Electronic Letters on Computer Vision and Image Analysis 24(2):338-354, 2025

Multi-Biased Models for Hyperspectral Anomaly Detection: A Paradigm to Improve Performance and Generalizability

Bradley J. Wheeler* and Hassan A. Karimi*

* *Geoinformatics Laboratory, School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15213, USA*

Received 14th August 2025; accepted 2nd December 2025

Abstract

Hyperspectral anomaly detection (HAD) poses a significant challenge as it requires modeling data with hundreds of measurements for each location in space. Many algorithms have been proposed to address problems in HAD, but most originate from one of several biases assumed of the data. This means that disparities between bias and variance can be observed among the algorithms in terms of their performance on individual datasets and more broadly across a diverse range of datasets. Ensemble learning enables the amalgamation of information across multiple biases to attenuate the trade-offs between bias and variance, improving individual dataset performance and generalizability across multiple datasets. Despite some work employing ensemble learning in HAD, amalgamating diverse HAD biases is an unexplored research direction. It is not clear whether amalgamating HAD biases improves performance, or what types and quantities of biases should be included and to what extent. To this end, this study employs 5 different ensembling methods to amalgamate 5 unique HAD biases to identify anomalies in 14 diverse datasets. The ensembling methods implemented consider equal, unequal, sparse, minimal, and mixed contributions among the biases. Results indicate that multi-biased models outperform single-biased models across all 14 datasets. In 12 of the 14 datasets peak performance was achieved by excluding or minimizing contribution to some of the biases, indicating that a mixture of sparse and minimal contributions was optimal. The results furnish empirical evidence as to the efficacy of multi-biased models to improve individual and generalized dataset performance, hence attenuating the bias-variance trade-off observed in the single-biased models. The results additionally provide direction for the most effective amalgamation strategies to construct optimal multi-biased HAD models.

Key Words: Keywords: Ensemble learning, hyperspectral anomaly detection, hyperspectral imaging, multi-biased modeling, regularization.

1 Introduction

Hyperspectral anomaly detection (HAD) requires modeling data that often contains hundreds of measurements for a single point in space. Common benchmark dataset collections, such as the ABU corpus [1], contain over 200 bands for most of the datasets in the collection. The same datasets cover a 100x100 pixel space, resulting

Correspondence to: bjw71@pitt.edu

Recommended for acceptance by Angel D. Sappa
https://doi.org/10.5565/rev/elcvia.2318
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

in 10,000 samples for each dataset. This combination creates a large, high-dimensional, and complex problem, posing substantial computational and analytical challenges.

Multiple algorithms have been developed to address different aspects of the complexities inherent in HAD problems. Common algorithmic strategies regularly identified in literature include reconstructive, clustering, machine learning, and statistical methods [2], [3], [4], [5]. These correspond to algorithms that implement biases based on reconstructive, density, partitioning, hyperspherical, and Gaussian principles. Therefore, while many HAD algorithms are regularly proposed, the biases they implement to model the data are limited in comparison.

As different biases are prone to modeling different distributions of data better, and the high complexity of hyperspectral data enables a large variety of distributions to be represented, performance of HAD algorithms vary accordingly as datasets change. Two surveys that perform impartial evaluations of multiple HAD algorithms highlight common disparities that result from how different biases model diverse datasets. In the first survey [2], it is observed that each dataset in the evaluation had a different algorithm (among the 12 algorithms evaluated) that best identified anomalies within it, signifying that as distributions in the data change, different biases were needed to best model the data. In the second survey [6], one algorithm had top performances on 6 of the 17 datasets but an algorithm that only had the top performance on one dataset had the best average performance across all 17 datasets, indicating disparities between algorithmic biases that performed well compared to those that generalized well. Different biases can cause the algorithms to perform better in different cases or generalize better across larger sets of datasets at the sacrifice of top individual performances. No singular bias, or algorithm implementing one of those biases, performs unanimously better when paired against a sufficient quantity of competing biases on a sufficient quantity of datasets.

Ensemble learning has shown to be an effective approach for addressing the performance disparities associated with individual biases. By amalgamating multiple biases, ensemble learning can capitalize on the strengths of each bias while mitigating their respective weaknesses, resulting in improved performance compared to using a singular bias. This method has been shown to attenuate the impact of the bias-variance trade-off, leading to enhanced performance not only on individual datasets but generally across a broader range of datasets as well.

While ensemble learning has been applied in HAD studies, its use has not been extensively explored, and the integration of ensemble methods with diverse biases is minimal. The most common use of ensemble learning is in isolation forests [7], [8], [9] where multiple decision trees (i.e., partition-biased models) are used to model the data. Similar single-biased ensemble studies have been conducted using autoencoders [10], collaboration representation-based detectors [11], [12], and one-class support vector machines [13], [14]. In each case the same algorithm, and consequently bias, was used for all models in the ensemble. Some studies extract features from the data to perform a feature fusion-based amalgamation of data [15], [16], [17], but these studies utilize a singular algorithm and consequently a singular bias as well. The only instance of ensembling HAD algorithms with multiple diverse biases identified was a study that employed a hard voting ensemble approach with four uniquely biased HAD algorithms [18] on four hyperspectral datasets. In their study all four biases were given equal contribution towards the final outcomes and performance was measured on each of the four selected datasets individually.

Given the state of existing literature, a gap exists in determining whether ensembling HAD algorithms with diverse biases can attenuate the bias-variance trade-off observed in singular biased HAD algorithm performances. Open research questions are: Do multi-biases models outperform singular-biased models? What types and quantities of biases should be included in the multi-biased models and to what extent? Addressing these questions is important as their answers could provide avenues for increasing performance on both individual datasets and broadly across diverse sets of algorithms, leading to new HAD algorithms that are more generalizable to multiple problems in HAD.

To address this gap and subsequent research questions, this study conducted experiments to furnish empirical evidence as to whether peak HAD performance is achieved by amalgamating multiple biases or by using HAD algorithms that implement a singular bias. Experiments were conducted employing ensembling algorithms that

incorporate bagging, stacking, and stacking combined with lasso, ridge, and elastic net regularization techniques to amalgamate the biases. These experiments were used to determine whether equal, unequal, sparse, minimal, or mixed levels of contributions from each bias perform best, respectively. Five HAD algorithms with 5 unique biases were utilized within the ensembling algorithms to identify anomalies across 14 diverse datasets. The contributions of this study are twofold: (a) it furnishes empirical evidence as to the efficacy of multi-biased models to improve individual and generalized dataset performance, hence attenuating the bias-variance trade-off observed in the single-biased models and (b) it provides direction for the most effective amalgamation strategies to construct optimal multi-biased HAD models.

The rest of this paper is organized as follows. Section 2 describes in detail the methodology employed in this study. Section 3 describes the overall experimental design including dataset selection and chosen parameters for the algorithms within the methodology. Section 4 provides the results and an analysis of the experiments. Section 5 discusses the broader implications of the experiments. Section 6 provides a summary of the study and concluding remarks.

