the nearest neighbor from the cluster C_2, and the example p_2 will not be considered as outlier using the simple nearest neighbor (*NN*) scheme. On the other hand, *LOF* approach is able to capture the example p_2 as outlier due to the fact that it considers the density around the points. Nevertheless, the example p_1 may be detected as outlier using both NN and LOF approaches, since it is too distant from both clusters.

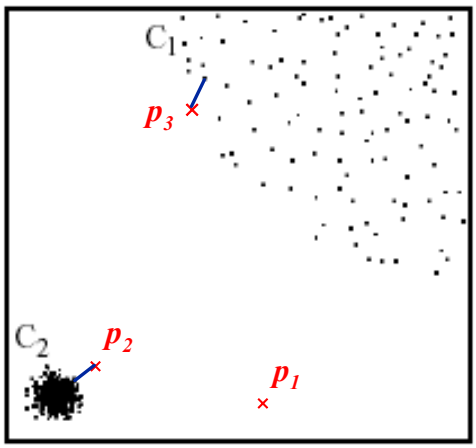

Figure 3. Advantages of the *LOF approach*

4. COMBINING OUTLIER DETECTION OUTPUTS

We propose two novel techniques for combining outlier detection algorithms. Their general framework is shown in Fig. 4. The procedure for combining outlier detection techniques proceeds in a series of T rounds, although these rounds may be run in parallel for faster execution. In every round t, the outlier detection algorithm is called and presented with a different set of features F_t that is used in distance computation. The set of features F_t is randomly selected from the original data set, such that the number of features in F_t is also randomly chosen between $\lfloor d/2 \rfloor$ and (d-1), where d is the number of features in original data set. When the number of features N_t in F_t is selected, N_t features are randomly selected without replacement from the original feature set.

Every outlier detection algorithm, as a result, outputs different outlier score vector AS_t that reflects the probability of each data record from the data set S being an outlier. For example, if $AS_t(i) > AS_t(j)$, data record x_i has higher probability of being outlier than data record x_j. At the end of the procedure, after T rounds, there are T outlier score vectors each corresponding to a single outlier detection algorithm. The function COMBINE (Figure 4) is then used to coalesce these T outlier score vectors AS_t , $t = \overline{1,T}$ into a unique anomaly score vector AS_{FINAL}, which is lastly used to assign a final probability of being an outlier to every data record from the data set.

- Given: Set S $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ $x_i \in \mathbf{X}^{\mathbf{d}}$, with labels $y_i \in Y = \{C, NC\}$, where C corresponds to outliers, NC corresponds to a normal class, and $\boldsymbol{d}$ corresponds to the dimensionality (number of features) of vector $\mathbf{X}$.
- Normalize data set S
- For $t = 1, 2, 3, 4, \ldots T$
  1. Randomly chose the size of the feature subset N_t from a uniform distribution between $\lfloor d/2 \rfloor$ and (d-1)
  2. Randomly pick, without replacement, N_t features to create a feature subset F_t
  3. Apply outlier detection algorithm O_t by employing the feature subset F_t
  4. The output of the outlier detection algorithm O_t is anomaly score vector AS_t
- Combine the anomaly score vectors AS_t and output a final anomaly score vector AS_{FINAL} as:

  AS_{FINAL} = COMBINE(AS_t), $t = 1, \ldots, T$

Figure 4. The general framework for combining outlier detection techniques

The problem of combining outlier score vectors is conceptually quite similar to the problem of meta search engines [32, 33, 34] where different rankings returned by individual search engines are combined in order to provide the pages that are most relevant to the search string. In both problems, there is no label that helps to understand how relevant the search results are and the rank of results from individual algorithms is important in the combining process, since it gives the notion of result relevance. Motivated by several approaches used in meta search engines, in this paper we explore two variants of the function COMBINE that integrates the outputs of multiple outlier detection algorithms. The first variant, denoted as *Breadth First* approach, is presented in Figure 5.

- Given: AS_t, $t = 1, \ldots, T$, and m is the size of data set S and each vector AS_t
- Sort all outlier score vectors AS_t into the vectors SAS_t and return indices Ind_t of the sorted vectors, such that $SAS_t(1)$ has the highest score and $Ind_t(1)$ is the index of the data record in S with the highest score $SAS_t(1)$
- Let AS_{FINAL} and Ind_{FINAL} be empty vectors.
- For $i = 1$ to m
  - For $t = 1$ to T
    - If the index $Ind_t(i)$ of the data record that is ranked at the i-th place by t-th outlier detection algorithm and that has the outlier score $AS_t(i)$ does not exist in the vector Ind_{FINAL}
      - Insert $Ind_t(i)$ at the end of the vector Ind_{FINAL}
      - Insert $AS_t(i)$ at the end of the vector AS_{FINAL}
- Return Ind_{FINAL} and AS_{FINAL}

