Sequential Ensemble Learning for Outlier Detection: A Bias-Variance Perspective

Shebuti Rayana
Stony Brook University
srayana@cs.stonybrook.edu

Wen Zhong
Stony Brook University
wzhong@cs.stonybrook.edu

Leman Akoglu
Carnegie Mellon University
lakoglu@cs.cmu.edu

Abstract—**Ensemble methods for classification have been effectively used for decades, while for outlier detection it has only been studied recently. In this work, we design a new ensemble approach for outlier detection in multi-dimensional point data, which provides improved accuracy by reducing error through both bias and variance by considering outlier detection as a binary classification task with unobserved labels.**

In this paper, we propose a sequential ensemble approach called CARE that employs a two-phase aggregation of the intermediate results in each iteration to reach the final outcome. Unlike existing outlier ensembles, our ensemble incorporates both the parallel and sequential building blocks to reduce bias as well as variance by (i) successively eliminating outliers from the original dataset to build a better data model on which outlierness is estimated (sequentially), and (ii) combining the results from individual base detectors and across iterations (parallelly). Through extensive experiments on 16 real-world datasets mainly from the UCI machine learning repository [1], we show that CARE performs significantly better than or at least similar to the individual baselines as well as the existing state-of-the-art outlier ensembles.

1. Introduction

As outlier detection is a widely researched area, there exist various approaches such as density based [2], [3] and distance based [4], [5] methods, which find unusual points by the distance to their k nearest neighbors (kNNs). However, there exists no known algorithm that could detect all types of outliers that appear in various domains. Hence, ensemble learning for outlier detection has become a popular research area more recently [6], [7], [8], which aims to combine multiple detectors to gain the "strength of many".

In contrast to outlier detection ensembles, classification ensembles have been studied for decades. One can categorize ensembles into two kinds. The first one is the parallel ensemble, where base learners are created independent of each other and their results are combined to get the final outcome; while the second one is the sequential ensemble, where base learners are created over iterations and have dependency among them. Specifically, several outlier ensembles are proposed based on two seminal works of classification ensembles: (i) the parallel ensemble Bagging [9], which creates base components from different subsamples of training datasets parallelly, and (ii) the sequential ensemble AdaBoost [10], which creates base components iteratively. Among those, some try to induce diversity among the base detectors [6], [8], [11], [12], [13], and others selectively combine outcomes from the candidate detectors [7], [14].

Existing outlier ensembles have several limitations, most importantly they avoid discussing the theoretical aspects of outlier detection. Recently, Aggarwal and Sathe [8] argue that although they appear to be very different problems, classification and outlier detection share quite similar theoretical underpinnings in terms of the bias-variance perspective. Specifically, one can consider the outlier detection problem as a binary classification task where the labels are unobserved, the inliers being the majority class and the outliers the minority class, and the error of a detector can be decomposed into bias and variance terms in a similar way. In existing outlier ensembles, various parallel frameworks combining multiple detector outcomes are designed to reduce variance only. Moreover, it remains challenging to reduce bias in a controlled way for outlier detection or remove inaccurate detectors due to the lack of ground truth. There exist a successful heuristic approach which remove outliers in successive iterations [15] to build more robust outlier models iteratively by reducing bias.

In this paper, we study the feasibility of bias-variance reduction under the unsupervised setting, and propose a sequential ensemble model called Cumulative Agreement Rates Ensemble (CARE), to reduce both bias and variance for outlier detection. Specifically, each iteration in the sequential ensemble consists of two aggregation phases: (1) in the first phase, we combine the results of feature-bagged base detectors using weighted aggregation, where weights are estimated in an unsupervised way through the Agreement Rates (AR) method by [16], and (2) in the second phase, the result of the current iteration is aggregated with the combined result from the previous iterations cumulatively. These two phase aggregations in each iteration aim to reduce the variance. Furthermore, we use the combined result from the previous iterations to improve the next iteration by removing the top (i.e., most obvious) outliers and perform a variable probability sampling to create the data model to be used for the next iteration. The removal of top outliers in successive iterations aims to reduce the bias.

To the best of our knowledge, this is the first work focusing on reducing both bias and variance for unsupervised outlier detection. In general, our contributions are following:

- We design CARE, a new approach which incorporates weighted aggregation of feature-bagged base detectors, where weights are estimated in an unsupervised fashion (Section 2.3.1 and 2.3.2).
- We devise a sequential ensemble over the weighted combination, which cumulatively aggregates the results from multiple iterations until a stopping condition is met (Section 2.3.3 and 2.3.4).
- We provide a new sampling approach called Filtered Variable Probability Sampling (FVPS) which utilizes the result from the previous iteration to create the data model for the next iteration (Section 2.3.3).
- CARE is designed to reduce both bias and variance and improves the overall result. Experiments in Section 3 with synthetic datasets support the claim.