2 Methods

2.1 Overview

The ensemble learning algorithms employed in this study all function in a similar manner (Fig. 1). Multiple bootstraps of the dataset are first generated. Then each bootstrap is processed by a set of HAD algorithms with diverse biases to identify how much each sample from the bootstrap deviates from the others. The HAD algorithms employed in this step are referred to as the base learners. Once complete, the deviation measurements from the base learners for each bootstrap are standardized between zero and one and then aggregated through the ensemble to produce a final deviation prediction for each spectral signature in the dataset. This is compared against the ground truth information provided with the dataset and evaluated using the AUROC.

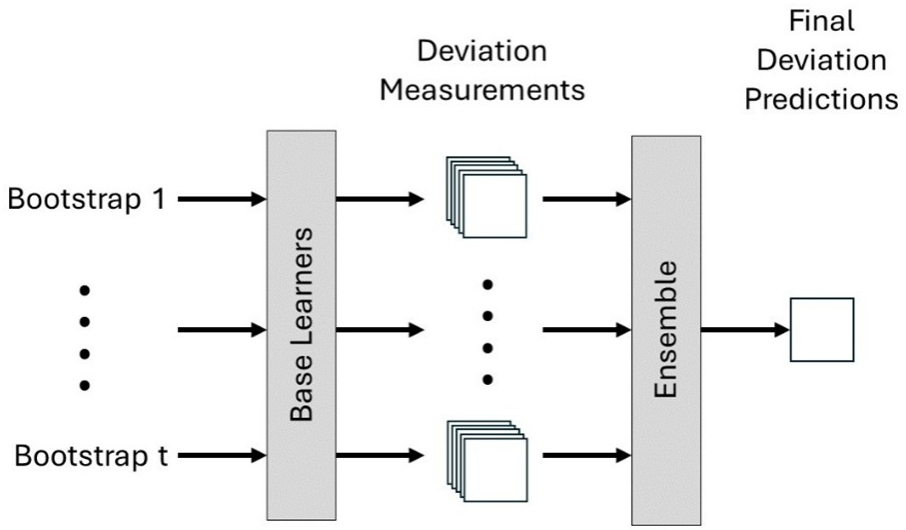


Figure 1: General framework for ensemble learning in HAD.

2.2 Base Learners

The base leaners chosen for this study were specifically selected to introduce a diverse range of modeling biases. A major benefit of ensemble learning is that it enhances the benefits and mitigates the limitations of the selected base learners. Therefore, by selecting HAD algorithms with diverse biases these benefits can be

maximized and produce more robust ensemble models. HAD algorithms with reconstructive [19], density [20], partitioning [21], hyperspherical [22], and Gaussian [23] biases were selected. These biases correspond to the deep learning [24], [25], [26], clustering [26], [27], [28], machine learning [29],[30], and statical [31], [32], [33] classes of HAD algorithms commonly identified in literature [2], [3], [4], [5].

2.3 Bagging Ensemble Learning

The bagging-based ensemble learning approach computes the average of the deviations generated by the base learners for each spectral signature. This assumes an equal contribution of each base learner in the algorithm. L defines a set of b base learners (Eq. (1)). A bootstrap x of the dataset can be passed into the base learners L to produce ρ which contains the deviation measurements produced by each of the b base learner for a bootstrap of height m and length n (Eq. (2)). This process can then be repeated t times with different bootstraps of the dataset to produce a set P of base learner deviation measurements (i.e., ρ) for each bootstrap (Eq. (3)). Lastly, the average of the deviation measurements across all bootstraps and base learners in P can be computed to produce final deviation predictions $\hat{\mathrm{y}}$ for each spectral signature in the dataset (Eq. (4)). Equation 4 produces an m by n matrix that can be compared to the ground truth using the AUROC.

$$L = [\ell_1, \ldots, \ell_b] \tag{1}$$

$$\rho = L(x) \in \mathbb{R}^{b \cdot m \cdot n} \tag{2}$$

$$P = [\rho_1, \ldots, \rho_t] \in \mathbb{R}^{t \cdot b \cdot m \cdot n} \tag{3}$$

$$\hat{y} = \frac{1}{bt} \sum_{j=1}^{t} \sum_{k=1}^{b} P_{jk} \in \mathbb{R}^{m \cdot n} \tag{4}$$

2.4 Stacking Ensemble Learning

Stacking ensemble learning for HAD learns a vector β of size b that can be broadcast onto each ρ in P to produce the final deviation predictions. β is used to control the magnitude of contribution each base learner provides to the final deviation predictions, relaxing the assumption of equal contribution made by the bagging ensemble learner. The algorithm is composed of a forward pass and back propagation step. In the forward pass the n by m deviation prediction matrix $\hat{y}$ is computed for a bootstrap x by multiplying each element of β with its corresponding base learner output matrix in ρ and summing the resulting products (Eq (5)). In the back propagation step the final deviation predictions are evaluated against the bootstrap outcomes y as the mean squared error between their corresponding spectral signature outcomes (Eq. (6)). β can then be updated to produce better predictions by subtracting the product of γ and the gradient of Equation 6 with respect to β from β (Eq. (7)). This process is repeated until Equation 6 converges. Then Equation 5 can be used to produce the final deviation predictions which can be measured against the ground truth provided with the dataset using the AUROC.

$$\hat{y} = \beta \cdot P_i = \sum_{j=1}^{b} \beta_j \cdot P_{ij} \in \mathbb{R}^{m \cdot n} \tag{5}$$

$$\mathcal{L}(\hat{y}_i, y_i) = \frac{1}{mn} \sum_{j=1}^{m} \sum_{k=1}^{n} \left(y_{ijk - (\beta \cdot P_i)_{jk}} \right)^2 \tag{6}$$

$$\beta = \beta - \frac{2\gamma}{mn} \sum_{j=1}^{m} \sum_{k=1}^{n} P_{ijk} \left(y_{ijk} - (\beta \cdot P_i)_{jk} \right) \tag{7}$$

2.5 Stacking with Regularization

2.5.1 Lasso Regularization

Lasso regularization for stacking ensemble learning is accomplished in HAD by penalizing the loss from Equation 6 for using too many base learners in the deviation predictions. This is accomplished by adding the sum of the absolute values of each element in the β vector to the loss, tempered by a λ parameter (Eq. (8)). This will promote sparsity in β and help prevent utilizing unneeded base learns. The added regularization term will change how β is updated in the back propagation step by adding the product of the λ parameter and the sign of each element in β to the vector that is subtracted from β (Eq. (9)). Like the stacking ensemble learning algorithm without regularization, deviation predictions are made using Equation 5, and learning continues until the loss converges. Then β can be used to make the final deviation predictions and the results can be evaluated using the AUROC.

$$\mathcal{L}(\hat{y}_i, y_i) = \frac{1}{mn} \sum_{j=1}^{m} \sum_{k=1}^{n} \left(y_{ijk} - (\beta \cdot P_i)_{jk} \right)^2 + \lambda \sum_{j=1}^{b} |\beta_j| \tag{8}$$

$$\beta = \beta - \frac{2\gamma}{mn} \sum_{j=1}^{m} \sum_{k=1}^{n} P_{ijk} \left(y_{ijk} - (\beta \cdot P_i)_{jk} \right) + \lambda \cdot sign(\beta) \tag{9}$$

