Received: 12 April 2019 | Revised: 29 October 2019 | Accepted: 12 December 2019

DOI: 10.4218/etrij.2019-0205

ORIGINAL ARTICLE

ETRI Journal WILEY

Adaptive boosting in ensembles for outlier detection: Base learner selection and fusion via local domain competence

Joash Kiprotich Bii | **Richard Rimiru** | **Ronald Waweru Mwangi**

Department of Computing, School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya

Correspondence
Joash Kiprotich Bii, Department of Computing, School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya.
Email: kbjoash@gmail.com

Unusual data patterns or outliers can be generated because of human errors, incorrect measurements, or malicious activities. Detecting outliers is a difficult task that requires complex ensembles. An ideal outlier detection ensemble should consider the strengths of individual base detectors while carefully combining their outputs to create a strong overall ensemble and achieve unbiased accuracy with minimal variance. Selecting and combining the outputs of dissimilar base learners is a challenging task. This paper proposes a model that utilizes heterogeneous base learners. It adaptively boosts the outcomes of preceding learners in the first phase by assigning weights and identifying high-performing learners based on their local domains, and then carefully fuses their outcomes in the second phase to improve overall accuracy. Experimental results from 10 benchmark datasets are used to train and test the proposed model. To investigate its accuracy in terms of separating outliers from inliers, the proposed model is tested and evaluated using accuracy metrics. The analyzed data are presented as crosstabs and percentages, followed by a descriptive method for synthesis and interpretation.

KEYWORDS

adaptive boosting, base learners, heterogeneous ensembles, outlier detection

1 | INTRODUCTION

The task of differentiating outlier data points from normal points given a particular definition of anomalous behavior is referred to as anomaly or outlier detection [1]. Outlier detection methods are useful in numerous applications, including the detection of defective mechanical parts [2], intrusions in wireless sensor networks [3], credit card fraud [4], and gene separation [5]. Outlier detection techniques have evolved over time to form both supervised and unsupervised techniques. For most outlier datasets, ground truth data are lacking; hence, most outlier detection methods are unsupervised [6]. However, in the unsupervised category, there have been reports of methods producing too many false positives and negatives [7]. To improve the detection accuracy of unsupervised methods, the use of ensembles or groups of learners and detectors to solve the outlier mining problem has become a norm [8–10]. Therefore, many novel outlier ensemble methods have been developed [10–13]. An ensemble is a collection of trained classifier models whose predictions are combined to reach a final decision [14]. Ensemble learning is a machine learning concept in which multiple learners are trained to solve the same problem. In contrast to ordinary machine learning approaches, which attempt to learn a single hypothesis from training data, ensemble methods attempt to construct a set

of hypotheses and combine them for detection [15]. An ensemble contains a number of learners, which are referred to as base learners. The generalization ability of an ensemble is typically much stronger than that of its individual base learners. In essence, ensemble learning is advantageous because it is able to boost weak learners, which are slightly better than random guesses, to create strong learners that can make more accurate predictions [16].

Problem statement: An ideal outlier detection ensemble should consider the strengths of individual base learners while carefully combining their outputs to create a strong learner to achieve unbiased overall detection accuracy with minimal variance. Existing outlier detection ensembles utilize either parallel and/or sequential combination structures to fuse multiple detectors (weak base learners) in an attempt to improve overall detection accuracy by deriving a combined result (majority vote) from the detectors. Parallel combination structures are designed with the goal of reducing variance while serial combination structures are designed with the goal of reducing bias [17]. However, trusting the results from all weak learners may deteriorate the overall performance of an ensemble because some learners can provide inaccurate results depending on data types and the underlying rules of a learner, particularly in the context of outliers with a lack of ground truth data. Outlier detection ensembles that mark safe instances as anomalous or that mark anomalous instances as safe can make systems unsafe, untrustworthy, or redundant. In certain applications, such as medical diagnosis, misclassifications can have catastrophic and irreparable consequences [18]. Therefore, it is necessary to study which detectors should be used as base learners, how they should be combined, and which types of data they are suitable for. To address these problems, this paper presents an adaptive boosting framework for heterogeneous ensembles for outlier detection (ADAHO). The advantage of ADAHO lies in its ability to integrate various high-performance base detectors by utilizing a hybrid structure to minimize bias and variance while improving overall detection accuracy. The main contributions of this work can be summarized as follows:

1. ADAHO selects high-performance or optimal heterogeneous base learners by assessing their capabilities in their local domains or areas of expertise. This is because every outlier detection technique performs best within a specific domain within the entire problem space [19].
2. ADAHO adaptively boosts the outputs of preceding base learners during the first phase of training and carefully combines high-performance base learners in the second phase to generate more accurate predictions compared to simply averaging the outcomes of all base learners.
3. ADAHO is able to fuse heterogeneous detectors that produce different types of outlier outputs at different scales into a unified function for identifying outliers.

The remainder of the paper is organized as follows. A literature review and background information on existing detection ensembles are presented in Sections 2 and 3, respectively. Our design methodology is presented in Section 4. Experiments and empirical results are discussed in Sections 5 and 6, respectively. Finally, in Section 7, conclusions are drawn and directions for future research are discussed.

2 | LITERATURE REVIEW

2.1 | Outlier detection

Outlier detection refers to the task of identifying anomalous patterns in a given dataset according to a particular definition of anomalous behavior [1]. Outlier detection methods are specific solutions to outlier detection problems. The output of an outlier detection method could be a labeled pattern. For example, *outlier* labels or *normal* labels, or scores assigned based on the degree to which a pattern is considered an outlier. Outliers are detected by analyzing events, where each event is designated by a data instance. Each data instance has features (ie, attributes) that describe it [3]. A data instance with a single feature is a univariate instance and a data instance with multiple features is a multivariate instance [1]. Features are crucial for distinguishing normal behavior from anomalous behavior. An outlier was defined by Hawkins as an observation that deviates far enough from other observations to arouse suspicion that it was generated by a different mechanism [20]. In most literature, the authors have described outliers as observations that appear to be inconsistent with the remainder of dataset, which is the main problem associated with handling outliers. Outlier detection methods attempt to solve this problem using various approaches, such as statistical and probabilistic knowledge, distance and similarity-dissimilarity functions, metrics for accuracy when dealing with labeled data, association rules, properties of patterns, and other domain features.

2.2 | Key aspects of the outlier detection problem

In outlier detection, the fundamental problem lies in discerning the unknown data space. A straightforward approach to detecting outliers is to find unusual points by computing a measure of normality or similarity to their neighboring points. However, several factors make this task very challenging. For example, Chandola's survey described seven factors [1]. (a) Defining a normal region (ie, one that covers every possible normal behavior) is very difficult. (b) Normal regions continue shifting in many cases and an existing understanding of a normal region may not hold true in the future. (c) The

margin between a normal region and outlier region is almost always uncertain; this means that an outlying point that lies near the margin could be a normal point or vice versa. (d) The exact definition of an outlier differs for different application areas. (e) Data classes or labels for training or testing are not always available. (f) In cases where outliers are generated by malicious acts, malicious agents can adjust themselves to appear normal, thereby making the task of identifying normal behavior much more difficult. (g) Noise is present in most data, and noise can mimic outliers, making it difficult to separate and eliminate real outliers.