We evaluate our method on 16 real-world datasets mostly from the UCI machine learning repository [1]. Our results show that CARE outperforms the baseline detectors in most cases and remains close to them in cases where it falls shorter. We also compare CARE with the existing state-of-the-art outlier ensembles [6], [8], [12]. Similarly, it provides significant improvement when it is the winner, and performs close otherwise (Section 4).

2. Proposed Approach

2.1. Overview

CARE takes the d-dimensional data, a value for k (nearest neighbor count), and a value for $MAXITER$ as input and outputs an outlierness score list **fs** and a rank list **r** (ranked based on most to least outlierness) of n data points. In the experiments we use $k = 5$, which is compatible with the state-of-the-art methods [6], [8]. As for $MAXITER$, we set it to 15, a relatively small value. We assume that our approach improves the base detectors over iterations, and the results are stabilized after only a few iterations.

The main steps of CARE are given in Algorithm 1. Step 3 creates the feature-bagged outlier detectors as base detectors of the ensemble. For the first iteration the sample set S contains the whole data D as shown in step 1. For each base detector, we randomly select $q \in [d/2, d-1]$ features to create b $(= 100)$ different feature-bagged base detectors. Motivated by Platanios *et al.* [16], step 4 calculates the pairwise agreements a_A for all possible pairs of base detectors and step 5 estimates the errors of the individual base detectors in an unsupervised way using a_A. Step 6 calculates weights for the base detectors using their corresponding errors. Step 7 combines the outlierness scores from the different base detectors with weighted aggregation to get final outlierness scores **ws** which are stored in E at step 8. Step 9 calculates the final outlierness scores **fs** by averaging the results of all previous iterations as well as the current iteration. Based on **fs**, step 10 generates the new data sample S (where, $|S| < |D|$) using the FVPS approach (see Section 2.3.3). We repeat steps 3-15 until the stopping condition at step 11 is met or upto the given maximum iteration $MAXITER$. Finally, step 16 generates the ranked list **r** of data points from most to least outlierness.

Unlike existing ensemble techniques, CARE incorporates a two-phase aggregation approach in each iteration; first, it combines the results from the individual base detectors (parallel) and second, it cumulatively aggregates the results from multiple iterations (sequential). The complexity of CARE depends on steps 3, 10, and 16, where step 3 has the highest complexity. As such, the complexity of CARE is $O(bn^2 logn)$.

Algorithm 1: CARE Outlier Detection Ensemble

```
Input: $d$-dimensional Data $D$, NN count $k = 5$,
    $MAXITER = 15$
Output: Score list (fs) and rank list (r) of points
 1: $S = D$ (initially); $E = \emptyset$; $iter = 0$
 2: while $iter \leq MAXITER$ do
 3:     Obtain results from ($b$) feature-bagged base
        detectors $(D, S, k)$ [Section 2.2]
 4:     Calculate pairwise agreement rates $a_A$ for all base
        detector pairs in set $A$
 5:     Estimate detector errors $\mathbf{e}$ $(b \times 1)$ based on $a_A$
        [Section 2.3.1]
 6:     Compute detector weights using estimated errors
        [Section 2.3.2]
 7:     Compute pruned weighted outlierness scores of data
        points to get combined scores ($\mathbf{ws}$) [Section 2.3.2]
 8:     $E = E \cup \mathbf{ws}$
 9:     $\mathbf{fs} = average(E)$
10:     Generate new data sample $S$ from $D$ using
        FVPS (w/o replacement) on $\mathbf{fs}$ [Section 2.3.3]
11:     if stopping condition is TRUE then
12:         $break$ [Section 2.3.4]
13:     end if
14:     $iter = iter + 1$
15: end while
16: $\mathbf{r} = sort(\mathbf{fs})$ (descending order)
```

Next we describe the main components of our proposed CARE in detail. In particular, we describe the base detectors in Section 2.2 and consensus approaches in Section 2.3.

2.2. Base Detectors

In this work, we are interested in *unsupervised* outlier detection approaches that assign outlierness scores to the individual points in the data.