2.5.2 Ridge Regularization

Ridge regularization for stacking ensemble learning is accomplished in HAD by penalizing the loss from Equation 6 for relying too heavily on any of the base learners. This is accomplished by adding the sum of the squares of each element in the β vector to the loss, tempered once again by a λ parameter (Eq. (10)). This encourages the learning of smaller, or potentially minimal, values for the elements in β. This helps reduce model complexity, prevent overfitting, and improve model generalizability. This also changes how β is updated in the back propagation step by adding twice the product of λ and β to the vector that is subtracted from β (Eq. (11)). Learning and evaluation follows the same procedure as Lasso regularization.

$$\mathcal{L}(\hat{y}_i, y_i) = \frac{1}{mn} \sum_{j=1}^{m} \sum_{k=1}^{n} \left(y_{ijk} - (\beta \cdot P_i)_{jk} \right)^2 + \lambda \sum_{j=1}^{b} \beta_j^2 \tag{10}$$

$$\beta = \beta - \frac{2\gamma}{mn} \sum_{j=1}^{m} \sum_{k=1}^{n} P_{ijk} \left(y_{ijk} - (\beta \cdot P_i)_{jk} \right) + 2\lambda\beta \tag{11}$$

2.5.3 Elastic Net Regularization

Elastic Net attempts to combine the benefits of both lasso and ridge regularization for HAD by balancing penalization for using too many base learners and relying too heavily on any of the base learners. This is accomplished through adding both the ridge and lasso regularizations to Equation 6 and tempering them independently with different λ parameters (Eq. (12)). This can help enforce sparsity in β while additionally encouraging the remaining values in β to be smaller, which in addition to helping prevent overfitting, may also help with handling correlated base learners. The resulting β update step (Eq. (13)) will include the gradients for both the lasso and ridge regression as defined in Equation 9 and Equation 11, respectively. Learning and evaluation follows the same procedure outlined in the other stacking ensemble learning algorithms.

$$\mathcal{L}(\hat{y}_i, y_i) = \frac{1}{mn}\sum_{j=1}^{m}\sum_{k=1}^{n}\left(y_{ijk} - (\beta \cdot P_i)_{jk}\right)^2 + \lambda_1 \sum_{j=1}^{b}|\beta_j| + \lambda_2 \sum_{j=1}^{b}\beta_j^2 \tag{12}$$

$$\beta = \beta - \frac{2\gamma}{mn}\sum_{j=1}^{m}\sum_{k=1}^{n} P_{ijk}\left(y_{ijk} - (\beta \cdot P_i)_{jk}\right) + \lambda_1 \cdot sign(\beta) + 2\lambda_2\beta \tag{13}$$

3 Experiments

3.1 Datasets

In this study the ABU airport, ABU beach, ABU urban, and HYDICE urban [1], [34] datasets were selected for experimentation (Table 1). This comprised a total of 14 datasets. Each value was vector normalized as the quotient of the value divided by the square root of the sum of the squares of all the values in the band from which the value originated. This was helpful to expedite convergence for the algorithms employed in this study.

Table 1: Dataset Characteristics

Dataset	Location	Anomalies	Number Bands	Spatial Resolution	Percent Anomalies
Airport 1	Los Angeles	airplanes	205	7.1	1.44
Airport 2	Los Angeles	airplanes	205	7.1	0.87
Airport 3	Los Angeles	airplanes	205	7.1	1.7
Airport 4	Gulfport	airplanes	191	3.4	0.6
Beach 1	Cat Island	ship	188	17.2	0.084
Beach 2	San Diego	planes	193	7.5	2.02
Beach 3	Bay Champagne	ship	188	4.4	0.11
Beach 4	Pavia	vehicles	102	1.3	0.302
Urban 1	Texas Coast	buildings	204	17.2	0.67
Urban 2	Texas Coast	buildings	207	17.2	1.55
Urban 3	Gainesville	mixed	191	3.5	0.52
Urban 4	Los Angeles	buildings	205	7.1	2.72
Urban 5	Los Angeles	houses	205	7.1	2.32
HYDICE	urban area	roofs and vehicles	175	1.56	0.26

3.2 Stacking Ensemble Learning Parameters

In the stacking-based ensemble learning algorithms different ranges of values were evaluated for the learning and regularization parameters. The learning parameters included γ (learning rate) and learning convergence. For γ, values of 1, 0.1, 0.01, and 0.001 were evaluated. In all experiments 0.1 was found to best accelerate convergence without suffering from non-convergence issues. Learning convergence was evaluated for when the difference between the current loss and the prior loss was less than 1x10-5, 1x10-8, and 1x10-10. A convergence of 1x10-8 was found to be ample in all cases. Therefore, a learning rate of 0.1 and a learning convergence of 1x10-8 was used in all the experiments.

The regularization parameters included the λ (regularization strength) parameter in the lasso, ridge, and elastic net algorithms. Values of 0.001, 0.01, 0.1, and 1 were evaluated for the λ parameter in each algorithm. The λ parameter that produced the best results for each algorithm was chosen by determining which value of λ enabled each algorithm to produce higher AUROC scores than the base learners most often. The regularization

algorithm and λ parameter that produced the top AUROC score on each dataset individually was also recorded and is reported in the "max reg" outcomes in the results section (section 4).

4 Results

Table 2 provides the AUROC scores the algorithms employed in this study achieved on each of the 14 datasets. The mean, standard deviation, and median ARUOC scores each algorithm achieved across all 14 datasets is additionally included at the bottom of the table. The base algorithm scores are provided first and are followed by the ensemble learning algorithms without and with regularization. The last column provides the maximum score achieved by any regularization method with any λ parameter on the given dataset (max reg).

Some important observations can be made from analyzing the base learner performances in Table 2 in isolation. It is noteworthy that the reconstruction-biased base learner (Recon) had the greatest number of top performances among the base learners with four top performances. This was followed by the hyperspherical-biased base learner (Hsphere) with three top performances and the highest AUROC score among the base learners on any of the datasets. However, the reconstruction- and hyperspherical-biased base learners also had the greatest number of bottom performances. Conversely, the Gaussian- and partition-biased base learners had fewer top performances and lower AUROC scores on individual datasets. However, they never had the bottom performance on any dataset, and they had the top mean, standard deviation, and median AUROC scores among the base learners. Therefore, performance between the base algorithms is divided between those that perform well and generalize well. This highlights the disparity present between the biases as a result of the bias-variance trade-off (as discussed in the introduction) and provides evidence that amalgamating multiple biases may improve performance.

The bagging ensemble learning algorithm performances in Table 2 provide empirical evidence into how ensemble learning can improve upon the base learner performances. Using the central tendencies of the base learners, the bagging algorithm was able to achieve a higher mean and median AUROC than any of the base learners, providing better overall performance across the datasets. It additionally had a lower standard deviation in AUROC scores than all base learners except the partition-biased base learner. This is expected as the central tendency of the base learner outcomes would mitigate the weaknesses of each algorithm individually, raising overall performance across more diverse sets of data. The bagging algorithm was additionally able to outperform all base learners on the Airport 3, Urban 1, and Urban 5 datasets, showcasing its ability to improve individual performance as well in some cases. Top performances of the bagging algorithm are more limited than other ensemble learning approaches as employing all the base learner outputs uniformly make it more sensitive to disparate base learner outputs. This is especially true in cases where the base learner outputs may produce skewed or multimodal distributions of results. Therefore, the major benefits of employing bagging-based ensemble learning are its improved generalizability and simplicity in implementation.