Figure 5. The Breadth-First scheme for combining outlier detection scores

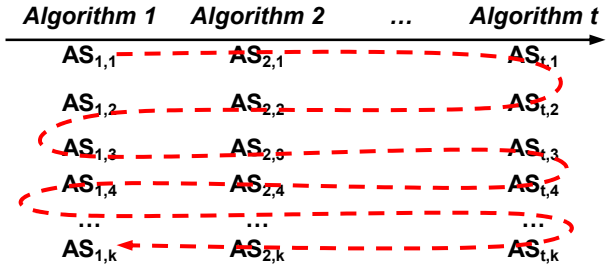

Figure 6. Illustration of the Breadth-First approach for combining outlier detection scores.

The *Breadth-First* combining method first sorts all the outlier detection vectors AS_t into the sorted vectors SAS_t and returns indices Ind_t that give the correspondence between the sorted elements of the score vectors and the original elements of the sorted vectors. For example, $Ind_t(1) = k$ means that in the t-th outlier detection score vector AS_t, data record x_k has the highest anomaly score $AS_t(k)$. Thus in Figure 6, $AS_{1,1}$ corresponds to the data record that is ranked as the most probable outlier by Algorithm 1, $AS_{1,2}$ corresponds to the data record that is ranked as the second most probable outlier by Algorithm 1, and so on.

After sorting all outlier score vectors AS_t, the *Breadth-First* approach simply takes the data records with the highest anomaly score from all outlier detection algorithms (scores $AS_{1,1}$, $AS_{2,1}$, $AS_{3,1}$, …, $AS_{t,1}$ in Figure 6) and inserts their indices in the vector Ind_{FINAL}, then takes data records with the second highest anomaly score (scores $AS_{1,2}$, $AS_{2,2}$, $AS_{3,2}$, …, $AS_{t,2}$ in Figure 6) and appends their indices at the end of the vector Ind_{FINAL}, and so on. If the index of the current data record is already in the vector Ind_{FINAL}, it is not appended again. At the end of the *Breadth-First* method, the index Ind_{FINAL} contains indices of the data records that are sorted according to their probability of being outlier, and the vector AS_{FINAL} contains these probabilities.

The final results of the *Breadth-First* method are in general sensitive to the order of outlier detection algorithms. However, the differences are minor since variations may happen only within T rankings (T is generally much smaller than the total number of data records), since at every i-th pass we go through T indices for data records ranked at i-th place in the outlier detection vector.

The second variant of the function COMBINE, denoted as *Cumulative Sum* approach, is presented in the Figure 7.

- Given: AS_t, $t = 1, \ldots, T$, and m is the size of each vector AS_t
- Sum all anomaly scores AS_t from all T iterations as follows:
- For $i = 1$ to m

  $AS_{FINAL}(i) = \sum_{t=1}^{T} AS_t(i)$
- Return AS_{FINAL}

Figure 7. The Cumulative Sum approach for combining outlier detection scores

This combining method first creates the final outlier score vector AS_{FINAL} by summing all the outlier score vectors AS_t from all T iterations, then sorts the vector AS_{FINAL} and finally identifies the data records with the highest outlier scores as outliers. For example, data record NC_1 in Figure 8 may be ranked as the first outlier by Algorithm 1, ranked as fourth by Algorithm 2, …, and ranked as second by Algorithm t. In the *cumulative sum* approach we sum all the scores that correspond to data record NC_1, namely scores $AS_{1,1}$, $AS_{2,4}$, …, and $AS_{t,2}$, and then sort all data records NC_i, $i = 1, \ldots, m$ according to newly computed score.

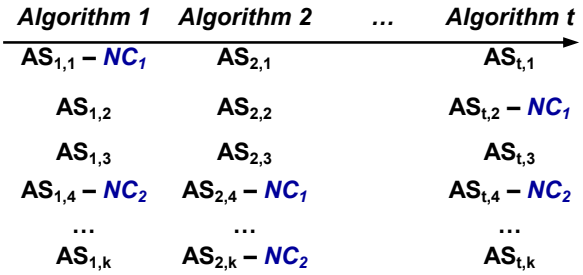

Figure 8. Illustration of the Breadth-First approach for combining outlier detection scores

It is important to note that this method is analogue to the ranking method in the meta search engines where the ranks are summed, but it is more flexible since in the ranking method an outlier detected by a single algorithm may not be detected in the final decision especially if it is ranked low by other detection algorithms. On the other hand, in the *Cumulative Sum* approach, the outlier that is detected by a single algorithm may have very large outlier score, and after all summations are performed may still have sufficiently large final outlier score to be detected. This fact is extremely important in the scenarios where outliers are visible only in a few dimensions, since in that case it is sufficient to select relevant features only in a small number of iterations, compute high outlier scores for these feature subsets and thus cause that these outliers are ranked high in the final score.