2.2.1. kNN based Outlier Detectors. In our work, we create two versions of CARE: (1) using the distance based approach AvgKNN (average k nearest neighbor distance of individual data point is used as outlierness score), and (2) using the popular density-based approach LOF [2]. We note that CARE is flexible to accommodate any other kNN based outlier detectors.

2.2.2. Feature Bagging. Like classification ensembles, feature bagging can be incorporated in outlier ensembles in order to explore multiple subspaces of the data to induce diverse base detectors and reduce variance. As such, in this work we use feature bagging to create multiple base detectors and combine their results with a goal to improve the outlier detection performance by reducing variance.

2.3. Consensus Approaches

Most of the existing outlier ensembles either combine outcomes of all the base detectors [11], [17], or selectively incorporate accurate base detectors in an unsupervised fashion discarding the poor ones [7], [13]. However, the definition of a poor detector varies across different application domains. Therefore, in this work we go beyond binary selection and estimate weights for individual base detectors to aggregate their results with a weighted combination. In the following two sections, we describe the error as well as weight estimation of the base detectors.

2.3.1. Error Estimation. Motivated by the *unsupervised* Agreement Rates (AR) method of error estimation for multiple classifiers by Platanios *et al.* [16], we estimate the errors (unsupervised) of the base detectors in our work. This estimation is based on the agreement rates for all possible pairs of base detectors in $A : |A| = 2$. Outlier detection can be considered as a binary classification problem with a majority class (inliers $= 0$) and a minority class (outliers $= 1$). However, most existing outlier detection algorithms provide outlierness scores for the data points. In order to adapt the AR approach, $\{0, 1\}$ labels are needed for the data points. We use Cantelli's inequality [18] to estimate a threshold th_i $(i = 1 \ldots b)$ with confidence level at 20% to find a cutoff point between inliers and outliers for each base detector to get binary class labels.

After estimating the class labels, we calculate the agreement rates. As inliers are the majority class and it is likely that most detectors would often agree on a large number of inliers, our main goal is to find agreement based on the outliers detected by the base detectors. Therefore, we take the union of all outliers ($= 1$) across different base detectors to obtain U, which we use to calculate the agreement rates for the detector pairs in A.

In the following sections we denote the base detectors as $f_i \in F$ $(i = 1 \ldots b, |F| = b)$, input data as D, and class labels as Y. The error event E_A of a set of detectors in A is defined as an event when all the detectors make an error:

$$E_A = \bigcap_{i \in A} [f_i(D) \neq Y] \ , \tag{1}$$

where $\bigcap$ denotes set intersection. The error rate of a set of detectors in A is then defined as the probability that all detectors in A make an error together and is denoted as

$$e_A = \mathbb{P}(E_A) \ . \tag{2}$$

The agreement rate of two detectors is the probability that both make an error or neither makes an error. As such, the pairwise agreement rate equation in terms of error rates for the sets in $A : |A| = 2$ can be written as

$$\begin{aligned} a_{\{i,j\}} &= \mathbb{P}(E_{\{i\}} \cap E_{\{j\}}) + \mathbb{P}(\bar{E}_{\{i\}} \cap \bar{E}_{\{j\}}) \\ &= 1 - e_{\{i\}} - e_{\{j\}} + 2e_{\{i,j\}}, \forall \{i,j\} \in A : i \neq j, \end{aligned} \tag{3}$$

where $\bar{\cdot}$ denotes the set complement. On the other hand, the agreement rates for the set of detectors in $A : |A| = 2$ can be directly calculated from the detector output and set U (defined earlier) as follows:

$$a_A = \frac{1}{|U|} \sum_{u=1}^{|U|} \mathbb{I}\{f_i(D_u) = f_j(D_u)\}, \forall \{i,j\} \in A : i \neq j \ . \tag{4}$$

Provided that one can easily compute the pairwise agreement rates $a_{\{i,j\}}$'s, which can be written in terms of the (unknown) individual and pairwise error rates of the detectors, we can cast the error rate estimation as a constrained optimization problem where the agreement equations in (3) form constraints that must be satisfied as follows:

$$\begin{aligned} \textbf{min.} \quad & \sum_{\hat{A}:|\hat{A}| \leq 2} e_{\hat{A}}^2 + \epsilon_{\hat{A}} \\ \textbf{s.t.} \quad & a_A = 1 - e_{\{i\}} - e_{\{j\}} + 2e_{\{i,j\}} \ , \ \forall \{i,j\} \in A \\ & 0 \leq e_{\hat{A}} < 0.5 + \epsilon_{\hat{A}} \ , \\ & 0 \leq \epsilon_{\hat{A}} \end{aligned} \tag{5}$$

where $\hat{A}$ contains individual as well as pairs of detectors (i.e., $\hat{A} = F \cup A$) and $\epsilon_{\hat{A}}$'s denote the slack variables.