In response to these challenges, outlier detection techniques have attempted to simplify the problem by adding various types of information, including the definition or nature of data and the types of outliers to be detected. Therefore, the application area dictates the methods used to solve the outlier detection problem; this means that there are a number of different methods used in various areas, such as data mining, statistics, and machine learning. For example, by narrowing down Hawkins's concept [20], two major outlier detection techniques can be derived: distance-based techniques and density-based techniques. Distance-based techniques find data points that lie far from their nearest neighbors and density-based techniques find data points that reside in lower-density regions compared to their nearest neighbors. Some researchers have also studied other types of outlier detection algorithms, such as supervised and unsupervised learning algorithms. Supervised learning algorithms detect outliers using labeled data; this means that records are classified as either "*normal*" or "*outlier.*" Unsupervised learning algorithms use unlabeled data; thus, the outliers (and inliers) are unknown [21].

2.3 | Domain significance and data associations

Knowing the associations among data instances is very significant for the outlier detection problem. Therefore, outlier detection methods can be classified based on their use of global data associations, meaning they generate decisions using all data points [10], or local domains, meaning decisions are made based on a few locally selected points [22]. In both cases, applications are domain specific. For example, in high-dimensional data spaces, where outliers lie far from the rest of the data, global associations among all data points are very useful [11]. However, such global associations may underestimate the probability of other outliers lying in local domains [23]. Furthermore, drawing the conclusion that a given data point picked from a random set of samples should lie in a similar region to all other data points could be an inaccurate assumption. Based on this analysis, methods such as the local outlier probability (LoOP) [24], local outlier factor (LOF) [25], and Gloss [23] have been developed. Until the development of recent solutions presented in [13] and [26], the significance of local data regions for classifier selection and fusion in outlier detection problems had rarely been considered. However, it is noteworthy that both of these works utilized homogeneous ensembles. In [13], methods for determining data locality and the regions of competence of base learners were mentioned based on either the *k*-nearest neighbors (*k*-NN) algorithm or data clustering.

2.4 | Value of fusion

In many cases, the nature of data is very unpredictable and base learners can make mistakes for different training instances. It is also the case that some base learners will excel at discerning certain subspaces while others may perform poorly in those subspaces [27], but the overall accuracy of the former learners may be worse than that of the latter. In other words, a base learner can have a domain of expertise in a local domain while performing poorly when considering the entire feature space. To take advantage of each learner's domain of expertise, it is crucial to fuse them to reduce the total error [28]. This strategy is preferable because it helps to overcome the deficiencies of individual detectors while improving overall detection accuracy. The authors of [16] proposed the concept of data locality, which was later improved in [29] to perform dynamic classifier selection in local spaces of training points. Techniques that dynamically select and fuse base classifiers have yielded better results compared to static techniques, such as those that simply take a vote based on all base classifier outputs. Voting with weights and voting without weights are the two most common methods for combining not only homogeneous, but also heterogeneous models [14]. Using an optimal Bayes concept to filter high-performance learners from low-performance learners informed dynamic classifier selection in [30]. This concept was extended into dynamic ensemble selection in [31], which introduced a second-layer fusion strategy using multiple base classifiers. However, in [31], heterogeneous learner fusion was not considered. In [26], the authors asserted that by using multiple base learners, stability can be improved compared to using only the best learner. However, in [13], a single learner or combination of learners was adopted depending on which configuration yielded better results. The concept of a "crowd of experts" is more helpful for diffusing biases compared to a single learner [26]. Experimental results demonstrated that using dynamic ensemble selection or groups of learners is more stable as compared to the case when dynamic classifier selection is used [31]. In [26], additional issues regarding ensemble formation were raised. First, existing fusion strategies are static or lack any learner

selection procedures, which reduces the value of fusion because individual base learners may not fully identify all outliers [32]. Second, the performance of a base learner has always been assessed based on an entire training dataset instead of focusing on the data domains related to a particular testing instance. When all data dimensions are taken into consideration, few outliers can be identified [26]. In most outlier datasets, the majority of the data are normal, meaning most outliers can be identified by looking at small subsets of features [33]. The concept of detector evaluation based on the locality of data instead of global evaluation, which was proposed in [26], is very important. A fusion strategy such as weighted averaging, which utilizes the Pearson correlations between learner outcomes and actual or simulated labels on all training data points [34], is desirable. Additional techniques that support the concept of outlier mining from local data domains include those proposed in [13,23,24]. Outlier detection using heterogeneous algorithms yields heterogeneous scores, which cannot be directly combined because different methods produce different outlier scales [33]. For example, *k*-NN yields scales that are smaller than those of the cumulative neighborhood technique. Different methods also yield different types of vectors, such as binary-valued or real-valued scores, implying that score unifying processes are required for score fusion.

2.5 | Bias-variance tradeoff

Generalization errors in classification can be examined using the bias-variance tradeoff framework [12]. When labels are provided with data samples (ie, supervised learning), this tradeoff is elaborately defined as the quantification of bias and variance. When labels are not provided, unsupervised methods are used; hence, data labels are treated as unknown entities. Outlier detection techniques can be viewed as binary problems with inliers (majority class) and outliers (minority class) by converting detector outlier scores into class labels [10]. Additionally, within the same reference, the error of a base learner can be divided into two parts, namely the reducible error and the irreducible error caused by noise in the data. It is possible to minimize the reducible error to increase detector accuracy by considering two types of errors, namely errors from squared bias and errors from variance. In an effort to reduce both types of errors, compromises (ie, tradeoffs) must be made between the two types. The difference between an expected output and actual unseen value over the training samples defines bias, whereas the difference between the outcome of a detector over a single training set and the expected outcome over the entire data space defines variance. A base learner with low bias is flexible for fitting data properly, but will fit every training set in a different way, resulting in high variance. In contrast, an inflexible base learner with less variance will yield significant bias. It is the objective of an outlier detection ensemble to reduce the errors caused by both bias and variance to reduce the overall effects of generalization errors.

The goal of this study was to improve accuracy as much as possible by reducing both bias and variance and improving the selective outlier ensemble approach by introducing the concept of heterogeneous base detectors into an ensemble. Our approach first identifies high-performance base learners based on their strengths in the domains in close proximity to their training cases (ie, local domains), then utilizes adaptive boosting, where incorrectly classified samples become inputs for subsequent base learners, leading to a final output based on a fusion function.

3 | ENSEMBLE FORMATION

Ensemble formation involves three key steps [10]. The first is the creation step, in which a number of base learners are created. The second is the selection step, in which high-performance learners are picked from the pool of created learners. The third is the fusion step, in which a fusion strategy is used to fuse base model outcomes to create a strong learner. Implementing a good fusion strategy during the third step is very critical because there is a potential threat of weak learners deteriorating the performance of other models in the ensemble, rather than improving their performance [11]. Some outlier ensemble formations are designed in parallel order, such as feature bagging [35] or dynamic combination of detector scores for outlier ensembles (DCSO) [26]. Some are designed with sequential ordering, such as SELECT [11]. Finally, some formations use hybrid ordering, such as BORE [36] and XGBOD [12]. For classification, ensemble methods include bagging [37], boosting [38], and stacking [39]. These ensemble methods can be biased toward some specific characteristics of a dataset because their training procedures utilize only a single base learner.

To introduce diversity and eliminate biases, different types of base learners can be introduced into an ensemble to form a heterogeneous ensemble. This allows different data characteristics to be learned by a diverse set of base learners. However, when all base learner outcomes are fused to generate a final result without considering how well each base learner performs, it can lead to poor overall results or generally weak algorithms [10,11] because poorly performing learners drag down scores of high-performance learners (eg, when averaging is used as a fusion method based on the mean of all scores [40]). Furthermore,

when maximization is used as a fusion strategy, it can yield unstable results [9]. However, it is possible to use either the average of maximization results or the maximum of average results because these two-stage fusion methods can potentially improve an algorithm's overall performance and stability [26].