The stacking-based ensemble learning algorithm performances in Table 2 demonstrate how breaking the assumption of equal base learner contribution improves performance and generalizability over bagging-based approaches. The stacking algorithm was able to improve mean, standard deviation, and median AUROC performances over all the base learners and the bagging algorithm. The stacking algorithm was additionally able to outperform all the base learners on 9 of the 14 datasets. By relaxing the assumption of equal base learner contribution, the strengths of the base learners were enhanced in addition to their weaknesses being mitigated. This made the stacking algorithm much less sensitive to disparate, or skewed or multimodal distributions of, base learner results. This made the employment of the base learners towards the final deviation predictions more adaptable to each dataset individually. These factors all led to the enhanced performance of the stacking algorithm.

The datasets where a base learner performed better than the stacking algorithm provide insight into a major difference between the stacking and bagging algorithms. A base learner performed better than the stacking algorithm on the Airport 2, Airport 3, Beach 4, Urban 2, and Urban 4 datasets. The Airport 3, Beach 4, Urban 2, and Urban 4 datasets have the lowest standard deviations in AUROC scores across all the base learning

Table 2: AUROC scores for all algorithms in the study, and the mean, standard deviation, and median AUROC scores for each algorithm across all 14 datasets.

Dataset	Recon	Partition	Density	Gauss	Hsphere	Bagging	Stacking	Lasso	Ridge	Elastic Net	Max Reg
Airport 1	0.93686	0.90583	0.76542	0.89242	0.84239	0.89789	0.94969	0.94253	0.94536	0.94553	0.94616
Airport 2	0.95473	0.9497	0.90052	0.94353	0.90831	0.9538	0.92921	0.94461	0.94969	0.95584	0.96337
Airport 3	0.91564	0.94155	0.92567	0.94047	0.92621	0.94644	0.91744	0.93548	0.93931	0.93923	0.95055
Airport 4	0.94857	0.97191	0.97456	0.97879	0.9388	0.9786	0.98568	0.9861	0.98444	0.9849	0.9861
Beach 1	0.98188	0.96125	0.96198	0.96764	0.91084	0.96221	0.99329	0.99278	0.99372	0.99298	0.99372
Beach 2	0.9039	0.92135	0.90811	0.92225	0.90126	0.90866	0.94578	0.95371	0.95301	0.95203	0.95371
Beach 3	0.9956	0.99751	0.93795	0.99763	0.95545	0.99031	0.9977	0.99741	0.99775	0.99811	0.99819
Beach 4	0.95582	0.98281	0.99189	0.98482	0.9944	0.99264	0.98827	0.98892	0.98852	0.9892	0.99446
Urban 1	0.95693	0.99166	0.99392	0.99314	0.98943	0.9941	0.99434	0.99409	0.99463	0.99472	0.99483
Urban 2	0.98733	0.99643	0.99785	0.99693	0.99935	0.99919	0.99919	0.99926	0.99901	0.99906	0.99937
Urban 3	0.89629	0.96651	0.97157	0.96853	0.95951	0.97029	0.97175	0.96641	0.96726	0.96669	0.97235
Urban 4	0.9673	0.98562	0.99019	0.97997	0.99414	0.99228	0.98738	0.9894	0.99078	0.9913	0.9944
Urban 5	0.87484	0.96085	0.96213	0.95399	0.96051	0.96963	0.96218	0.96261	0.96329	0.96331	0.9691
HYDICE	0.99714	0.99028	0.93189	0.99535	0.85995	0.98291	0.99851	0.99852	0.99872	0.99872	0.99884
Mean	0.94806	0.96595	0.94383	0.96539	0.93861	0.96707	0.97289	0.97513	0.97611	0.97654	n/a
Stdev	0.03831	0.02828	0.06061	0.0315	0.04989	0.0314	0.02749	0.02332	0.02213	0.02178	n/a
Median	0.95528	0.96921	0.96205	0.97366	0.94713	0.97444	0.98653	0.98751	0.98648	0.98705	n/a

algorithms among all the datasets, i.e., the base algorithms were most in agreement with each other on these datasets. Therefore, the stacking algorithm was not able to as readily learn which algorithm should contribute more because of the homogeneity of the base learner outputs. This is in sharp contrast to the bagging algorithm which relied on agreement among the base learners to perform better.

Figure 2 provides the magnitudes of contribution assigned to the base learners for the final deviation predictions learned by the stacking algorithm. For 11 of the 14 datasets the Gaussian-biased algorithm had the strongest contribution towards the predictions. Therefore, the stacking algorithm learned to use a stronger generalizing base learner for its predictions in most cases. This is expected, as it enables the algorithm to begin with a more robust model and then fine-tune the model with more specific information from higher performing models specific to each dataset. It is noteworthy that some base learners were assigned an inverse contribution for some datasets. This would be anticipated, as it would help with noise removal from the predicted deviations of the stacking algorithm where the base learners are not in agreement for the deviation of specific spectral signatures.

Another important consideration are base learners that were assigned small levels of contributions. Four cases exist where the magnitude of contribution assigned to one of the base learners was less than one percent of the total range of contribution magnitudes assigned to the other base learners for a given dataset. These were the density-, Gaussian-, and hyperspherical-biased base learners for the HYDICE, Airport 1, and Beach 3 and HYDICE, respectively. Therefore, few cases exist where base learners were assigned negligible levels of contribution towards the final deviation predictions and highlights the stacking algorithm's tendency to consider all base learners as being important towards the final deviation predictions.

The AUROC scores in Table 2 for the lasso, ridge, and elastic net regularization methods demonstrate that regularization further improves results over stacking alone. All three regularization methods were able to improve upon the mean and standard deviation in AUROC scores over the stacking algorithm. The lasso, ridge, and elastic net algorithms were additionally able to outperform the stacking algorithm on 9, 10, and 9 of the 14 datasets, respectively. A Value of 0.001 was found to be the most effective for the λ parameter across all three models. These results have two important implications. First, the results provide empirical evidence that further relaxing the contributions of base learners to encourage sparsity and minimal contributions can improve HAD performance. Second, the low value of λ indicates that employing regularization is a delicate balance and diminishing or removing the contribution of too many base learners is detrimental to performance. However, this delicate balance is to be expected as the prior results from the bagging and stacking algorithms indicate that contributions from multiple base learners tend to outperform any one base learner individually.