5. EXPERIMENTS

Our experiments were performed on several synthetic data and real life data sets summarized in Table 2. In all our experiments, we have assumed that we have information about the normal behavior (class) in the data set. Therefore, in the first training phase, we have applied outlier detection algorithms only to the normal data set (without any outliers) in order to set specific false alarm rates, and in the second (testing) phase, we have applied outlier detection algorithms to test data sets (with all outliers). Using this procedure we can achieve better detection performance that using completely unsupervised approach.

5.1 Experiments on Synthetic Data Sets

Our first synthetic data set (synthetic -1 in Table 2) has 5100 data records, wherein 5000 data records correspond to normal (majority) behavior, and 100 data records represent outliers. The data set has five original (contributing) features that determine which data records are outliers (Figure 9). Normal behavior (blue points in Figure 9) is modeled as a Gaussian distribution of five original contributing features, while the outliers (red crosses in Figure 9) are points that are far from the generated Gaussian distribution. We added 5 noisy features in order to test robustness of "feature bagging" approach to the detection performance.

Our experiments on the synthetic-1 data set were performed using only *LOF* approaches. The computed ROC curves for this scenario for *LOF* approach, Breadth-First and Cumulative Sum approaches employing *LOF* as single outlier detection algorithm are presented in Figure 10.

Table 2. Summary of data sets used in experiments

Dataset	Modifications made in the data set	Size of dataset	Number of features		number of outliers (rare class records)	Percentage of outliers
			continuous	discrete		
Synthetic -1	-	5100	5+5	0	100	1.96%
Synthetic -2	-	5050	8	0	50	0.99%
Satimage	smallest class vs. rest	6435	36	0	626	9.73%
Coil 2000	-	5822	85	0	348	5.98 %
Rooftop	-	17829	9	0	781	4.38 %
Lymphography	merged classes 2&4 vs. rest	148	18	0	6	4.05 %
Mammography	-	11183	6	0	260	2.32 %
KDDCup 1999	U2R vs. normal	60839	34	7	246	0.40 %
Ann-thyroid	class1 vs. class3	3428	6	15	73	2.13%
Ann-thyroid	Class2 vs. class3	3428	6	15	177	5.16%
LED	each class vs. rest	10000	0	7	~1000	~10%
Letter recognition	each class vs. rest	6238	617	0	240	3.85%
Segment	each class vs. rest	2310	19	0	330	14.29%
Shuttle	classes 2, 3, 5, 6 & 7 vs. class 1	14500	9	0	2 - 809	0.014% - 5.58%

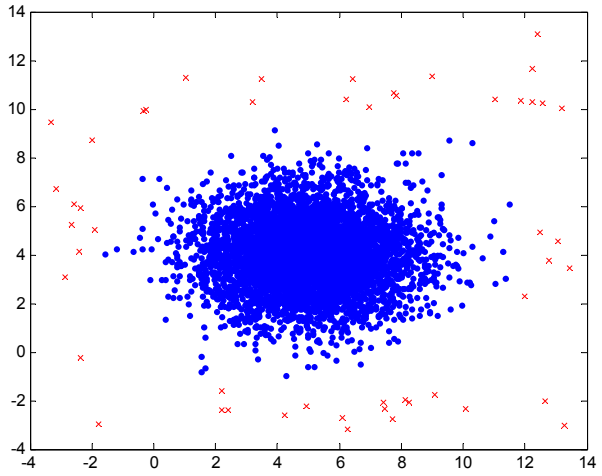

Figure 9. Distribution of two contributing features for the synthetic-1 data set (blue points represent normal behavior, red crosses represent outliers).

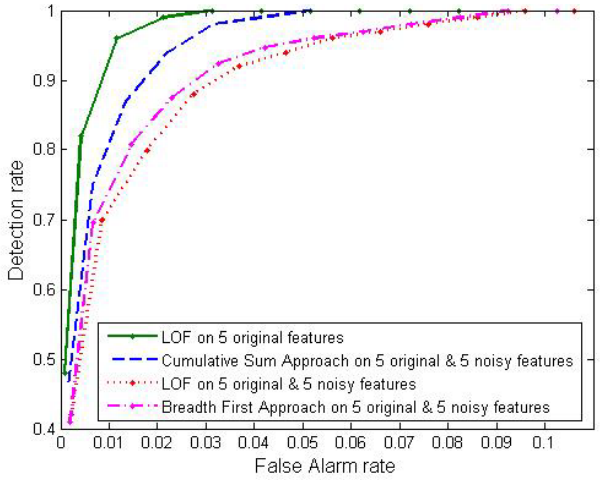

Figure 10. ROC curves for single LOF approach and two combining methods employing LOF approach when applied to the synthetic-1 data set with 5 original and 5 noisy features. The number of combined outlier detection algorithms for all data sets was set to 10. The figures are best viewed in color

