Modified Autoencoder Training and Scoring for Robust Unsupervised Anomaly Detection in Deep Learning

NICHOLAS MERRILL AND AZIM ESKANDARIAN, (Senior Member, IEEE)
Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24060, USA
Corresponding author: Nicholas Merrill (nmerrill@vt.edu)

ABSTRACT The autoencoder (AE) is a fundamental deep learning approach to anomaly detection. AEs are trained on the assumption that abnormal inputs will produce higher reconstruction errors than normal ones. In practice, however, this assumption is unreliable in the unsupervised case, where the training data may contain anomalous examples. Given sufficient capacity and training time, an AE can generalize to such an extent that it reliably reconstructs anomalies. Consequently, the ability to distinguish anomalies via reconstruction errors is diminished. We respond to this limitation by introducing three new methods to more reliably train AEs for unsupervised anomaly detection: cumulative error scoring (CES), percentile loss (PL), and early stopping via knee detection. We demonstrate significant improvements over conventional AE training on image, remote-sensing, and cybersecurity datasets.

INDEX TERMS Hyperspectral, HSI, deep learning, anomaly detection, unsupervised, autoencoder, outlier detection, condition monitoring.

I. INTRODUCTION

Anomaly (outlier) detection is of critical importance across many domains, including fraud identification, video surveillance, medical applications, remote sensing, and network monitoring. Deep learning methods have demonstrated the ability to model high-dimensional data, to define complex boundaries between normal and anomalous behavior, and to scale to large volumes of data [1]. A majority of the progress in the field of deep learning has been made on supervised tasks in which labeled data is available during training. However, supervised learning is not well suited to anomaly detection because obtaining a sufficient number of labeled anomalous examples is often infeasible [2]–[4].

Semi-supervised approaches to deep anomaly detection (DAD) attempt to circumvent the need for labeled anomalous examples by using only readily available, normal examples to build a model of the data. Examples that do not conform to the model are then identified as anomalous. Unfortunately, semi-supervised techniques are susceptible to overfitting or underfitting, the effect of which is poor recall or precision, respectively [3]–[5].

An even more challenging problem, but a more realistic assumption, is that of unsupervised anomaly detection, where no labels are available. The training data is not clean and may contain anomalies or ambiguous normal examples. The unsupervised training process alone attempts to separate normal and anomalous examples. Unsupervised methods can be used to more readily label normal or anomalous examples for semi-supervised or supervised learning. Also, when patterns distinguishing anomalous and normal behavior change over time, unsupervised approaches are necessary [2]. Because of the limitations of other approaches, unsupervised DAD is a critical area of research and of vital practical importance [4].

Autoencoders (AE) and their many variants can fit highly complex patterns and scale to large collections of data [6]–[8]. As a result, AEs form the fundamental architecture of unsupervised DAD. A contractive AE consist of two models that are jointly trained: the encoder and decoder. The encoder obtains a compressed representation of the initial input, while the decoder attempts to reconstruct the input from the compressed representation. The compression places a constraint on the AE, forcing the network to generalize features [4].

A conventional, AE-based anomaly detector uses the reconstruction error as an anomaly score. This approach assumes that normal examples will be reconstructed more accurately than anomalous examples based on greater frequency during training. However, this method also assumes that anomalies cannot be reconstructed accurately. In reality,
AEs can often generalize well enough to closely reconstruct anomalous inputs [9], [10].

This over-generalization cannot be corrected by simply enforcing more regularization, restricting capacity, or reducing training time, all of which may jeopardize the reconstruction of normal examples [9]. Moreover, without labels, the optimal hyperparameters cannot be determined from an extensive search in the unsupervised setting.

To address the problem of AEs over-generalizing to fit anomalous data, we propose two modifications to both the anomaly scoring and objective functions of AEs used for unsupervised DAD. Rather than assuming an AE cannot learn to generalize anomalies, we instead assume that the anomalies require greater time to learn. That is, anomalous examples during training will have higher reconstruction scores over more training steps. We are then able to capture anomalous examples by leveraging the unsupervised AE training process.