The elastic net algorithm proved to be the best performer and generalizer among the regularization algorithms (Table 2). The elastic net algorithm was the top performing algorithm among the regularization algorithms on 7 of the 14 datasets. The lasso and ridge algorithms had 3 and 4 top performances, respectively. The elastic net algorithm additionally had the best mean and standard deviation in AUROC score of any of the algorithms evaluated in this study and outperformed the base learners on 9 of the 14 datasets. As such, the elastic net model was able to improve individual and generalized performance, and therefore attenuated the bias-variance trade-off observed among the single-biased models. Because elastic net combines the benefits of both lasso and ridge regularization, a balance can be obtained between their objectives. Ridge regularization can help temper the quantity of base learners excluded by lasso regularization. Lasso regularization can help ensure ridge regularization does not shrink the contributions of the base learners to the extent that underfitting occurs.

The effects of applying the regularization parameters to the stacking algorithm are visualized in Figure 2. Common among the lasso, ridge, and elastic net regularization methods is the sharp drop in contribution assigned to the Gaussian-biased base learner for most datasets in all three algorithms. As performance generally improved for the regularization methods, this drop in magnitude for the Gaussian-biased base likely indicates that the stacking algorithm relied on this base learner too heavily, which lead to overfitting. The changes to the other base learner contribution magnitudes were less distinct and more localized to specific datasets. Most followed a shrinking trend, which is in line with the objectives of the regularization algorithms and provides further evidence of overfitting by the stacking algorithm which necessitated the use of fewer base learners or

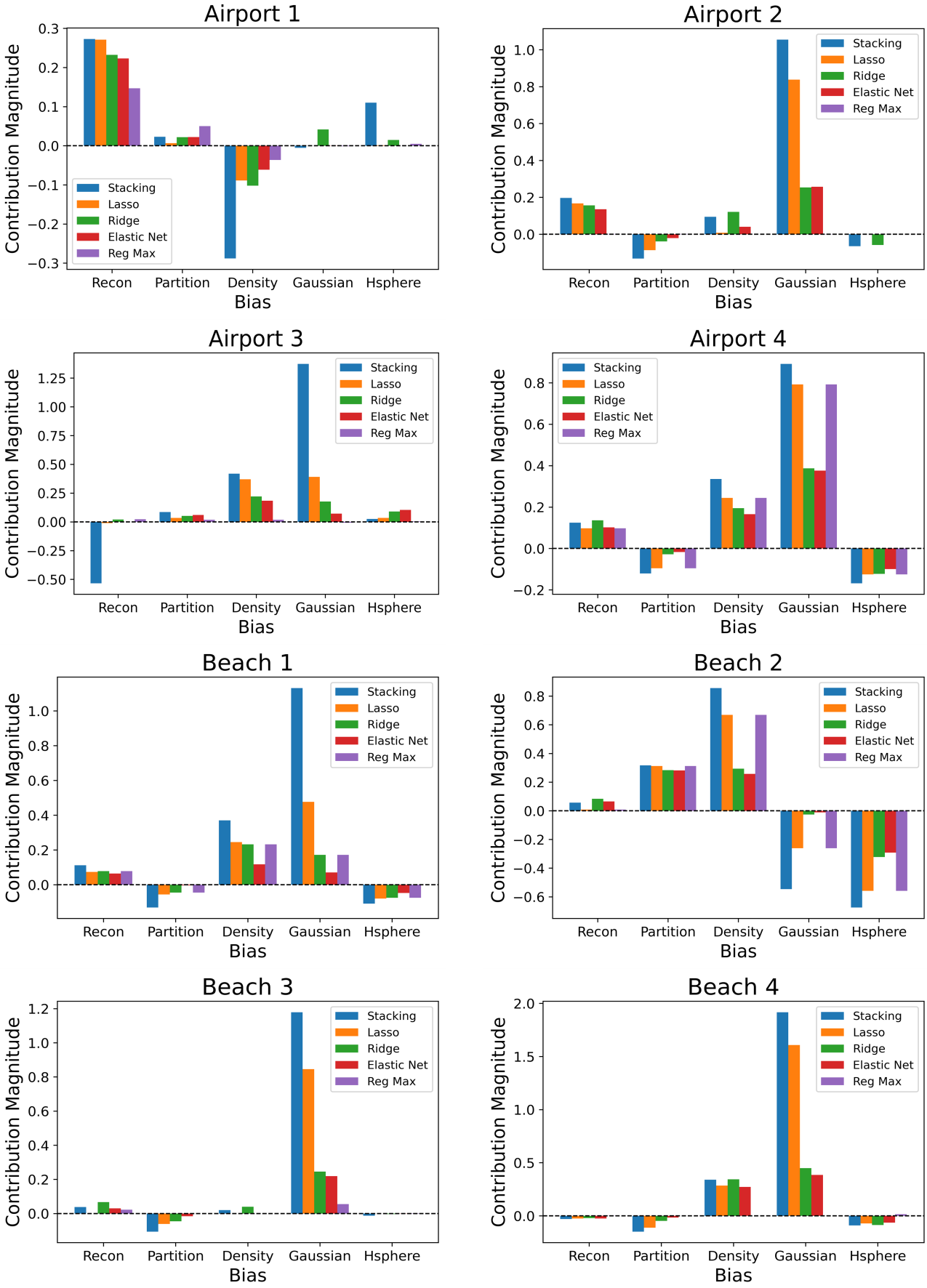


Figure 2: Contribution magnitudes by each stacking-based algorithm for each dataset for each base learner.