In their AR approach, Platanios *et al.* assume that the error rates should be strictly < 0.5. Different from theirs, we allow the error rates to be above 0.5, for which we introduce a slack variable $\epsilon_{\hat{A}} \geq 0$ in constraints $0 \leq e_{\hat{A}} \leq 0.5 + \epsilon_{\hat{A}}$. In real-world settings, it is possible to have poor base detectors having large errors (i.e., worse than random).

Although the above constrained optimization approach estimates error rates of individual as well as of all possible pairs of base detectors, we only utilize the error rates of the individual detectors to calculate their corresponding weights for aggregation, which we describe next.

2.3.2. Weighted Aggregation. In CARE, we propose to use *weighted aggregation* to improve the ensemble as the most common aggregation functions *average* and *maximum* have some limitations. We calculate the weights of the base detectors from their estimated errors (as in Section 2.3.1), such that the weights are positive and inversely proportional to the corresponding errors. Inspired by AdaBoost [10], we calculate weights using the following equation:

$$w_i = \frac{1}{2} \log \left(\frac{2}{e_i} - 1 \right) \ , \ i = 1 \ldots b \tag{6}$$

where $w_i \geq 0$ is the weight of detector i with estimated error $e_i \in [0, 1]$, for $i = 1 \ldots b$. Moreover, as we assume that in real-world settings the base detector pool will have poor (i.e., worse than random) detectors, we discard the detectors with error $e_i \geq 0.5$.

After discarding p detectors with error $e_i \geq 0.5$, we combine the outlierness scores from the base detectors using weighted aggregation. In order to do weighted aggregation, we need to unify the outlierness scores, as different base detectors employ different feature sets, hence provide scores with varying range and scale. To standardize, we use Gaussian Scaling [19] to convert the outlierness scores of AvgKNN or LOF into probability estimates Pr_i $(i = 1 \ldots b - p) \in [0, 1]$. We calculate the final outlierness score $ws(x)$ of a data point x using the weighted average of the probability estimates as follows:

$$ws(x) = \frac{\sum_{i=1}^{b-p} w_i \times Pr_i(x)}{\sum_{i=1}^{b-p} w_i} \qquad (7)$$

Above, $\sum_{i=1}^{b-p} w_i$ is used to normalize the outlierness scores.

Thus far, we described steps 3–7 of Algorithm 1. Next we describe the iterative nature of our sequential ensemble.

2.3.3. Sequential Ensemble. With the weighted aggregation combining multiple feature-bagged base detectors we aim to reduce variance, but our additional goal is to reduce bias. One commonly used bias reduction approach is to remove outliers in successive iterations [20] in order to build more robust outlier models iteratively. This is a type of sequential ensemble. The basic idea is that the outliers interfere with the creation of a model of normal data, and the removal of points with high outlier scores is beneficial for the model in the following iteration.

As such, we adopt a sequential ensemble approach in CARE where we use the result from the previous iteration to improve the next. In particular, we select a subsample S from the original data D (where $|S| < |D|$) to use it as a *new data model based on which we calculate the outlierness scores* for all the data points in D. For example, when we need the average kNN distance of a data point $x \in D$, we calculate the distance to its k-nearest neighbors $N_i \in S$. The goal is to construct S that includes as few of the true outliers as possible, such that it serves as a more reliable data model. To do so, we design a sampling approach which we call Filtered Variable Probability Sampling (FVPS). Following are the steps of the FVPS:

- Discard top T outliers detected in previous step from D, where T is the number of outliers selected using Cantelli's inequality [18] on final outlierness scores **fs** (threshold is selected at 20% confidence level to find the cutoff point between outliers and inliers).
- Select l uniformly at random between $\min\{1, \frac{50}{n}\}$ and $\max\{1, \frac{1000}{n}\}$, where n is the size of D.
- Build sub-sample S (where $|S| = l \times (n - T)$) by sampling from D' (outliers-discarded) based on the probability of the points being normal (i.e., $(1 - \mathbf{fs})$).