Analyzing ROC curves from Figure 10, it can be observed that *LOF* approach applied with five original and five noisy features has much worse ROC curve than *LOF* approach that used only five original features. This was reasonable to assume since density computations in *LOF* approach are significantly influenced by noisy and/or irrelevant features, and thus the *LOF* performance also degrades. However, when the proposed methods for combining outlier detection algorithms are applied on the synthetic-1 data set with five original and five noisy features, it can be observed that they were able to alleviate the effect of noisy features and to outperform single *LOF* approach. Furthermore, the *Cumulative Sum* combining method has very similar ROC curve as the LOF approach only with 5 original contributing features.

On the other hand, the *Breadth First* approach is slightly worse than the *Cumulative Sum*, but still better than *LOF* approach with both contributing and noisy features. That means that if there are irrelevant features in the data sets, combining methods are able to decrease the influence of noisy features regarding the detection performance. Depending on the number of relevant and irrelevant features this decrease can vary. Our earlier experiments also show that this decrease is rather small if the number of irrelevant features significantly outnumbers the number of relevant features. To investigate the influence of the noisy features to the detection performance, we have created two additional synthetic data sets with 10 and 20 noisy features in addition to five contributing features. Instead of ROC curves, for these two data sets we have

reported areas under the curve (AUC), since AUC allows us to easier compare all three scenarios. From Table 3, it can be observed that with increasing number of noisy features, the gap between single LOF and the combining methods is indeed decreasing. That means that the combining methods can alleviate the influence of the noisy features only till a certain level. The AUC of ideal ROC curve corresponds to 1, and it is computed using the trapezoidal rule.

Table 3. AUC (areas under the curve) for single LOF, cumulative sum and the breadth first approaches depending on the number of noisy features in the data set.

Number of noisy features	Single LOF	Cumulative Sum approach	Breadth First approach
5	0.9862	**0.9948**	0.9899
10	0.9745	**9835**	0.9781
20	0.9489	**0.9547**	0.9501

Our second synthetic data set (synthetic-2 data set) has also 5050 data records, wherein 5000 data records correspond to normal (majority) behavior, and 50 data records represent outliers. This data set has 8 features and all 8 features are responsible for determining the outliers, i.e. the data set does not have any noisy features. Like in the synthetic-1 data set (see Figure 9), the normal behavior in this data set corresponds to a Gaussian distribution of eight contributing features, while analogously to the first data set the outliers are data points far from the normal behavior. The computed ROC curves for this data set for *LOF* approach, *Breadth-First* and *Cumulative Sum* approaches are presented in Figure 11. Note that ROC curves for the synthetic-2 data set use different axis scale than ROC curves for the synthetic-1 data set in order to observe true differences.

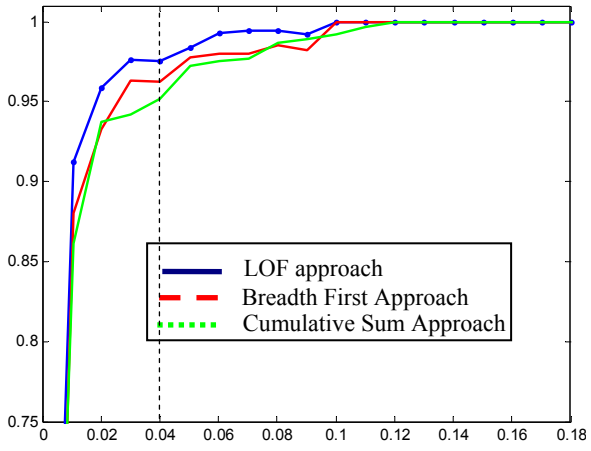

Figure 11. ROC curves for single LOF approach and two combining methods employing LOF approach when applied to the synthetic-2 data set. The number of combined outlier detection algorithms for all data sets was set to 10.

It can be observed from Figure 11 that in the scenario when all features that determine the outliers are important, there is a slight decrease in detection performance of combining methods. However, this decrease is minor (e.g. for false alarm = 4%, detection rate was decreased approximately only 1% for the *breadth first* approach and only 2% for the *cumulative sum* approach. For the false alarm of 10% all three methods achieve 100% detection rate, so the only differences are for the lower false alarm rates. The degradation in performance of the combining methods compared to the single *LOF* approach is understandable since combining methods do not use all the features in any of the iterations, but at the same time due to the nature of the generated data set all the features are important for detecting outliers. However, in real life scenarios, it is hardly the case that all the features are relevant for detecting outliers. To check this claim, we also performed experiments on numerous real life data sets.