In an attempt to generate base learners sequentially, the SELECT [11] and CARE [10] algorithms were developed. These methods select high-performance base learners while disregarding low-performance base learners during fusion. SELECT produces a simulated label or ground truth by averaging base learner outlier probabilities, each of which may have biases in different directions, then disregarding low-performance base learners or selecting base learners to retain based on the weighted Pearson correlations between simulated labels and base learner outlier probabilities [11]. This technique yields promising results for temporal graphs and multi-dimensional data. Other approaches, such as the Full ensemble [33], Div-E [41], and ULARA [42], were not designed around the concept of local domain competency or bias variance reduction, unlike SELECT. In the Full ensemble, all base detector outcomes are used, even those with strong biases, which deteriorates the final detector. Although ULARA computes the relative weights of base detectors, it still includes those with strong biases. Div-E hand picks diverse base detectors, including those with strong biases, based on the concept of maximizing diversity, which generally reduces accuracy.

Additional fusion techniques, such as majority voting, weighted majority voting, summation, product, maximum, minimum, fuzzy integral, Dempster-Shafer, and decision templates [28], have also been proposed. For voting, the value with the most votes, highest mean ranking, or highest mean probability is selected [14]. For weighted voting, each base detector is assigned a coefficient or weight according to its performance. It is important to note that most fusion approaches utilize the entire data space for decision making by calculating an overall average of learner scores [26]. We refer to such approaches as (G) fusion techniques. Some of these techniques can be summarized as follows. (i) In global threshold summation (G_TS), all learner outputs below a certain threshold are discarded and the sum of the remaining learner outputs is calculated. (ii) In global maximization (G_Max), the maximum outlier score over all base learners is calculated. (iii) In global averaging (G_AVG), the average of the scores of all base learners is taken as the final outlier score. (iv) In the global average of maximums method (G_AoM), base learners are divided into groups and the maximum score for each subgroup is taken as that subgroup's score. The final score is obtained by calculating the average of all subgroup scores. (v) In the global maximum of averages method (G_MoV), base learners are divided into groups and the average score for each subgroup is taken as the subgroup score. The final score is obtained by choosing the maximum value among all subgroup scores. (vi) In global weighted averaging (G_WA), a simulated training label or ground truth label is generated by calculating the average of all base learner outputs. Each base learner is then assigned a weight by calculating the Pearson correlation between its training output and the simulated label. After the weights are obtained, the final score is calculated as the average of the weights of all base learner scores.

$$\rho(s,t)=\frac{\sum_{i=1}^{n}\left(s_i-\bar{t}\right)\left(t_i-\bar{s}\right)}{\sqrt{\sum_{i=1}^{n}\left(s_i-\bar{t}\right)^2}\sqrt{\sum_{i=1}^{n}\left(t_i-\bar{s}\right)^2}}. \tag{1}$$

The Pearson correlation in (1) measures the similarity between two vectors $\mathbf{s}$ and $\mathbf{t}$ (n denotes vector length), where $\bar{\mathbf{s}}$ and $\bar{\mathbf{t}}$ are the averages of the vectors.

4 | METHODOLOGY

The proposed method utilizes the principles of prediction based on historical outcomes. Our research design uses experiments, derivations, and reasoning as tools for understanding a problem or behavior. In other words, we implemented an experimental research design. This design involves the manipulation of variables and predictions based on past observations and data analysis to identify outliers. This paper presents an adaptive boosting technique for outlier detection called ADAHO. This method combines a set of heterogeneous weak learners to obtain an optimized composite model that provides more accurate and reliable predictions compared to using a single model. It uses weighted versions of a single training dataset instead of random subsampling or bootstrapping. Therefore, the training dataset does not need to be as large as that required by many other methods. Successive weak classifiers are trained using reweighted versions of the training data, where the weights depend on the accuracy of the previous classifiers. Error rates are computed in the context of an instance's local neighbors, rather than a global training set. Testing instances focus on local domains or regions within a training dataset. At every iteration, the training instances are weighted according to the misclassifications (errors) of previous classifiers. This allows weak learners to focus on patterns that were poorly classified by previous classifiers. The final result is a fusion of carefully selected results from individual learners. The main steps in the ADAHO algorithm are presented in Algorithm 1. Step 1: Different detection algorithms (base detectors) are selected with unique measures to score individual vectors. Step 2: A subset of the detector results is selected using a selection strategy. Step 3: The selected results are fused using different fusion methods to create intermediate aggregate outcomes. Step 4: Subsets of the outcomes from Step 3 are selected. Step 5: The selected subset of outcomes is fused.

4.1 | Base learners

Heterogeneous base classifiers are selected because ensemble methods are very effective when base classifiers of dissimilar types are used [10,43]. Based on the differences between classifiers, the unique properties in data can be discovered or learned. When base learners of the same type (homogeneous) are used, the advantages of learner fusion are lost, unless different data subsamples, parameters, or features are used for training each classifier [27,43]. The proposed algorithm uses different base learners (heterogeneous) to construct a group of models to improve efficiency. Distance- and density-based methods are well-known unsupervised methods for outlier detection. Distance-based methods [44,45] attempt to identify global outliers that lie far from the rest of the data. Such methods typically use the *k*-nearest neighbors (*k*NN) algorithm to detect outliers. Density-based methods [25] attempt to find local outliers that lie in less dense regions compared to their *k*NN.

This paper focuses on unsupervised outlier detection. Such techniques assign scores to individual data points, facilitating the ranking of such points based on their probability of being outliers. Motivated by the critical importance of data locality and dynamic learner fusion in DCSO [13,26] and the concept of heterogeneous detector formations in SELECT [11], we adopted two distance-based algorithms to detect global outliers and two density-based algorithms to detect local outliers. (a) The local distance-based outlier factor (LDOF) [45] calculates the distance from a point to its *k*NN and compares that value to the average distances between nearest neighbors. (b) The traditional *k*NN algorithm uses the *k*NN distances of individual instances is used as outlier scores. (c) The LOF [25] calculates the local deviation of a given data point relative to its neighbors. (d) Finally, LoOP [46] is a local-density-based outlier detection method that provides outlier scores in the range of [0,1].

Formally, given a dataset $\mathcal{D}$ of size d features, we divide the dataset into training and testing sets as $X_{\text{train}} \in \mathcal{D}^{n\times d}$ to signify training data with n data points and $X_{\text{test}} \in \mathcal{D}^{m\times d}$ to signify test data with m points.