In step 1 of FVPS, we obtain D' by filtering the outliers detected in the previous step to reduce bias. Here, we choose confidence level 20% to get a larger T in order to remove as many outliers as possible. Inspired by Aggarwal and Sathe [8], we use variable sampling in step 2. Varying the subsample size at fixed k effectively varies the percentile value of k in the subsample for different iterations, as k is scaled by the inverse of various subsample sizes. For some datasets smaller value of k is better, for others larger is better. Therefore, in CARE we select a small value of k (e.g., 5) and employ variable sampling to incorporate the illusion of using different k in different iterations, which introduces diverse detectors iteratively. After deciding the sample size in step 2, we use probability sampling to create the data model S in step 3. Here, we choose a point from D' to include in S based on its probability of being normal.

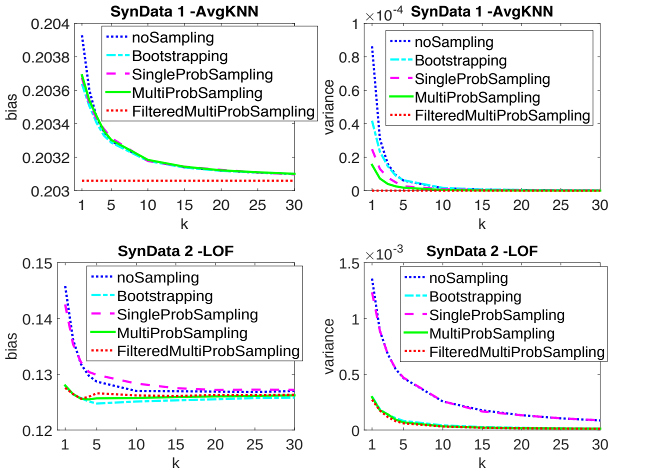

Figure 1. bias (left) and variance (right) vs. k (avg'ed over 10 test datasets) on two synthetic datasets. Notice that our approach (red) w/ probability sampling after top outliers being filtered reduces both bias and variance.

FVPS introduces diverse detectors based on different S in each iteration, hence, we aggregate (e.g. cumulative average) the outlierness scores **ws** over the iterations to compute final scores **fs** to further reduce variance and improve the sequential ensemble (step 9 in Alg. 1). Note that **fs** is also what FVPS uses for discarding outliers and sampling set S.

2.3.4. Stopping Criterion. In CARE, we utilize the pairwise agreement rates a_A between all possible pairs of base detectors to find a stopping point for the sequential ensemble. Experiments reveal a useful strategy: if the distribution of a_A's is skewed towards higher agreement rates, then the error estimates of the base detectors tend to be more accurate. Intuition is, it is unlikely that most pairs would have high agreement and yet agree on the wrong labels. Therefore, we use the area under the curve (auc) of the complementary cumulative distribution function ($ccdf$) of a_A's as the quantitative measure to decide the stopping point. The auc of $ccdf$ is large if the distribution of a_A's is skewed towards higher agreement rates and vice versa. We assume that as CARE sequentially progresses over iterations, the base detectors improve, and hence the auc of $ccdf$ for pairwise agreement rates gets larger. However, if at any iteration $t+1, t \in [0, MAXITER]$, the $auc(t+1)$ falls below the average by more than the standard deviation of $auc(0, \ldots, t+1)$, the sequential ensemble stops and returns the result at iteration t or otherwise iterates until $MAXITER$ and returns the final result.

3. Reducing Bias and Variance with CARE

According to [8], ensembles with feature-bagged base detectors and with variable sampling tend to reduce variance. In this section, we provide quantitative results through experiments on synthetic datasets to show that filtering top T outliers and probability sampling in our sequential ensemble reduce bias along with variance. For each synthetic dataset, we use a data generation model $\mathbb{M}$ to create R training datasets $D_i,\ i = 1 \ldots R$ of size $m = 210$ (200 inliers and 10 outliers) and 10 test datasets $D_j^{Test},\ j = 1 \ldots 10$ of size $n = 1000$ by randomly drawing points from $\mathbb{M}$. Bias and variance of different procedures for different values of k (i.e., # nearest neighbors) for a test data D_j^{Test} are calculated w.r.t. the training data $D_i^{'}$, $i = 1 \dots R$ sampled from D_i as follows:

$$bias = \sqrt{\frac{\sum_{x=1}^{n} (f^*(x) - \overline{f}(x))^2}{n}} \quad (8)$$

$$var = \frac{\sum_{x=1}^{n} \sum_{i=1}^{R} (f(x, D_i^{'}, k) - \overline{f}(x))^2}{n \times R} \quad (9)$$