5.2 Experiments on Real Life Data Sets

All real life data sets used in our experiments have been used earlier by other researchers for the problem of detecting rare classes [11, 22, 25, 30]. These data sets are summarized in Table 2. Since rare class analysis is conceptually the same problem as the outlier detection, we employed those data sets for the purpose of outlier detection, where we detected rare classes as outliers. In addition to the data sets reported in Table 2, we have also used several data sets from UCI repository [8] that do not directly correspond to the rare class problems or outlier detection problems but can be converted into binary problems by taking one small class (with less than ~10% proportion present in the data set) and remaining data records or the biggest remaining class as a second class. Therefore, we selected the following data sets for the conversion into binary data sets: ann-thyroid, LED, letter recognition, segment, and shuttle. The same procedure was used earlier [18] when experimenting with the rare class learning. Using this technique, we have formed additional 50 data sets. Some of the data sets selected to perform the experiments have both continuous and discrete features. Since *LOF* approach is based on computing distances between data records, measuring a distance between two discrete (categorical) values is not always straightforward. In our implementation, for computing distances between data records that have discrete attributes we have used the concept of inverse document frequency (IDF) already used in outier detection problems [38], where each value of categorical attribute is represented with the inverse frequency of its appearance in the data set.

When performing experiments on COIL 200 [30], mammography [11] and rooftop [22] data sets, we did not change any class distribution. However, in the original lymphography data set [8], there are four classes, but two of them are quite small (2 and 4 data records), so we merged them and considered them as outliers compared to other two large classes (81 and 61 data records). When performing experiments on KDDCup'99 data set, we selected to detect the smallest intrusion class (U2R), which had only 246 instances. Since the outliers are detected as deviations from the normal behavior, we have modified original data set (311029 data records with five classes) such that the new data set contained only the data records from the normal class (60593 data records) and from the U2R class. In such modified data set, we have tried to detect the U2R class using outlier detection algorithms. Finally, for satimage data set we chose the smallest class to represent outliers and collapsed the remaining classes into one class as was done in [11]. This procedure gave us a skewed 2-class dataset, with 5809 majority class examples and 626 minority class examples (outliers). For 50 created binary data sets, we have typically selected one of the smallest classes and then converted either the remaining data records or the biggest remaining class into the majority class. Therefore, for ann-thyroid data set we have detected classes 1 and classes 2 as outliers vs. the class 3

as the normal (majority) class. Similarly, for shuttle data set we have created five data sets by selecting classes 2, 3, 5, 6 and 7 to be detected as outliers compared to the biggest remaining class 1. For other real life data sets (LED, letter recognition, and segment), we have simply selected each of the classes to be detected as outliers and merged all remaining classes to correspond to the normal (majority) class.

For our experiments performed on first six real life data sets from Table 2, the computed ROC curves for *LOF* approach, *Breadth-First* and *Cumulative Sum* approaches are presented in Figure 12. Due to the lack of space the experimental results for remaining 50 created binary data sets were presented using areas under the curves (AUC) (Table 4). The computed AUCs, for chess, LED, letter, segment and shuttle data sets have been averaged over all generated data sets for the original data set. For example, there were 26 binary data sets generated from the original **letter** data set (since there are 26 classes), and AUCs were averaged over all these 26 data sets when reporting experimental results in Table 4.

Table 4. AUC (areas under the curve) for single LOF, cumulative sum and breadth first approaches for 50 real life data sets obtained by converting original data into binary problems.

Data set	Single LOF approach	Cumulative sum approach	Breadth first approach
ann-thyroid class1 vs. class 3	0.869	**0.869**	0.856
ann-thyroid class2 vs. class3	0.761	**0.769**	0.753
LED (average)	0.699	0.695	**0.703**
letter (average)	0.816	**0.820**	0.818
segment (average)	0.820	**0.845**	0.825
shuttle (average)	0.825	**0.839**	0.834

Analyzing Figure 12 and Table 4, it can be observed that both, *Cumulative Sum* and *Breadth First* combining methods outperformed single LOF outlier detection approach on all real life data sets. The improvements in the detection performance were the smallest (approximately 5% in detection rate for chosen false alarm rate) on the COIL 2000 data set (Figure 12a) and on the satimage data set (Figure 12f). This was probably due to the poor performance of individual outlier detection algorithms on these two data sets, so combining their outputs could not lead to significant improvements. When detecting outliers on the rooftop data set (Figure 12b), the improvements were slightly better than for the Coil 2000 data set, but again not large due to weak performance of individual outlier detection algorithms. Nevertheless, the improvements in detection rate for the false alarm rates ranging from 10% to 50% are not small and they vary from 4% to 14%.