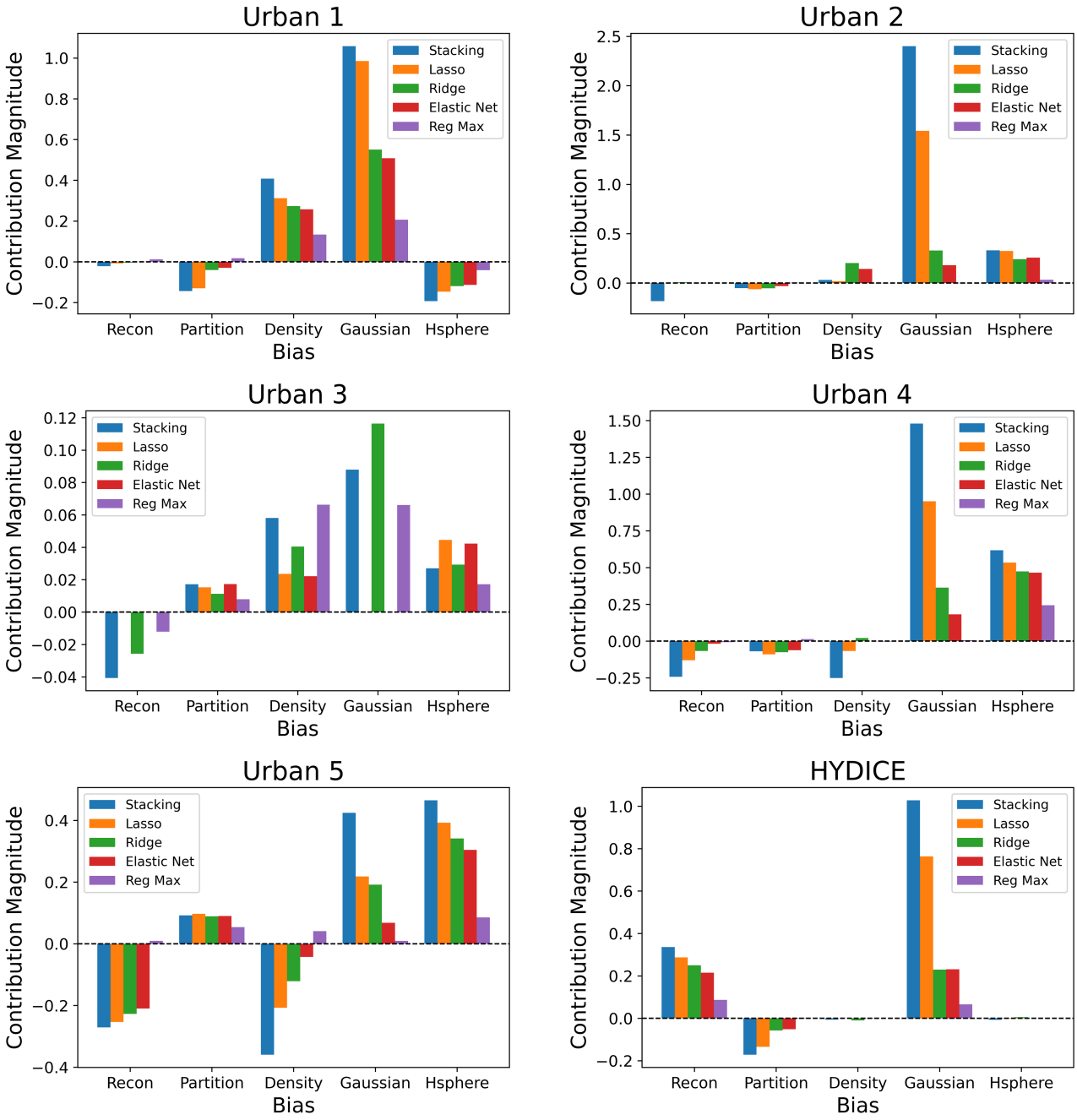


Figure 2: (continued)

smaller base learner contribution magnitudes.

Table 3 demonstrates the effects of the learning objectives of the regularization algorithms on the contribution magnitudes. The contribution magnitude delta is the difference between the means of the contribution magnitudes assigned by the stacking algorithm and the corresponding regularization algorithm identified in the algorithm column. The "Num Non-Contributors" column identifies the frequency with which a contribution magnitude assigned to a base learner was less than one percent of the total range of magnitudes assigned to the other base learners for a given dataset. This is indicative of where contributions of targeted base learners are removed, where sparsity is being enforced.

Table 3: Contribution magnitude comparisons of the regularization algorithm

Algorithm	Contribution Magnitude Delta	Num Non-Contributors
Lasso	0.1174	14
Ridge	0.1916	1
Elastic Net	0.2242	13

The values in Table 3 are in line with the learning objectives of the regularization algorithms. The objective of the lasso algorithm is sparsity; thus it has the higher number of non-contributing magnitudes. The ridge algorithm attempts to make the contributions of all the base learners small. This has the effect of increasing the contribution magnitude delta from the stacking algorithm by over 63%, signifying a large decrease in assigned magnitudes to the base learners universally. Their universal decrease is additionally evidenced by the low num non-contributors generated by the algorithm, indicating that the contribution magnitudes of the base learners were lowered uniformly. The elastic net algorithm balances both the lasso and ridge algorithms keeping nearly all the non-contributors (13 of the 14) and shrinking the other coefficients to further increase the delta from the stacking algorithm.

The max reg results in Table 2 and Figure 2 provide empirical evidence towards the quantity and magnitude of contributions required from the base learners to achieve optimal HAD performance. The max reg results in Table 2 demonstrate that employing multiple base learners improves performance over utilizing the base learners individually for all datasets. The max reg results in Figure 2 indicate that some subset of base leaners or a small level of contribution from most base learners was optimal. Therefore, in this study, more than one HAD bias was always required to achieve maximal performance on each of the datasets. This provides evidence for the viability of multi-biased HAD models.

Table 4 provides the regularization algorithm and corresponding λ parameter(s) that achieved the highest AUROC score on each dataset individually. It is worth noting that each dataset benefitted from different regularization objectives and strengths. This, in addition to the varying degrees of contribution assigned to the base learners in the max reg cases, further demonstrates how different each dataset is and why no singular HAD bias is likely to universally perform well across a more diverse range of datasets. The Airport 2, Airport 3, Beach 4, Urban 2, Urban 4, and Urban 5 datasets benefitted from much stronger levels of regularization than the other datasets. This resulted in mostly minimal magnitude of contributions from the base learners (Fig 2). Comparatively, the Airport 4, Beach 1, Beach 2, Beach 3, Urban 1, Urban 3, and Urban 6 datasets benefitted from weaker levels of contribution that resulted in moderate to stronger magnitudes of contribution from the base learners, or some stronger and some weaker magnitudes of contribution in the cases where lasso regularization was most beneficial. This demonstrates that both the quantity and magnitude of contribution is important towards optimizing performance.

Two instances did exist where regularization did not improve performance over bagging or stacking. In the Airport 1 dataset the stacking algorithm achieved the highest AUROC score among all the algorithms in the study (Table 2). This indicates that contributions from all the base learners are likely important toward optimizing performance for this dataset. The partitioning-, density-, Gaussian-, and hyperspherical-biased base learners also had their lowest performances among all the datasets on the Airport 1 dataset. This created the

Table 4: Best regularization model and parameter(s) for each dataset individually

Dataset	Regularization Algorithm	λ Lasso	λ Ridge
Airport 1	Ridge	-	0.005
Airport 2	Elastic Net	0.1	0.001
Airport 3	Elastic Net	0.1	0.005
Airport 4	Lasso	0.001	-
Beach 1	Ridge	-	0.001
Beach 2	Lasso	0.001	-
Beach 3	Elastic Net	0.001	0.005
Beach 4	Lasso	0.05	-
Urban 1	Ridge	-	0.005
Urban 2	Elastic Net	0.1	0.1
Urban 3	Ridge	-	0.005
Urban 4	Lasso	0.1	-
Urban 5	Ridge	-	0.1
HYDICE	Elastic Net	0.001	0.005

noisiest inputs for the ensemble learning algorithms to decipher among the datasets, increasing the difficulty of learning. In the Urban 5 dataset bagging produced the highest AUROC score among all the algorithms in the study (Table 2). This indicates that all the contributions from the base learners are equally important and further optimization could not be found from modifying their magnitude of contribution. However, the results of the base learners are largely uniform for the Urban 5 dataset, which increases the difficulty of finding optimal solutions with stacking-based algorithms, such as the regularization algorithms employed in this study. The results in Figure 2 suggest that an inverse contribution of the reconstruction and density-biased base learners may help performance. Therefore, introducing a prior into the stacking algorithm that encourages convergence towards these trends may help identify more optimal contribution magnitudes using the regularization algorithms on the Urban 5 dataset.