Next, a pool of heterogeneous base detectors $B = \{b_1, \ldots, b_R\}$ is created by initializing the four weak learning algorithms (LDOF, *k*NN, LOF, and LoOP) using a variety of parameter settings for each. All base detectors are then trained and used to classify outliers in X_{train}. In this phase, an adaptive boosting technique is adopted to create a strong learner from the weak learners that were initialized. This technique samples a training set X_{train} from the initial dataset $\mathcal{D}$ according to a uniform distribution, meaning the initial weight distributions {W} are given a value of $1/N$, where N is the number of training data. These weight distributions are updated adaptively in each iteration based on prediction results. Correctly predicted

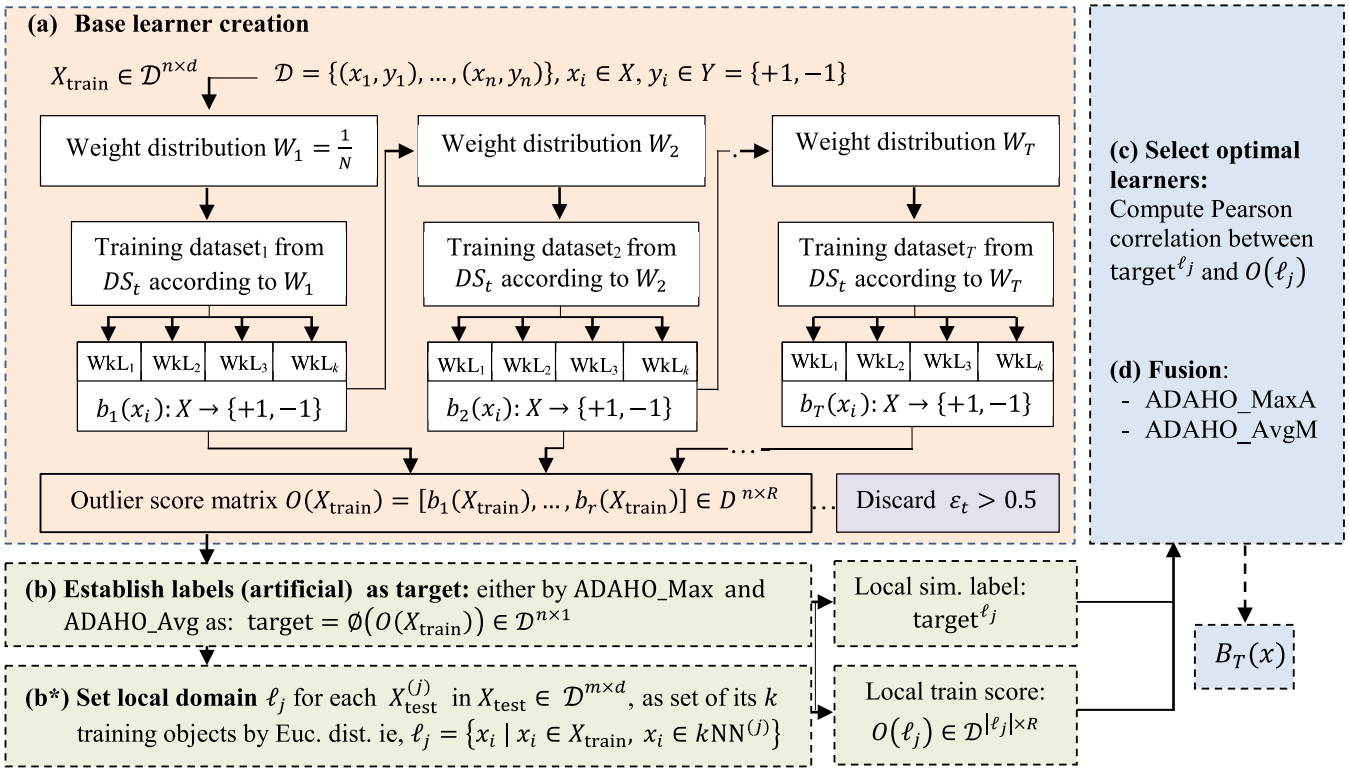

FIGURE 1 ADAHO information flow from leaner creation to outcome fusion. Colors represent different stages. WkL$_k$ are heterogeneous weak learners

ALGORITHM 1: ADAHO

Input: Dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where: $x \in X$ and $y \in Y = \{+1, -1\}$ in the forms of training set $X_{\text{train}} \in D^{n \times d}$ and testing set $X_{\text{test}} \in D^{m \times d}$; a set of heterogeneous weak learning algorithms $\{L\}$; number of iterations T.

Initialize: a set of weights $\{W\}$ by setting $\{W_1(\text{i}) = 1/N\}$ for $(i = 1, \ldots, N)$

For $t = 1, \ldots, T$ rounds;

1. Fit/train X_{train} using a weak learning algorithm $\{L\}$ to get a classifier (base detector) $b_t(x) : X \to \{-1, +1\}$ and calculate the local weighted error of b_t for the i^{th} training instance as
$$\varepsilon_t(i) = \sum_{j \in N(i)} W_t' . I(b_t(x_j) \neq y_j), \quad (2)$$
where
$W_t' = \frac{W_t(j)}{\sum_{l \in N(i)} W_l(l)}$,
where $N(i)$ indicates the neighbourhood of the i^{th} training instance and $I(.)$ is an indicator function such that:
$I_n^t = \begin{cases} (b_t(x_i) \neq y_i) = 1 \\ (b_t(x_i) = y_i) = 0 \end{cases}$.
2. If $\varepsilon_t(i) > 0.5$, set $W_{t+1}(i) = 1/N\ (i = 1, \ldots, N)$ and go to 1.
3. **1st Selection:** Goal: select b_t with the lowest $\varepsilon_t(i)$
4. Estimate the weighted error rate of this base detector for X as
$$\varepsilon_n = \frac{\sum_{j \in N(i)} W_t' . I(b_t(x_j) \neq y_j)}{\sum_{i=1}^{N} W_n^t}. \quad (3)$$
5. Calculate the weighted coefficient of c for detector b_t as
$$\alpha_t(i) = \frac{1}{2} \ln \left\{ \frac{1 - \varepsilon_n(i)}{\varepsilon_n(i)} \right\}. \quad (4)$$
6. Re-weight and update examples (ie, those incorrectly classified receive more weight and those correctly classified receive less weight) as follows:
$$W_{t+1}(i) = \frac{W_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) \text{ if } b_t(x_i) = y_i \\ \exp(\alpha_t) \text{ if } b_t(x_i) \neq y_i \end{cases}, \quad (5)$$
where Z_t is a normalization factor.
7. Iterate: for each test instance, output the outlier score of X_{test}

while (test instance $X_{\text{test}}^{(j)}$ in X_{test})
{ - Set local domain ℓ_j using kNN
- From target, pick elements in ℓ_j to generate local simulated target^{ℓ_j}
while (detector b_r in B)
{ - Obtain outlier scores related to X_{train} in the local domain $b_r(\ell_j)$
- Using (1), assess the local competency of b_r (ie, between target^{ℓ_j} and $b_r(\ell_j)$) }
2nd selection: **if** (ADAHO_Max or ADAHO_Avg)
{Pick $B_r^*(X_{\text{test}}^{(j)}$, where B_r^* has the greatest Pearson correlation with target^{ℓ_j} }
else { Pick subgroup g among similar base learners and add to B_r^*
fusion: if (ADAHO_AvgM or ADAHO_MaxA)
{ Output: $\text{avg}\left(B_r^*(X_{\text{test}}^{(j)})\right)$} **else**
Output: $\max\left(B_r^*(X_{\text{test}}^{(j)})\right)$}}
} end while

Output: Final ensemble (strong) detector B_t.
$$H_t(x) = \text{sign}\left(\sum_{t=1}^{T}\left(\sum_{i=1}^{N} \frac{\alpha_t(i)}{d^2(x, x_i)}\right) b_t(x)\right),$$
where $d^2(x, x_i)$ is the Pearson correlation between the new test instance and i^{th} training instance.