The greatest enhancements in outlier detection were achieved for the mammography (Fig. 12d) and KDD Cup'99 (Figure 12e) data sets. For those data sets single outlier detection results had respectively reasonable detection performance, so combining their outputs further improved overall results. However, when performing experiments on lymphography data set (Figure 12c), the detection rate of a single LOF approach was 100% already at 10% false alarm rate, so the combining methods could not improve detection performance very much. In order to illustrate even such a slight improvement of combining methods for this data set, we reported their ROC curves only for small false alarm rates (less than 0.15).

From Table 4, it can be observed that the small improvements were also achieved for those binary data sets that were created by taking one small class as outlier class and remaining data records as a second class. This can be explained by the fact that the remaining classes that were merged together to form a single majority class were quite different, so it was not possible to distinct separated class from the remaining data. It can be also observed that in two data sets when the binary data sets were created by taking the small class as outlier class and the biggest one as the normal class, the improvements of the combining methods were more apparent.

Finally, it can be observed that for all 66 real life data sets used in our experiments and for all values of false alarm rate, both combining methods were consistently better than the single LOF approach. The only exceptions are the lymphography data set, KDDCup'99 data set and certain generated data sets from LED and letter data sets, where for low false alarm rates (less than 0.05 for lymphography data set, less than 0.1 for KDDCup'99 data set and less than 0.2 for data sets created from LED and letter data sets) detection rates of all three approaches were quite similar.

6. CONCLUSIONS

A novel general framework for combining outlier detection algorithms was presented. Experiments on several synthetic and various real life data sets indicate that proposed combining methods can result in much better detection performance than the single outlier detection algorithms. The proposed combining methods successfully utilize benefits from combining multiple outputs and diversifying individual predictions through focusing on smaller feature projections. Data sets used in our experiments contained different percentage of outliers, different sizes and different number of features, thus providing a diverse test bed and showing wide capabilities of the proposed framework. The universal nature of the proposed framework allows that the combining schemes can be applied to any combination of outlier detection algorithms thus enhancing their usefulness in real life applications.

Although performed experiments have provided evidence that the proposed methods can be very successful for the outlier detection task, future work is needed to fully characterize them especially in very large and high dimensional databases, where new algorithms for combining outputs from multiple outlier detection algorithms are worth considering. It would also be interesting to examine the influence of changing the data distributions when detecting outliers in every round of combining methods, employing not only the distance-based but also other types of outlier detection approaches.

7. ACKNOWLEDGMENTS

This work was partially supported by Army High Performance Computing Research Center contract number DAAD19-01-2-0014, by the ARDA Grant AR/F30602-03-C-0243 and by the NSF grant IIS-0308264. The content of the work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred. Access to computing facilities was provided by the AHPCRC and the Minnesota Supercomputing Institute.