5 Discussion

The results of this study indicate that HAD performance was universally improved by employing multiple HAD algorithm biases to model each dataset (Table 2). By computing the central tendencies of multiple HAD algorithm biases, the individual weaknesses of each bias were mitigated. This had a pronounced effect on the overall performance achieved across all the datasets, (i.e., the generalizability) in addition to improving performance over the base learners on datasets individually. As such, multi-biased HAD models were able to improve individual and generalized performance, and therefore attenuated the bias-variance trade-off observed among the single-biased models. This provided empirical evidence for the utility of multi-biased HAD algorithm solutions.

Central to achieving peak HAD performance was identifying the optimal level of inclusion or exclusion to assign each base learner towards the final deviation predictions for a given dataset (Fig 2). By enabling different levels of contribution for each of the base learners, the strength of targeted base learners could be enhanced in addition to the limitations of others being mitigated. This diminished sensitivities of the multi-biased models to disparities between the base learners and enabled the inverse contribution of base learners to temper false positive results.

While leveraging multiple base learners improved performance on all datasets used in this study, the minimization or exclusion of targeted base learners proved most impactful toward achieving peak performance.

Results of the regularization methods in Table 2, and the low λ values used to produce those results, indicate a fine balance between excluding base learner biases that are not useful, while retaining sufficient information to enhance the strengths and mitigate the weaknesses over individual base learner biases. Peak performance among all the algorithms employed in this study was achieved by the algorithm that best balanced the removal of unneeded base learners while appropriately minimizing the contribution of the remaining base learners. This helped improve generalizability, prevent overfitting, and reduce model complexity (Table 3).

It is important to note that the optimal regularization objective and strength was not uniform across all datasets (Table 4). Some datasets benefitted from greater sparsity of contribution among the base leaners, lower complexity across all the base learners, or some combination of both. Some datasets ultimately benefitted from much strong levels of regularization as well, enabling more defined contributions from a select subset of base learners or small contributions across all base learners. This further demonstrates the complexity and diversity among hyperspectral images and difficulty in modeling the nuances across multiple datasets with a single base learner or more restrictive ensemble learning assumptions.

While multi-biased models did outperform single-biased models across all datasets in this study, performance on 2 of the 14 datasets was not optimized by employing regularization. Therefore, while multi-biased models were always beneficial, relaxing assumptions about the uniformity or quantity of base learner contributions was not. In the Airport 1 dataset, the poor performance of 4 of the 5 base learners created noise that was difficult for the ensemble learning algorithms to decipher. In the Urban 5 dataset, the largely uniform performance of the base leaners made delineating between them hard. Both noise and uniformity among the base leaners made converging on optimal contribution magnitudes challenging. A potential future direction may be to set the regularization term to converge to a specified prior or set the β parameter to a given prior directly. Information from the base learners and the bagging and stacking ensemble methods could be used to inform the prior. In this way, learning could be coerced to converge towards a more informed solution instead of being initialized stochastically.

6 Conclusion

This study investigated how the bias-variance trade-off observed in single-biased HAD models is affected by multi-biased HAD models. It was hypothesized that amalgamating multiple biases attenuates the effect of the bias-variance trade-off, improving performance on individual datasets and generalizability across a broad range of datasets. This study additionally explored whether equal, unequal, sparse, minimal, or mixed contributions among the base learners were optimal in the amalgamation process. This was done to provide empirical evidence as to the efficacy of multi-biased HAD models and provide direction as to the optimal method of amalgamation.

To conduct this study 5 distinct HAD biases were selected to identify anomalies across 14 diverse datasets. The results from each single-biased model were first computed. Then results from multi-biased models were computed. The multi-biased models were generated from ensemble learning algorithms that incorporated bagging, stacking, and stacking combined with lasso, ridge, and elastic net regularization techniques. These techniques enabled the generation of multi-biased models that assumed equal, unequal, sparse, minimal, or mixed contributions among the base learners. The quantity of top performances achieved by each model and overall performance achieved across all 14 datasets was computed to determine which model was the best performer and whether the bias-variance trade-off was successfully attenuated.

The results showed that multi-biased models were successful in achieving peak performance and attenuating the bias-variance trade-off over the single-biased models. Amalgamating the biases with elastic net regularization enabled the proper exclusion or reduction of unhelpful biases to achieve peak overall performance and improve performance on the greatest quantity of individual datasets over the singe-biased models. Consequently, the elastic net model was able to improve individual and generalized performance, and therefore attenuated the bias-variance trade-off observed among the single-biased models. Two cases did exist where employing regularization did not improve performance over other multi-biased models. This highlights sensitivities of

regularization to cumulatively poor and uniform information from the base learners. However, multi-biased models still outperformed single-biased models on all 14 datasets. The results of this study provide empirical evidence as to the efficacy of multi-biased HAD models over single-biased HAD models and provide direction for the most effective amalgamation strategies to construct optimal multi-biased HAD models.

References

[1] X. Kang, X. Zhang, S. Li, K. Li, J. Li, and J. A. Benediktsson, "Hyperspectral anomaly detection with attribute and edge-preserving filters," IEEE Transactions on Geoscience and Remote Sensing, July 2017.

[2] Y. Xu, L. Zhang, B. Du, and L. Zhang, "Hyperspectral Anomaly Detection Based on Machine Learning: An Overview," Ieee J Sel Top Appl, vol. 15, pp. 3351–3364, 2022, doi: 10.1109/jstars.2022.3167830.

[3] X. Hu et al., "Hyperspectral Anomaly Detection Using Deep Learning: A Review," Remote Sens-basel, vol. 14, no. 9, p. 1973, 2022, doi: 10.3390/rs14091973.

[4] H. Su, Z. Wu, H. Zhang, and Q. Du, "Hyperspectral Anomaly Detection: A survey," Ieee Geoscience Remote Sens Mag, vol. 10, no. 1, pp. 64–90, 2022, doi: 10.1109/mgrs.2021.3105440.

[5] I. Racetin and A. Krtalić, "Systematic Review of Anomaly Detection in Hyperspectral Remote Sensing Applications," Appl Sci, vol. 11, no. 11, p. 4878, 2021, doi: 10.3390/app11114878.

[6] N. R. Shah et al., "Hyperspectral anomaly detection: a performance comparison of existing techniques," Int. J. Digit. Earth, vol. 15, no. 1, pp. 2078–2125, 2022, doi: 10.1080/17538947.2022.2146770.

[7] Y. Huang, Y. Xue, Y. Su, and S. Han, "Hyperspectral Anomaly Detection Based on Isolation Forest with Band Clustering," IGARSS 2020 - 2020 IEEE Int. Geosci. Remote Sens. Symp., vol. 00, pp. 2416–2419, 2020, doi: 10.1109/igarss39084.2020.9323988.

[8] S. Li, K. Zhang, P. Duan, and X. Kang, "Hyperspectral Anomaly Detection With Kernel Isolation Forest," IEEE Trans. Geosci. Remote Sens., vol. 58, no. 1, pp. 319–329, 2020, doi: 10.1109/tgrs.2019.2936308.

[9] X. Song, S. Aryal, K. M. Ting, Z. Liu, and B. He, "Spectral–Spatial Anomaly Detection of Hyperspectral Data Based on Improved Isolation Forest," IEEE Trans. Geosci. Remote Sens., vol. 60, pp. 1–16, 2022, doi: 10.1109/tgrs.2021.3104998.