samples from weak learners receive low weights because they are considered to be easy samples. Difficult samples receive higher weights. In this manner, in the next iteration, the learners are able to focus on the difficult samples and attempt to provide better predictions. The weighted error $\varepsilon_t(i)$ of each learner is then calculated, as shown in Algorithm 1, Step 1, (2). The weak learner with $\varepsilon_t > 0.5$ is discarded, as shown in Algorithm 1, Step 2. The learner with the lowest error is selected and its outputs are used for future fusion (Algorithm 1, step 3). In Step 4 in Algorithm 1, every learner's error rate ε_n is estimated using (3) and a combination weight α_t is calculated using (4) for every learner. The weighted coefficients of base learners are used to calculate the outputs. The greater a weight is, the more influence the corresponding learner has on the overall results. Therefore, over T iterations, the ensemble considers l weak learners with different combination weights or weighted coefficients α_t, as shown in Algorithm 1, Step 5. The results of the selected base learners form an outlier score matrix $O\left(X_{\text{train}}\right)$ as follows:

$$O\left(X_{\text{train}}\right) = \left[b_1\left(X_{\text{train}}\right), \ldots, b_r\left(X_{\text{train}}\right)\right] \in D^{n \times R}, \quad (6)$$

where $\boldsymbol{b}_i(.)$ is the score vector from the rth base detector. Each base detector score $b_r\left(X_{\text{train}}\right)$ is normalized using the Z-norm function [9,43] as $z = \left(b_r - \mu\right)/\sigma$, where μ is the mean and σ is the standard deviation. Because LOF outputs normalized scores, its outcomes are not renormalized again in this step. This process is summarized in Figure 1 part A above.

4.2 | Establishing local domains using artificial labels (*target*)

ADAHO assesses the capability or competency of each base detector prior to fusion, but most outlier data have no actual labels or ground truth information. Therefore, $O\left(X_{\text{train}}\right)$ is utilized to generate a simulated ground truth for X_{train} (called target) using two methods: by mean (denoted as ADAHO_Avg, averaging all scores, (7)) and by maximization (denoted as ADAHO_Max, obtaining a maximum score across all detectors, (8)).

$$\text{ADAHO\_Avg} = \frac{\sum_{i=1}^{n} b_i\left(X_{\text{train}}\right)}{r} \in \mathcal{D}^{n \times 1}, \quad (7)$$

$$\text{ADAHO\_Max} = \text{Max}\left\{b_1\left(X_{\text{train}}\right), \ldots, b_r\left(X_{\text{train}}\right)\right\} \in \mathcal{D}^{n \times 1}. \quad (8)$$

Both ADAHO_Max and ADAHO_Avg generate scores for training data, unlike G_AVG and G_MAX, which only generate scores for test data. An aggregation $\emptyset$ representing ADAHO_Avg or ADAHO_Max (ADAHO_Avg ∪ ADAHO_Max) is then performed across all base detectors to yield the

target, which is used for initial detector selection. This process is denoted as

$$\text{target} = \emptyset\left(O\left(X_{\text{train}}\right)\right) \in \mathcal{D}^{n\times 1}. \tag{9}$$

In terms of precision, avg$k$NN (see (10)) yields better results than kNN, so it is used here for selecting the local domain [2]. For a test instance $X_{\text{test}}^{(j)}$, the local domain $\mathscr{C}_j$ is derived as a set of its k-nearest training objects based on Euclidean distance [26] as

$$\mathscr{C}_j = \left\{x_i | x_i \in X_{\text{train}}, x_i \in \text{avg}k\text{NN}^{(j)}\right\}, \tag{10}$$

where $\text{avg}k\text{NN}^{(j)}$ is the average of a set of a $X_{\text{test}}^{(j)}$'s nearest neighbours bound by the ensemble. In an attempt to tame the curse of dimensionality, this technique was adopted by borrowing from the concept of feature bagging [35]. This process is illustrated in Figure 1 part B.

4.3 | Optimal base detector selection

In addition to the first selection based on the error rate of each learner, a second selection is performed to filter out noisy outcomes by obtaining the local simulated label $\text{target}^{\mathscr{C}_j}$ for every test instance, where the values of target with respect to the local domain $\mathscr{C}_j$ are used as follows:

$$\text{target}^{\mathscr{C}_j} = \left\{\text{target}_{x_i} | x_i \in \mathscr{C}_j\right\} \in \mathcal{D}^{|\mathscr{C}_j|\times 1}, \tag{11}$$

where $\left|\mathscr{C}_j\right|$ is the size of $\mathscr{C}_j$. The local training outlier scores $O\left(\mathscr{C}_j\right)$ (Figure 1, part C) can be obtained from the previously generated training score matrix $O\left(X_{\text{train}}\right)$ as follows:

$$O\left(\mathscr{C}_j\right) = \left[b_1\left(\mathscr{C}_j\right), \ldots, b_r\left(\mathscr{C}_j\right)\right] \in \mathcal{D}^{|\mathscr{C}_j|\times R}. \tag{12}$$

To determine the competence of each base detector in a local domain, ADAHO calculates the Pearson correlation similarity measure between the base detector score $b_r\left(X_{\text{train}}^{\mathscr{C}_j}\right)$ and simulated label $\text{target}^{\mathscr{C}_j}$. This method is considered to be more reliable for outlier detection because it uses a similarity measure for evaluating detectors instead of absolute accuracy [41], which is helpful because most outlier datasets are unpredictable and imbalanced. ADAHO then picks the base detector b_r^* with the greatest similarity measure relative to the optimal base detector in a test sample $X_{\text{test}}^{(j)}$ and its outlier score $b_r^*\left(X_{\text{test}}^{(j)}\right)$ is retained as an intermediate result for later use.

4.4 | Fusion of base detector outcomes

Because our base detectors are heterogeneous, their scores may vary in terms of range and interpretation [34]. Therefore, fusing scores directly would be inappropriate, meaning an agreement must be achieved within the ensemble. Based on the literature, agreement methods can be grouped into two main categories: rank-based and score-based methods. In rank-based methods, detector scores are ordered into ranked lists, which makes all detector scores equivalent and allows for easy fusion. Aggregation is then performed to merge all of the scores into a single ranked list. Similarly, score-based methods convert outlier scores into probabilities using either exponential or Gaussian scaling based on posterior probabilities, regularization, or normalization. This makes the outlier scores across different detectors comparable, meaning a final score can be calculated via averaging or maximization. Because rank-based aggregation yields a relatively crude ordering of data instances [11], ADAHO uses such aggregation sparingly for rearranging outlier scores. We adopt a score-based method, such as mixture modeling, which converts outlier scores into probabilities by modeling them as samples from a mixture of exponential (for inliers) and Gaussian (for outliers) distributions and provides binary classes for instances with probabilities greater than 50% receiving

TABLE 1 Benchmark datasets used in our experiments

Dataset [43]	Dim (d)	Inst. (n)	Inliers	Outliers	%
Mnist	100	7603	6903	700	9.207
Letter	32	1600	1500	100	6.250
Cardio	21	1831	1655	176	9.612
Annthyroid	6	7200	6666	534	7.417
Pima	8	768	500	268	34.895
Vowels	12	1456	1406	50	3.434
Thyroid	6	3772	3679	93	2.466
Pendigits	16	6870	6714	156	2.271
Breastw	9	683	444	239	34.990
Stamps	9	340	309	31	9.120

a value of one (ie, outliers) and those with probabilities less than 50% receiving a value of zero (ie, inliers). We then apply ADAHO_MaxA to the top-h performing detectors with respect to their targets or apply ADAHO_AvgM, where the average of h chosen detectors with respect to their targets is taken as a subgroup score (Figure 1 Part D). The final score is obtained by taking the maximum among all subgroup scores. To reduce bias, ADAHO_MaxA and ADAHO_AvgM are used to reduce the risk of picking only the best-performing base detector. The bias in an ensemble is significantly reduced by the fact that only the top-h performing base detectors with respect to their targets are selected and that these h detectors do not increase overall variance.