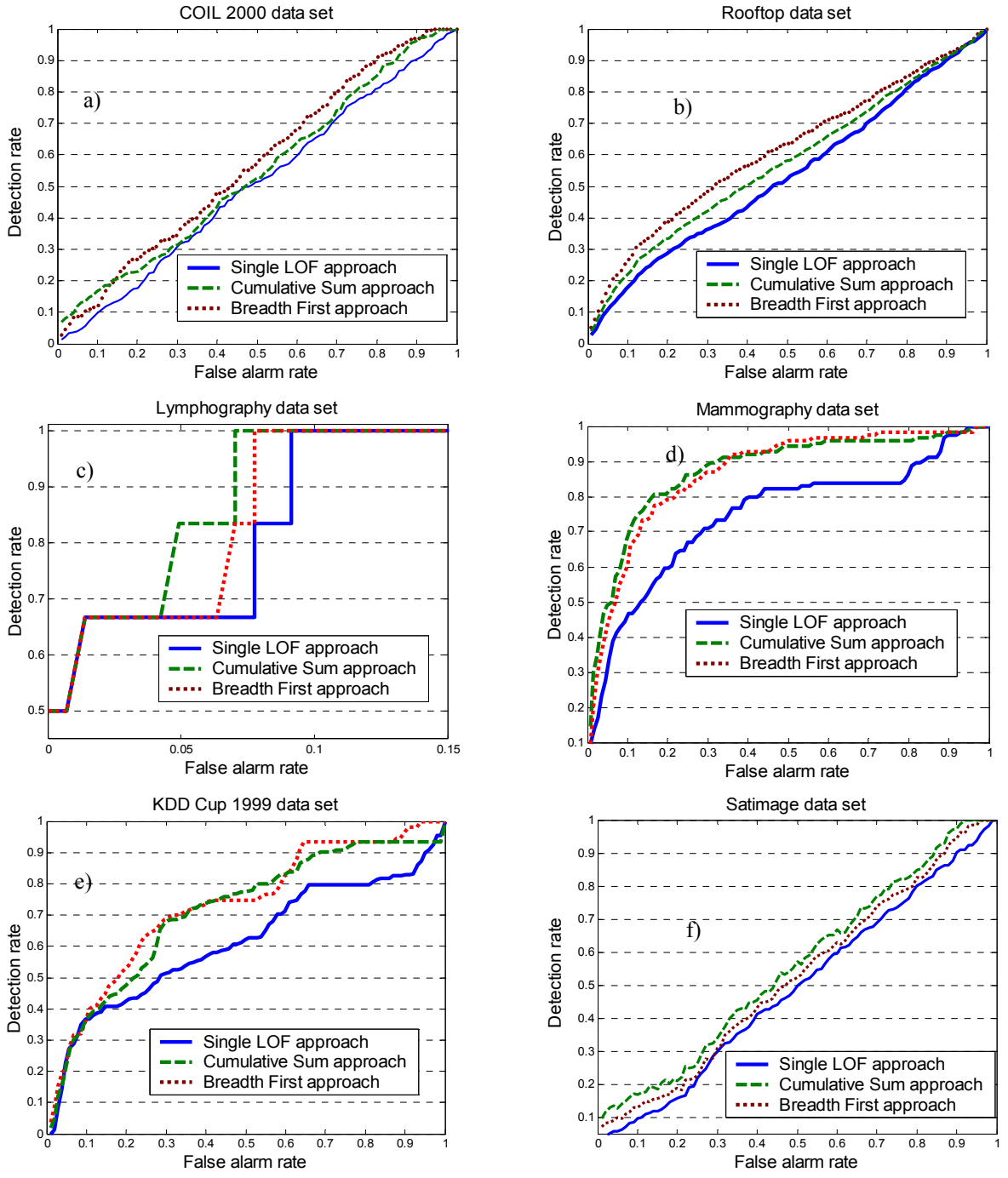

Figure. 12. ROC curves for single *LOF* approach and two combining methods employing *LOF* approach when applied to all five data sets. The number of combined outlier detection algorithms for all data sets was set to 50, except for the mammography data set when this number was 10 due to small number of features (6) in the data set. The figures are best viewed in color.

8. REFERENCES

[1] C. Aggarwal, Re-designing distance functions and distance-based applications for high dimensional data, *ACM SIGMOD Record*, vol. 30, 1, pp. 13 - 18, March 2001.

[2] C. Aggarwal and P. Yu, Finding Generalized Projected Clusters in High Dimensional Spaces, In *Proceedings of the ACM SIGMOD international conference on Management of data*, Dallas, TX, 70-81, 2000.

[3] C.C. Aggarwal, P. Yu, Outlier Detection for High Dimensional Data, In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Santa Barbara, CA, May 2001.

[4] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, Automatic subspace clustering of high dimensional data for

data mining applications, In *Proceedings of the ACM SIGMOD international conference on Management of data*, Seattle, WA, 94-105, June 1998.

[5] V. Barnett and T. Lewis, *Outliers in Statistical Data*. New York, NY, John Wiley and Sons, 1994.

[6] K. Beyer, J. Goldstein, R. Ramakrishnan and U. Shaft, When is nearest neighbor meaningful?, In *Proceedings of the 7th International Conference on Database Theory (ICDT'99)*, Jerusalem, Israel, 217-235, 1999.

[7] N. Billor, A. Hadi and P. Velleman BACON: Blocked Adaptive Computationally-Efficient Outlier Nominators, *Computational Statist & Data Analysis*, vol. 34, pp. 279-298, 2000.

[8] C. Blake,C. Merz, UCI Repository of machine learning databases,www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[9] L. Breiman, Bagging Predictors, *Machine Learning*, vol. 24, 2, pp. 123-140, August 1996.

[10] M.M. Breunig, H.P. Kriegel, R.T. Ng and J. Sander, LOF: Identifying DensityBased Local Outliers, *ACM SIGMOD Conference*, vol. Dallas, TX, May 2000.

[11] N. Chawla, A. Lazarevic, L. Hall,K. Bowyer, SMOTEBoost: Improving the Prediction of Minority Class in Boosting, In *Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003*, Cavtat, Croatia, September 2003.

[12] E. Eskin, Anomaly Detection over Noisy Data using Learned Probability Distributions, In *Proceedings of the International Conference on Machine Learning*, Stanford University, CA, 2000.

[13] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data, in *Applications of Data Mining in Computer Security*, *Advances In Information Security*, S. Jajodia D. Barbara, Ed. Boston: Kluwer, 2002.