[10] Y. Li, K. Jiang, W. Xie, J. Lei, X. Zhang, and Q. Du, "A Model-Driven Deep Mixture Network for Robust Hyperspectral Anomaly Detection," IEEE Trans. Geosci. Remote Sens., vol. 61, pp. 1–16, 2023, doi: 10.1109/tgrs.2023.3309960.

[11] Y. Lu, X. Zheng, H. Xin, H. Tang, R. Wang, and F. Nie, "Ensemble and random collaborative representation-based anomaly detector for hyperspectral imagery," Signal Process., vol. 204, p. 108835, 2023, doi: 10.1016/j.sigpro.2022.108835.

[12] S. Wang, X. Hu, J. Sun, and J. Liu, "Hyperspectral anomaly detection using ensemble and robust collaborative representation," Inf. Sci., vol. 624, pp. 748–760, 2023, doi: 10.1016/j.ins.2022.12.096.

[13] P. Gurram, T. Han, and H. Kwon, "Hyperspectral anomaly detection using sparse kernel-based ensemble learning," Algorithms Technol. Multispectral, Hyperspectral, Ultraspectral Imag. XVII, pp. 80481C-80481C–9, 2011, doi: 10.1117/12.883383.

[14] P. Gurram, H. Kwon, and T. Han, "Sparse Kernel-Based Hyperspectral Anomaly Detection," IEEE Geosci. Remote Sens. Lett., vol. 9, no. 5, pp. 943–947, 2012, doi: 10.1109/lgrs.2012.2187040.

[15] X. Yang, X. Huang, M. Zhu, S. Xu, and Y. Liu, "Ensemble and Random RX With Multiple Features Anomaly Detector for Hyperspectral Image," IEEE Geosci. Remote Sens. Lett., vol. 19, pp. 1–5, 2022, doi: 10.1109/lgrs.2022.3167758.

[16] R. Wang, F. Nie, Z. Wang, F. He, and X. Li, "Multiple Features and Isolation Forest-Based Fast Anomaly Detector for Hyperspectral Imagery," IEEE Trans. Geosci. Remote Sens., vol. 58, no. 9, pp. 6664–6676, 2020, doi: 10.1109/tgrs.2020.2978491.

[17] S. Wang et al., "Subfeature Ensemble-Based Hyperspectral Anomaly Detection Algorithm," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 15, pp. 5943–5952, 2022, doi: 10.1109/jstars.2022.3191725.

[18] M. Younis, M. Hossain, A. Robinson, L. Wang, and C. Preza, "Hyperspectral unmixing-based anomaly detection," Comput. Imaging VII, vol. 12523, pp. 1252302-1252302–13, 2023, doi: 10.1117/12.2664706.

[19] S. Wang, X. Wang, L. Zhang, and Y. Zhong, "Auto-AD: Autonomous Hyperspectral Anomaly Detection Network Based on Fully Convolutional Autoencoder," IEEE Trans. Geosci. Remote Sens., vol. 60, pp. 1–14, 2022, doi: 10.1109/tgrs.2021.3057721.

[20] J. Qu, Q. Du, Y. Li, L. Tian, and H. Xia, "Anomaly Detection in Hyperspectral Imagery Based on Gaussian Mixture Model," IEEE Trans. Geosci. Remote Sens., vol. 59, no. 11, pp. 9504–9517, 2021, doi: 10.1109/tgrs.2020.3038722.

[21] S. Li, K. Zhang, P. Duan, and X. Kang, "Hyperspectral Anomaly Detection With Kernel Isolation Forest," IEEE Trans. Geosci. Remote Sens., vol. 58, no. 1, pp. 319–329, 2020, doi: 10.1109/tgrs.2019.2936308.

[22] A. Banerjee, P. Burlina, and C. Diehl, "A Support Vector Method for Anomaly Detection in Hyperspectral Imagery," IEEE Trans. Geosci. Remote Sens., vol. 44, no. 8, pp. 2282–2291, 2006, doi: 10.1109/tgrs.2006.873019.

[23] H. Kwon and N. M. Nasrabadi, "Kernel Rx-Algorithm: A Nonlinear Anomaly Detector for Hyperspectral Imagery," IEEE Trans. Geosci. Remote Sens., vol. 43, no. 2, pp. 388–397, 2005, doi: 10.1109/tgrs.2004.841487.

[24] H. Jiang, "A manifold constrained multi-head self-attention variational autoencoder method for hyperspectral anomaly detection," In 2021 International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA), Dec. 2021.

[25] J. Zhang, Y. Xu, T. Zhan, Z. Wu, and Z. Wei, "Anomaly Detection in Hyperspectral Image Using 3D-Convolutional Variational Autoencoder," 2021 IEEE Int. Geosci. Remote Sens. Symp. IGARSS, vol. 00, pp. 2512–2515, 2021, doi: 10.1109/igarss47720.2021.9554184.

[26] B. Tu, X. Yang, N. Li, C. Zhou, and D. He, "Hyperspectral anomaly detection via density peak clustering," Pattern Recognit. Lett., vol. 129, pp. 144–149, 2020, doi: 10.1016/j.patrec.2019.11.022.

[27] G. Zhang, N. Li, B. Tu, Z. Liao, and Y. Peng, "Hyperspectral Anomaly Detection Via Dual Collaborative Representation," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 13, pp. 4881–4894, 2020, doi: 10.1109/jstars.2020.3009324.

[28] B. Tu, N. Li, Z. Liao, X. Ou, and G. Zhang, "Hyperspectral Anomaly Detection via Spatial Density Background Purification," Remote Sens., vol. 11, no. 22, p. 2618, 2019, doi: 10.3390/rs11222618.

[29] M. Bernhardt, J. Heather, and M. Smith, "New models for hyperspectral anomaly detection and unmixing," Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery, June 2005.

[30] J. Nothard, N. Kent, C. West, J. Wood, and W. Oxford, "Full system modeling for hyperspectral target detection and identification," Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery, Jan. 2003.

[31] H. Kwon and N. M. Nasrabadi, "Kernel Rx-Algorithm: A Nonlinear Anomaly Detector for Hyperspectral Imagery," IEEE Trans. Geosci. Remote Sens., vol. 43, no. 2, pp. 388–397, 2005, doi: 10.1109/tgrs.2004.841487.

[32] A. Schaum, "Joint Subspace Detection of Hyperspectral Targets≫," 2004 IEEE Aerosp. Conf. Proc. (IEEE Cat 04TH8720), vol. 3, pp. 1818–1823, 2004, doi: 10.1109/aero.2004.1367963.

[33] H. Kwon, S. Z. Der, and N. M. Nasrabadi, "Adaptive anomaly detection using subspace separation for hyperspectral imagery," Opt. Eng., vol. 42, no. 11, pp. 3342–3351, 2003, doi: 10.1117/1.1614265.

[34] L. S. Kalman and E. M. B. III, "Classification and material identification in an urban environment using HYDICE hyperspectral data," Imaging Spectrometry III, Oct. 1997.