5 | EXPERIMENTS

5.1 | Experimental goals

The goal of our experiments was to determine which classifiers serve as the best weak learners for constructing the base detectors for the outlier detection ensemble. A weak learner, as defined in the literature, is a learner with an error rate less than 50% ($\varepsilon_t(i) < 0.5$). In this experiment, ADAHO automatically rejected learners with error rates greater than 50% and restarted its loop. Only learners with $\varepsilon_t(i) < 0.5$ were accepted and stored in B for later selection based on their performance in their areas of expertise or local domains. Optimal learners were then carefully selected for fusion in a second phase. This process is followed by the testing of different combinations or fusion strategies (orders) for the formation of an improved outlier detection ensemble. To this end, two different experiments were performed. One compared existing global methods (discussed in Section 3) to the proposed ADAHO variants and the other compared outlier score outputs, which is helpful for identifying improvements.

5.2 | Datasets and assessment metrics

Table 1 lists 10 publicly available outlier detection benchmark datasets from ODDS [43] that were used in our experiments. The dimensions, instances, and numbers of inliers and outliers are shown. In each dataset in the experiment, 70% of the data was used for training and the remaining 30% was set aside for testing. To evaluate performance, the average scores of ten independent trials were used to calculate the area under the receiver-operating-characteristic curve (ROC, AUC) and average precision because these metrics are widely used in outlier research [8,9,11,12,36]. To determine if two results contain significant differences, the non-parametric Friedman test was adopted. This test was

TABLE 2 ROC values (highest values bolded)

Dataset	G_Avg	G_MoV	G_Max	G_AoM	G_Wa	G_Ts	ADAHO_Avg	ADAHO_MaxA	ADAHO_Max	ADAHO_AvgM
Mnist	0.8456	0.8487	0.8248	0.8452	0.8462	0.8171	0.8475	0.8522	0.7711	**0.8532**
Letter	0.7824	0.7930	**0.8333**	0.8209	0.7807	0.7900	0.7717	0.7853	0.8260	0.7766
Cardio	0.8669	0.8764	0.8697	0.8802	0.8681	0.8729	0.8590	0.8807	0.8390	**0.8912**
Annthyroid	0.7583	0.7768	0.7556	0.7538	0.7664	0.7651	0.7444	0.7866	0.7460	0.7886
Pima	0.6929	0.6902	0.6629	0.6755	0.6936	0.6248	0.6958	0.6890	0.6539	**0.6960**
Vowels	0.9164	0.9174	**0.9212**	0.9170	0.9160	0.9198	0.9175	0.9084	0.9137	0.9098
Thyroid	0.9555	0.9546	0.9284	0.9409	0.9564	0.9543	0.9478	0.9523	0.9312	**0.9699**
Pendigits	0.8277	0.8408	0.8387	0.8521	0.8324	0.8447	0.8137	0.8555	0.7137	0.8643
Breastw	0.7261	0.7039	0.6489	0.6737	0.7352	0.6184	0.6452	0.6943	0.7135	**0.7743**
Stamps	0.8845	0.8926	0.8458	0.8662	0.8852	0.8803	0.8787	0.8618	0.8424	**0.8884**

followed by the statistical analysis of results [47]. In every trial, $p < 0.05$ was considered to be statistically relevant.

5.3 | Experimental design

In our first experiment, we compared six techniques considered to be global (Section 3) or baseline based on their testing domains, namely G_Th, G_Avg, G_Max, G_Wa, G_MoV, and G_AoM, with four ADAHO variants (ADAHO_Avg, ADAHO_Max, ADAHO_MaxA, and ADAHO_AvgM, as described in Table 2 and Algorithm 1). For G_AoM and G_MoA, two subgroups (ie, 2g) were created, each containing five unique base detectors to avoid similar results. For ADAHO, k was set to 30 for the sake of consistency in local domain definition. Z-normalization was applied to equalize base detector scores prior to fusion [34]. For consistency during performance assessment, all classifiers used a pool of at least four base detectors. Because the base learners were heterogeneous, outcome diversity was guaranteed.

In our second experiment, evaluation was performed to select optimal base learners based on their competence in their local domains.

Comparisons were performed using Pearson correlation and Euclidean distance. A smaller Euclidean distance between two data points results in a larger weight, while a smaller Pearson correlation results in a smaller weight. Therefore, to align the ranking scales, we performed inverse ranking of Euclidean distances.

6 | RESULTS AND DISCUSSION

6.1 | Performance of ADAHO

In the outlier detection field, the ROC and AUC are important metrics for evaluating detection quality. The results of our first experiment in terms of AUC are listed in Table 2. A total of 10 datasets were considered. The various baseline outcomes of global averaging, maximization, average-of-maximum, maximum-of-average, weighted averaging, and global threshold summation versus the ADAHO variants (ADAHO_Avg, ADAHO_Max, ADAHO_MaxA, and ADAHO_AvgM) are presented. Compared to the global methods, the ADAHO variants provided enhanced performance, particularly ADAHO_AvgM, where notable improvements are highlighted (bold). This is a strong indication that the ADAHO algorithm performs better by using local domain detector selection criteria. It achieves improved results for at least 6 datasets out of 10, which is a very promising outcome. For our second experiment, the average precision values are presented. Again, ADAHO yields improvements

TABLE 3 Mean average precision values (highest values bolded)

Dataset	G_Avg	G_MoV	G_Max	G_AoM	G_Wa	G_Ts	ADAHO_Avg	ADAHO_MaxA	ADAHO_Max	ADAHO_AvgM
Mnist	0.3810	0.3840	0.3800	0.3795	0.3817	0.3735	0.3832	0.3873	0.3252	**0.3878**
Letter	0.2287	0.2372	0.3059	0.2766	0.2271	0.2317	0.2201	0.2295	**0.3245**	0.2306
Cardio	0.3415	0.3607	0.3565	0.3763	0.3434	0.3528	0.3274	0.3859	0.3096	**0.4016**
Annthyroid	0.2200	0.2294	0.2312	**0.2415**	0.2205	0.2176	0.2182	0.2274	0.2248	0.2352
Pima	0.4988	0.4953	0.4712	0.4819	0.4994	0.4498	0.4991	0.4944	0.4615	**0.5041**
Vowels	0.3682	0.3689	0.3659	0.3631	0.3683	0.3682	**0.3812**	0.3577	0.3381	0.3438
Thyroid	0.3944	0.4022	0.2749	0.3387	0.4029	0.3070	0.3443	0.3854	0.2537	**0.4550**
Pendigits	0.0676	0.0722	0.0732	0.0794	0.0679	0.0731	0.0608	0.0792	0.0524	**0.0843**
Breastw	0.4894	0.4748	0.4148	0.4476	0.4984	0.4265	0.4233	0.4665	0.4627	**0.5554**
Stamps	0.3593	0.3559	0.3043	0.3286	0.3605	0.3537	0.3497	0.3209	0.3092	**0.3678**

in at least seven datasets out of ten using the ADAHO_AvgM variant.

Regarding the performances of the selected baseline classifiers, G_Max exhibited improvement for two datasets (*letter* and *vowels*) in the first experiment (high ROC values) while G_AoM exhibited improvement for only one dataset (*annthyroid*) in the second experiment in terms of average precision (Table 3). For the other tests in both experiments, ADAHO exhibited superior results. Furthermore, regarding the global methods, average-of-maximum and maximum-of-average are superior to simple maximization or averaging. This is attributed to the fact that fusion considers a second dimension and yields more stable outcomes, which is also why the ADAHO yield higher scores. These results also support the case presented in [40]. For averaging and maximization in the global domain, local competency assessment provides weaker overall results, but less variance and bias based on global averaging. It is preferable for all learner outcomes to be used in the final fusion process to reduce bias, but this could deteriorate overall ensemble performance because some low-performance (noisy) learner outputs would be included. In contrast, selecting one optimal learner yields a smaller variance drop compared to averaging based on all learner outcomes, which can also lead to deteriorated overall ensemble performance based on strong bias from the single selected learner.