Here, $\overline{f}(x) = \frac{\sum_{i=1}^{R} f(x, D_i^{'}, k)}{R}$, $f^*(x)$ is the actual label of data point $x \in D_j^{Test}$, and $f(x, D_i^{'}, k)$ is the normalized outlierness score of x w.r.t. sampled training set $D_i^{'}$ for k nearest neighbors. We design five procedures where each procedure has a different approach for sampling $D_i^{'}$. These five different procedures are: (i) *noSampling*: $D_i^{'} = D_i$, (ii) *Bootstrapping*: sampling m times (w/ replacement) from D_i to get $D_i^{'}$, (iii) *SingleProbSampling*: probability sampling on $f(D_i, D_i, k)$ for a single iteration to get $D_i^{'}$, (iv) *MultiProbSampling*: probability sampling on $f(D_i, D_i^{'}, k)$ for multiple (i.e. 10) iterations where $D_i^{'} = D_i$ initially, and (v) *FilteredMultiProbSampling*: filtered (top T outliers removed from D_i) probability sampling on $f(D_i, D_i^{'}, k)$ for multiple iterations (i.e. 10) where $D_i^{'} = D_i$ initially.

In this section, we provide results on only two synthetic datasets (20 dim.) for brevity, where the inliers are drawn from a mixture of Gaussian distributions and outliers are drawn from (1) power law, and (2) uniform distribution. Figure 1 shows bias (left) and variance (right) vs. k, where for the top two plots AvgKNN is used to calculate $f(x, D_i^{'}, k)$, and for the bottom two plots LOF is used. We can see from the figure that FilteredMultiProbSampling (red) reduces both bias and variance more than any other procedures.

4. Experiments

4.1. Datasets

We evaluate CARE on 16 real-world outlier detection datasets (Table 1, all available at http://odds.cs.stonybrook.edu/#table1) mostly from the UCI ML repository [1].

TABLE 1. REAL-WORLD DATASETS USED FOR EVALUATION, WHERE d IS DATA DIMENSIONALITY, AND % INDICATES THE % OF OUTLIERS.

Dataset	**#Pts** n	**Dim.** d	**% Outlier Class**
Lympho	148	18	classes 1,4 (4.1%)
WBC	278	30	21 malignant (5.6%)
Glass	214	9	class 6 (4.2%)
Vowels	1456	12	50 sampled class 1 (3.4%)
Cardio	1831	21	176 pathologic (9.6%)
Thyroid	3772	6	from [21] (2.5%)
Musk	3062	166	classes 213,211 (3.2%)
Optdigits	5216	64	150 sampled digit 0 (3%)
Satimage-2	5803	36	71 sampled class 2 (1.2%)
Letter	1600	32	from [22] (6.25%)
Pima	768	8	pos class (35%)
Satellite	6435	36	3 smallest classes (32%)
Breastw	683	9	malignant class (35%)
Arrhythmia	452	274	classes 3-5,7-9,14,15 (15%)
Ionosphere	351	33	bad class (36%)
Mnist	7603	100	700 digit 6 (9.2%)

4.2. Results

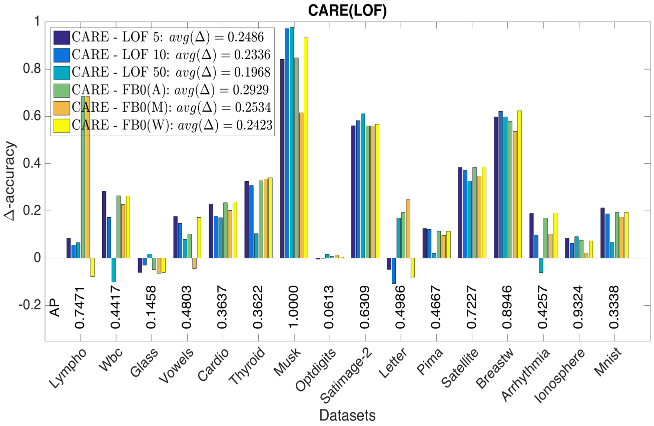

Figure 2. ΔAP (Average Precision) from CARE(LOF) to LOF based baseline approaches on all the datasets. Notice that CARE boosts detection performance significantly for 14/16 datasets over most of the baseline approaches. $avg(\Delta)$ denotes average of ΔAP values across datasets.