[14] Y. Freund, R. Schapire, Experiments with a New Boosting Algorithm, In *Proceedings of the 13th International Conference on Machine Learning*, Bari, Italy, 325-332, July 1996.

[15] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier Detection Using Replicator Neural Networks, In *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science 2454*, Aix-en-Provence, France, 170-180, September 2002.

[16] M. Joshi, R. Agarwal, V. Kumar, PNrule, Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction, In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Santa Barbara, CA, May 2001.

[17] M. Joshi, R. Agarwal and V. Kumar, Predicting Rare Classes: Can Boosting Make Any Weak Learner Strong?, In *Proceedings of the Eight ACM Conference ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, July 2002.

[18] M. Joshi and V. Kumar, CREDOS: Classification using Ripple Down Structure (A Case for Rare Classes), In *Proceedings of the SIAM International Conference on Data Mining*, Lake Buena Vista, FL, April 2004.

[19] E. Knorr and R. Ng, Algorithms for Mining Distance based Outliers in Large Data Sets, In *Proceedings of the Very Large Databases (VLDB) Conference*, New York City, NY, August 1998.

[20] E. Kong and T. Dietterich, Error-Correcting Output Coding Corrects Bias and Variance, In *Proceedings of the 12th International Conference on Machine Learning*, San Francisco, CA, 313-321, 1995.

[21] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava and V. Kumar, A comparative study of anomaly detection schemes in network intrusion detection, In *Proceedings of the Third SIAM International Conference on Data Mining*, San Francisco, CA, May 2003.

[22] M. Maloof, P. Langley, T. Binford, R. Nevatia and S. Sage, Improved Rooftop Detection in Aerial Images with Machine Learning, *Machine Learning*, vol. 53, 1-2, pp. 157 - 191, October-November 2003.

[23] M. Markou and S. Singh, Novelty detection: a review—part 1: statistical approaches, *Signal Processing*, vol. 83, 12, pp. 2481 - 2497, December 2003.

[24] P. McBurney and Y. Ohsawa, *Chance Discovery*, Advanced Information Processing Springer, 2003.

[25] R. Michalski, I. Mozetic, J. Hong and N. Lavrac, The Multi-Purpose Incremental Learning System AQ15 and its Testing Applications to Three Medical Domains, In *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, 1041-1045, 1986.

[26] F. Provost, T. Fawcett, Robust Classification for Imprecise Environments, *Machine Learning*, vol. 42, pp. 203-231, 2001.

[27] S. Ramaswamy, R. Rastogi, K. Shim, Efficient Algorithms for Mining Outliers from Large Data Sets, In *Proceedings of the ACM SIGMOD Conference*, Dallas, TX, May 2000.

[28] A. Strehl, J. Ghosh, Cluster ensembles - a knowledge reuse framework for combining multiple partitions, *Journal of Machine Learning Research*, vol. 3, pp. 583-617, March 2003.

[29] E. Suzuki, J. Zytkow, Unified Algorithm for Undirected Discovery of Exception Rules, In *Proceedings of the Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD2000*, Lyon, France, 169-180, September 13-16, 2000.

[30] P. van der Putten, M. van Someren, CoIL Challenge 2000: The Insurance Company Case, Sentient Machine Research, Amsterdam and Leiden Institute of Advanced Computer Science, Leiden LIACS Technical Report 2000-09, June, 2000.

[31] D. Yu, G. Sheikholeslami and A. Zhang, FindOut: Finding Outliers in Very Large Datasets, *The Knowledge and Information Systems (KAIS) journal*, vol. 4, 4, October 2002.

[32] A. E. Howe, D. Dreilinger, SavvySearch: A meta-search engine that learns which search engines to query, *AI Magazine*, Vol. 18., No. 2, 1997.

[33] S. Lawrence, C. L. Giles, Inquirus, the NECI meta search engine, In *Proceedings of Seventh International World Wide Web Conference*, Brisbane, Australia, 95-105, 1998.

[34] B. U. Oztekin, G. Karypis, V. Kumar, Expert Agreement and Content Based Reranking in a Meta Search Environment using Mearf, In *Proceedings of Eleventh International World Wide Web Conference*, Honolulu, Hawaii, May 2002.

[35] S. D. Bay, M. Schwabacher: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington DC, 29-38, 2003.

[36] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, C. Faloutsos: LOCI: Fast Outlier Detection Using the Local Correlation Integral. In *Proceedings of IEEE International Conference on Data engineering*, Bangalore, India March 2003.

[37] P. Sun, S. Chawla, On Local Spatial Outliers, In *Proceedings of Fourth IEEE International Conference on Data Mining (ICDM'04)*, Brighton, United Kingdom, November 2004.

[38] L. Ertoz, Similarity Measures, *PhD dissertation*, University of Minnesota, in progress, 2005.