By using the maximum value to generate a simulated label or ground truth, both the global and ADAHO_Max methods become less stable. For example, ADAHO_Max only performed better than global average-of-maximum on the *letter* and *vowels* datasets in our first experiment and only on the *breastw* dataset in our second experiment (Table 3). It is evident that if only a learner's maximum score is used, then the overall ensemble will have high variance. However, applying a second fusion, such as averaging, mitigates this effect. This finding was also reported in [40]. Furthermore, to reduce the variance of an ensemble, ADAHO takes advantage of the G_AoM effect by calculating the mean of outlier scores from subsets of optimal learners, meaning ADAHO_AvgM further reduces the variance of the final ensemble compared to ADAHO_Max.

To reduce ensemble bias, ADAHO_AvgM calculates an average in the second-level fusion, which also improves accuracy. This is evident in the experimental results as ADAHO_AvgM yields higher scores for six datasets in terms of the ROC, namely *mnist, cardio, pima, thyroid, breast*, and *stamps* (Table 2), and for seven datasets in terms of average precision, namely *mnist, cardio, pima, thyroid, pendigits, breastw* and *stamps* (Table 3). Based on our experiments, it is clear that calculating the maximum after the mean does not significantly improve classification results. This is evident for ADAHO_MaxA, which is not improved significantly by either global averaging or maximum-of-averaging. In summary, ADAHO_AvgM is a superior fusion strategy based on its ability to reduce both variance and bias, which answers the question regarding which is the best fusion strategy for outlier detection ensembles.

6.2 | Evaluation of competence

The similarity between base learner outcomes and the simulated ground truth determines the evaluation of competency. However, it is clear that only small differences can be observed when a Friedman test is performed on both Euclidean distances and Person correlations with respect to the ROC and average Precision because performance variance is so minimal (less than 1%). Furthermore, the weights assigned to the base learners did significantly improve model performance because the weight assignment process proved to be computationally expensive.

To reduce this computational cost, additional measures, such as the determination of the value of k for the size of a local domain, could be normalized so long as they do not affect performance, even when outliers are within a dense domain. In this work, when the size of k was set to large values such that the local target and detector outcomes could be normalized, Euclidean distance became the most effective method.

6.3 | Challenges and possible enhancements

First, ADAHO's local domain derivation technique relies on obtaining the closest neighbors to a test instance using a Euclidean distance approach. This approach poses two challenges: (a) significant time is consumed while establishing a test instance's nearest neighbors and (b) performance in multidimensional spaces may be degraded, especially when some features or attributes are inconsequential.

To overcome these challenges, the first problem of local domain definition can be handled using either clustering [13] or prototyping [32], which can significantly reduce the time required for setting the local domain because not all data points are required for these two methods. Regarding the second challenge of handling multidimensional data, additional fusion techniques could be implemented. Techniques such as pruning [11] and model fusion [41] could be more effective for multidimensional data setups compared to pure averaging or maxing and aggregation fusion. Based on these solutions, ADAHO could become more versatile and effective for various separations of data spaces while retaining its advantages of reducing bias and variance.

7 | CONCLUSIONS AND FUTURE WORK

This paper proposed the ADAHO model, which is an adaptive boosting algorithm for the formation of outlier detection ensembles. Four variations that calculate the maximum, average, maximum-of-average, and average-of-maximum base learner scores were presented. ADAHO calculates optimal base learner outcomes with respect to their local domains and fuses these outputs with the goal of reducing both variance and bias while improving overall ensemble accuracy. ADAHO was tested on ten benchmark datasets and it provided improved results compared to existing methods that use global scores. The variant ADAHO_Avg showed the most promising fusion results and was identified as a superior fusion strategy because it yielded improved results for 6 out of 10 datasets in terms of the ROC and seven out of ten datasets in terms of mean average precision. This variant also reduced variance and bias because it averages the variance of base learner outcomes in the second stage. ADAHO utilizes heterogeneous base detectors, meaning diversity among learner outcomes is guaranteed.

In the future, some additional areas that could be investigated include the following. The setting of local domains could be optimized to reduce time consumption when identifying a test instance's nearest neighbors. Additionally, an optimized method for determining the value of k should be considered because dataset features change continuously, meaning k must be dynamic. For unsupervised methods, a simulated ground truth is necessary and we used a simple average in this study. This method could be extended based on clustering. Finally, additional tests and validations should be conducted using the proposed model to establish its practical usage in high-dimensional spaces.

ORCID


Joash Kiprotich Bii https://orcid.org/0000-0003-1969-1019

REFERENCES

1. V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection: a survey*, ACM Comput. Surveys **41** (2009), no. 3, 15:1–58.
2. E. Burnaev, P. Erofeev, and D. Smolyakov, *Model selection for anomaly detection*, in Proc. Int. Conf. Machine Vision (Barcelona, Spain), Oct. 2015, pp. 987525:1–6.
3. M. Xie et al., *Anomaly detection in wireless sensor networks: a survey*, J. Netw. Comput. Applicat. **34** (2011), no. 4, 1302–1325.
4. S. Ghosh and D. L. Reilly, *Credit card fraud detection with a neural-network*, in Proc. 27th Hawaii Int. Conf. Syst. Sci. (Wailea, HI, USA), Jan. 1994, pp. 621–630.
5. Y. Wang and R. Rekaya, *LSOSS: Detection of cancer outlier differential gene expression*, Biomarker Insights **5** (2010), 69–78.
6. C. C. Aggarwal, *Outlier ensembles: position paper*, ACM SIGKDD Explorations **14** (2013), no. 2, 49–58.
7. S. Das et al., *Incorporating expert feedback into active anomaly discovery*, in Proc. IEEE Int. Conf. Data Mining (Barcelona, Spain), Dec. 2016, pp. 853–858.
8. A. Emmott et al., *A meta-analysis of the anomaly detection problem*, arXiv preprint, arXiv:1503.01158, 2015.
9. C. C. Aggarwal and S. Sathe, *Theoretical foundations and algorithms for outlier ensembles*, ACM SIGKDD Explorations Newsletter **17** (2015), no. 1, 24–47.
10. S. Rayana, W. Zhong, and L. Akoglu, *Sequential ensemble learning for outlier detection: A bias-variance perspective*, in Proc. IEEE Int. Conf. Data Mining (Barcelona, Spain), Dec 2016, pp. 1167–1172.
11. S. Rayana and L. Akoglu, *Less is more: Building selective anomaly ensembles*, Trans. Knowledge Discovery Data **10** (2016), no. 4, 1–33.
12. Y. Zhao and M. K. Hryniewicki, *XGBOD: Improving supervised outlier detection with unsupervised representation learning*, in Proc. Int. Joint Conf. Neural Netw. (Rio de Janeiro, Brazil), July 2018, pp. 1–8.
13. B. Wang and Z. Mao, *Outlier detection based on a dynamic ensemble model: Applied to process monitoring*, Inf. Fusion **51** (2019), 244–258.
14. M. N. Haque et al., *Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data, Classification*, PLoS ONE **11** (2016), no. 1, e0146116:1–e146128.
15. Z. Zhi-Hua, *Ensemble Learning*, National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China, 2012.
16. T. K. Ho, J. J. Hull, and S. N. Srihari, *Decision combination in multiple classifier systems*, IEEE Trans. Pattern Analysis Machine Intell. **16** (1994), no. 1, 66–75.
17. R. Shebuti and A. Leman, *An ensemble approach for event detection in dynamic graphs*, in KDD ODD^2 Workshop (New York, USA) 2014.
18. D. Khullar, A. K. Jha, and A. B. Jena, *Reducing diagnostic errors-why now*, New England. J. Med **373** (2015), 2491–2493.
19. N. Isadora et al., *Ensemble learning method for outlier detection and its application to astronomical light curves*, The Astronomical J. **152** (2016), no. 3, 71:1–13.
20. D. Hawkins, *Identification of Outliers*, Chapman and Hall, London, 1980.
21. M. Milou, *Outlier detection in datasets with mixed-attributes*, Vrije Universiteit Amsterdam (Sept. 2015), Thesis [online: https://beta.vu.nl/nl/Images/stageverslag-meltzer_tcm235-614959.pdf, last accessed March 31, 2019].
22. E. Schubert, A. Zimek, and H. P. Kriegel, *Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection*, Data Mining Knowledge Discovery **28** (2014), no. 1, 190–237.
23. B. Van Stein, M. Van Leeuwen, and T. Back, *Local subspace-based outlier detection using global neighborhoods (Gloss)*, in Proc. IEEE Int. Conf. Big Data (Washington, DC, USA) Dec. 2016, pp. 1136–1142.
24. H.-P. Kriegel et al., *LoOP: Local outlier probabilities*, in Proc. ACM Conf. Inf. Knowledge Mananag. (Hong Kong, China) Nov. 2009, pp. 1649–1652.
25. M. Breuning et al., *LOF: Identifying density based local outliers*, in Proc. ACM SIGMOD Int. Conf. Manag. Data (Dallas, TX, USA), 2000, pp. 93–104.
26. Y. Zhao and M. K. Hryniewicki, *DCSO: Dynamic combination of detector scores for outlier*, Ensembles (2018), https://doi.org/10.13140/RG.2.2.11165.77288.