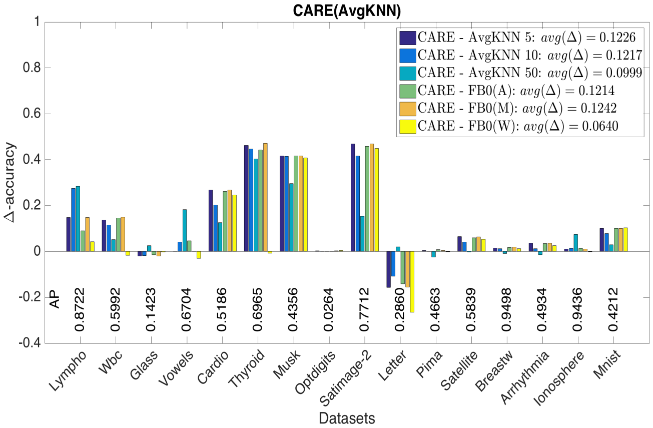

Figure 3. Δ AP values from CARE(AvgKNN) to AvgKNN based baselines. CARE improves over more than half of the baselines on 14/16 datasets.

4.2.1. CARE vs state-of-the-art baselines.

We first compare CARE with simple LOF and AvgKNN based baseline approaches; using $k = \{5, 10, 50\}$, as well as non-sequential feature bagging (FB0) approaches with three types of aggregation; average (A), maximum (M), and weighted (W). Figure 2 shows the Δ Average Precision (AP: area under the precision-recall curve) values from CARE(LOF) to these six baselines all using the LOF algorithm. That is, the bars depict $\text{AP}^{\text{CARE}} - \text{AP}^{\text{baseline}}$. Results show that CARE outperforms all the base detectors on 9/16 datasets, and more than half of them on 14/16 datasets. Negative Δ values are much smaller as compared to positive ones, which indicates that in cases where CARE is not better than the baselines, it remains close. In the legend of the figure, we provide the overall ΔAP values averaged across all the datasets and positive values indicate that CARE performs better than the individual baselines on average. Similarly, Figure 3 contains the ΔAP values from CARE(AvgKNN) to six baselines using AvgKNN based subroutines. Again, the average Δ values (in the legend) across different datasets indicate that CARE outperforms the individual baselines on average. From these two figures we also conclude that CARE(LOF) provides greater improvement over the baselines compared to CARE(AvgKNN). Figure 2 and Figure 3 also contain the absolute AP values (below the bars) of CARE(LOF) and CARE(AvgKNN) respectively for all datasets.

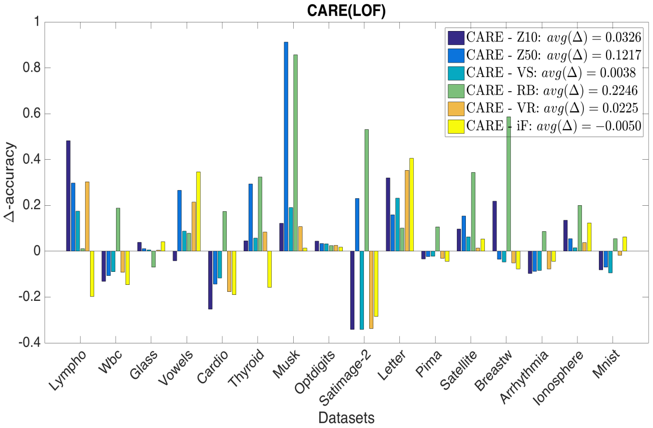

Figure 4. ΔAP from CARE(LOF) to LOF based state-of-the-art ensembles on all the datasets. Notice that CARE outperforms existing ensembles significantly on several datasets and achieves comparable performance otherwise. $avg(\Delta)$'s in the legend denote average of ΔAP across datasets.

4.2.2. CARE vs state-of-the-art ensembles. Next we compare CARE with the existing state-of-the-art outlier ensembles, including Aggarwal and Sathe's variable sampling (VS), rotated bagging (RB), and variable rotated bagging (VR) approaches [8], Zimek *et al.*'s subsampling approach [6], as well as the Isolation Forest (iF) of Liu *et al.* [12]. We employ $b = 100$ base detectors for each of these existing ensembles such that they are comparable with CARE. We present the ΔAP from CARE(LOF) to these six state-of-the-art outlier ensembles using the LOF algorithm (except for iF) in Figure 4. For Zimek's subsampling approach we only present the results for sample sizes 10% and 50% (Z10, Z50). In Figure 4, we can see that CARE mostly improves over Z50 and RB, and remains close to VS. Although iF is little better than CARE(LOF) with $avg(\Delta) = -0.0050$, for some datasets e.g., Vowels and Letter where iF performs poorly with AP values 0.1341 and 0.0929 respectively, CARE(LOF) provides $2.6\times$ improvement with AP value 0.4803 for Vowels, and $4.4\times$ improvement with AP value 0.4986 for Letter. Moreover, we note that the magnitude of positive Δ values are larger than the negative ones on average. This indicates that CARE(LOF) provides major improvement in cases when it is the winner and performs similarly in other cases. CARE(AvgKNN) provides similar results, where the corresponding figure can be found at [23] (as for page limit).