27. A. S. Britto, R. Sabourin, and L. E. S. Oliveira, *Dynamic selection of classifiers - a comprehensive review*, Pattern Recogn. **47** (2014), no. 11, 3665–3680.
28. R. Polikar, *Ensemble based systems in decision making*, IEEE Circuits Syst. Mag. **6** (2006), no. 3, 21–45.
29. K. Woods, W. P. Kegelmeyer, and K. Bowyer, *Combination of multiple classifiers using local accuracy estimates*, IEEE Trans. Pattern Analysis Machine Intell. **19** (1997), no. 4, 405–410.
30. G. Giacinto and F. Roli, *A theoretical framework for dynamic classifier selection*, in Proc. Int. conf. Pattern, Recogn. (Barcelona, Spain), Sept. 2000, pp. 8–11.
31. A. H. R. Ko, R. Sabourin, and A. S. Britto, *From dynamic classifier selection to dynamic ensemble selection*, Pattern Recogn. **41** (2008), no. 5, 1735–1748.
32. R. M. O. Cruz, R. Sabourin, and G. D. Cavalcanti, *Dynamic classifier selection: recent advances and perspectives*, Inf. Fusion **41** (2018), 195–216.
33. H. V. Nguyen, H. H. Ang, and V. Gopalkrishnan, *Mining outliers with ensemble of heterogeneous detectors on random subspaces*, in Proc. Int. Conf. Database Syst. Adv. Appicat. (Tsukuba, Japan), 2010, pp. 368–383.
34. A. Zimek, R. J. G. B. Campello, and J. Sander, *Ensembles for unsupervised outlier detection: Challenges and research questions*, ACM SIGKDD Explorations **15** (2014), no. 1, 11–22.
35. A. Lazarevic and V. Kumar, *Feature bagging for outlier detection*, in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery data Mining (Chicago, IL, USA), Aug. 2005, pp. 157–166.
36. B. Micenkova, B. McWilliams, and I. Assent, *Learning representations for outlier detection on a budget (BORE)*, arXiv Preprint: 1507.08104, 2015.
37. L. Breiman, *Bagging predictors*, Machine Learn. **24** (1996), no. 2, 123–140.
38. Y. Freund and R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, J. Comput. Syst. Sci. **55** (1997), no. 1, 119–139.
39. D. H. Wolpert, *Stacked generalization*, Neural Netw. **5** (1992), no. 2, 241–259.
40. C. C. Aggarwal and S. Sathe, *Outlier ensembles: An introduction*, Springer, New York, NY, USA, 2017.
41. E. Schubert et al., *On evaluation of outlier rankings and outlier scores*, in Proc. SIAM Int. Conf. Data Mining (Anaheim, CA, USA), 2012, pp. 1047–1058.
42. A. Klementiev, D. Roth, and K. Small, *An unsupervised learning algorithm for rank aggregation*, in Proc. Eur. Conf. Machine Learn. (Warsaw, Poland), 2007, pp. 616–623.
43. ODDS Library, 2016, [Available from: http://odds.cs.stonybrook.edu. last accessed December 2019].
44. E. M. Knorr and R. T. Ng, *Algorithms for mining distance-based outliers in large dataset*, in Proc. Int. Conf. Very Large Data Bases (New York, NY, USA), 1998, pp. 392–403.
45. J. Zhang, *Towards outlier detection for high-dimensional data streams using projected outlier analysis strategy*, Dissertation, Dalhousie University, Halifax, Canada, 2008.
46. H. P. Kriegel et al., *Outlier detection in axis-parallel subspaces of high dimensional data*, in Proc. Pacific-Asia Conf. Knowledge Discovery Data Mining (Bangkok, Thailand), 2009, pp. 831–838.
47. J. Demsar, *Statistical comparisons of classifiers over multiple data sets*, J. Machine Learn. Research **7** (2006), 1–30.

AUTHOR BIOGRAPHIES

Joash Kiprotich Bii received his BS degree in computer technology from the School of Computing and Information Technology, Jomo Kenyatta University of Agriculture & Technology, Nairobi, Kenya in 2007. He graduated with an MS degree in software engineering in 2013 and is completing his PhD in computer science at the same University. Currently, he is a lecturer of computer science at the School of Science and Informatics at Maasai Mara University, Narok, Kenya. His main research interests are deep learning systems, data science, software engineering, mobile computing, and machine learning.

Richard Rimiru received his BS degree in statistics and computer science from the Jomo Kenyatta University of Agriculture & Technology (JKUAT), Nairobi, Kenya. He received his MS degree in computer science from the National University of Science & Technology, Zimbabwe and his PhD in computer science and technology from Central South University, Changsha, China. He is serving as a senior lecturer at the School of Computing & Information Technology at JKUAT, Kenya. His research interests include database systems, artificial intelligence, knowledge-based systems, computer networks, and mobile computing. He has authored and co-authored over thirty papers in peer-reviewed journals.

Ronald Waweru Mwangi is an associate professor of computer science at the School of Computing and Information Technology of the Jomo Kenyatta University of Agriculture and Technology, Kenya. He graduated with a BS degree in mathematics from Kenyatta University in 1989, an MS degree in operations research and cybernetics from the Shanghai University of Science and Technology, China, in 1995, and a PhD in information systems engineering from Hokkaido University, Japan in 2004. His research interests include machine learning, simulation and modeling, and software engineering.