5. Conclusion

In this paper, we proposed CARE, a new sequential ensemble approach for outlier mining with a goal to achieve low detection error through reduced variance and bias. Two main components of CARE are its parallel and sequential building blocks. The former helps reduce variance by a weighted combination of multiple base detectors an the latter is designed to reduce both bias and variance through FVPS and cumulative aggregation. We evaluate our method on 16 real-world datasets. Extensive experiments validate that CARE provides significant improvement over the baseline methods as well as the state-of-the-art outlier ensembles when it is the winner and performs close enough otherwise.

We refer to [23] for further analysis of CARE and additional results. Future work includes speeding up CARE to apply on massive datasets. All source codes of our method and data are shared openly at http://shebuti.com/sequential-ensemble-learning-for-outlier-detection/.

Acknowledgment This research is supported by the ARO Contract W911NF-14-1-0029, DARPA TCP Contract FA8650-15-C-7561, NSF CAREER 1452425, IIS 1408287, and an R&D gift from Northrop Grumman. Any conclusions expressed in this material are of the authors' and do not necessarily reflect the views, expressed or implied, of the funding parties.

References

[1] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *SIGMOD*, 2000, pp. 93–104.

[3] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *ICDE*, 2003.

[4] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Advances in Knowledge Discovery and Data Mining*, 2009, pp. 813–822.

[5] M. E. Otey, A. Ghoting, and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets," *DMKD*, vol. 12, no. 2-3, pp. 203–228, 2006.

[6] A. Zimek, M. Gaudet, R. J. Campello, and J. Sander, "Subsampling for efficient and effective unsupervised outlier detection ensembles," in *KDD*, 2013.

[7] S. Rayana and L. Akoglu, "Less is more: Building selective anomaly ensembles with application to event detection in temporal graphs." *SDM*, vol. 17, 2015.

[8] C. C. Aggarwal and S. Sathe, "Theoretical foundations and algorithms for outlier ensembles." *ACM SIGKDD Explor. News.*, vol. 17, no. 1, pp. 24–47, 2015.

[9] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[10] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[11] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *ACM SIGKDD*, 2005, pp. 157–166.

[12] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *ICDM*. IEEE, 2008, pp. 413–422.

[13] A. Zimek, R. J. Campello, and J. Sander, "Ensembles for unsupervised outlier detection: Challenges and research questions," *SIGKDD Explor. Newsl.*, vol. 15, no. 1, pp. 11–22, 2013.

[14] J. Gao and P.-N. Tan, "Converting output scores from outlier detection algorithms into probability estimates," in *ICDM*, 2006, pp. 212–221.

[15] C. C. Aggarwal, "Outlier ensembles: position paper," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 49–58, 2013.

[16] A. Platanios, A. Blum, and T. M. Mitchell, "Estimating accuracy from unlabeled data," in *In Proceedings of UAI*, 2014.

[17] S. Rayana and L. Akoglu, "An ensemble approach for event detection in dynamic graphs." in *ACM SIGKDD ODD^2 Workshop*, 2014.

[18] G. Grimmett and D. Stirzaker, *Probability and Random Proc.*, 2001.

[19] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Interpreting and unifying outlier scores." in *SDM*, 2011.

[20] C. C. Aggarwal, "Outlier ensembles: position paper." *SIGKDD Explor. Newsl.*, vol. 14, no. 2, pp. 49–58, 2012.

[21] F. Keller, E. Müller, and K. Böhm, "HiCS: high contrast subspaces for density-based outlier ranking," in *ICDE*, 2012, pp. 1037–1048.

[22] B. Micenková, B. McWilliams, and I. Assent, "Learning outlier ensembles: The best of both worlds–supervised and unsupervised," in *ACM SIGKDD ODD^2 Workshop*, 2014.

[23] S. Rayana, W. Zhong, and L. Akoglu, "Sequential ensemble learning for outlier detection: A bias-variance perspective," *arXiv preprint arXiv:1609.05528*, 2016.