The main contributions of this paper are as follows:
1) We introduce cumulative error scoring (CES). Conventionally, AE training is stopped after a number of training steps (a hyperparameter). The reconstruction errors are then calculated for all examples. With CES, at the end of each training epoch, the reconstruction error of each input is calculated, and the errors are added to the previous history of errors.
2) We introduce percentile loss (PL) training. A typical AE loss (objective) function, such as mean-squared-error, aims to reduce the error across an entire batch of examples. Yet in doing so, anomalies contribute disproportionately to parameter updates, which consequently trains the AE to reconstruct anomalies. Rather than making updates based on the loss of the entire minibatch, PL only accounts for the lowest $P$ percentile of examples based on their error.
3) We introduce early stopping via knee detection. The conventional reconstruction-error based loss is not a good indicator of anomaly detection performance and can exhibit erratic epoch-to-epoch behavior. Instead, the average loss of each example is used to create a smoother loss curve, from which a knee can be reliably detected. Rather than choosing a stopping epoch, which is unreliable, training is halted based on the number of multiples past the detected knee.

We expand upon our previous related work by demonstrating the ability of CES and PL to prevent AEs from generalizing anomalies across a number of applications, allowing greater reliability in unsupervised DAD [11].

II. RELATED WORK
Semi-supervised anomaly detection commonly employs traditional approaches such as the one-class support vector machine (SVM). However, a number of DAD methods have shown promising results, especially in larger datasets with higher dimensional features. [4], [12]. Semi-supervised DAD methods often employ AE architectures, as well as generative models such as variational autoencoders (VAE) and generative adversarial networks (GANs) to model normal data [12], [13]. Other hybrid approaches use the embedding space of AEs, VAEs or hidden layers of pre-trained classifiers to first reduce the dimensionality of examples, before applying more traditional detection methods [14]–[16]. VAEs can be used to shape the embedding space for one-class classification [17]. Alternatively, deep one-class classification approaches involve modifying the training objective to extract features that differentiate anomalies [4], [18], [19].

Methods designed for semi-supervised DAD can often be applied in an unsupervised setting as well. Yet many methods encounter difficulties in hyperparameter selection and suffer from sensitivity to anomalies in the training data [20]. Traditional methods of unsupervised anomaly detection generally use measurements of distances, densities, or clustering to differentiate normal and anomalous points. Amongst these are k-means, nearest-neighbor, and Gaussian mixture models [21]. Isolation forest is a notable, decision tree-based exception [22]. These approaches, however, suffer from issues of scaling in both data volume and complexity [3], [4], [23].

AEs and related, reconstruction-based methods are central to most unsupervised DAD approaches. Deep AEs have been used to model and detect anomalies in high dimensional multivariate point [24], image [25], temporal [26], and spatiotemporal data [27]. AE architectures have incorporated both 2D and 3D convolutional layers and recurrent modules such as RNNs, GRUs, and LSTMs [6]–[8]. We refer to Chalapathy et. al. for a recent survey of DAD methods [4]. This paper focuses on the contractive, fully-connected architecture, but the methods can be easily extended to other AE architectures or reconstruction-based, DAD methods.

Though recent works have trained AEs for anomaly detection tasks, few have remarked on their significant limitations in the unsupervised setting. Beggel et al. proposed a hybrid approach that iteratively refined the training set by using a one-class SVM in the latent space of an adversarial autoencoder (AAE) to remove suspected anomalous examples [9]. Other boosting techniques and AE cascades have likewise shown resistance to the generalization of anomalies [28]. Gong et al. used a memory-augmented AE to memorize normality and limit the latent representations in order to prevent the model from learning anomalous examples [10]. Robust Convolutional Autoencoders (RCAEs) and related methods extend robust PCA to DAD by learning a nonlinear subspace via an AE that captures most of the normal features, while providing a margin to account for anomalies during training [29], [30].

Our work is distinct in the fact that our methods focus on the reconstruction in the ambient space and do not require any modifications to network architecture. CES and PL can potentially be combined with existing, reconstruction-based techniques, including the those discussed, to further improve robustness and performance.
III. MODIFIED UNSUPERVISED AUTOENCODER TRAINING AND SCORING

The standard AE is composed of two networks. An encoder network, $\mathcal{E}$, maps data from an input example $x \in X \subset \mathbb{R}^D$ in ambient space to a reduced latent space $z \in Z \subset \mathbb{R}^K$. Then a decoder network, $D$ maps the latent space representation back to the ambient space, $x' \in X' \subset \mathbb{R}^D$.

The encoder and decoder are jointly trained by an objective function that aims to minimize the reconstruction error between the set of inputs $X$ and the reconstructions $X'$. Measures of reconstruction error are used to both train the AE and to provide a measure of normality for each example. Here we use the $l2$-based mean squared error (MSE),

$$MSE(x) = L(x, x') = \|x - (D \circ \mathcal{E})x\|^2. \quad (1)$$

A. CUMULATIVE ERROR SCORING

Figure 1 shows reconstruction error of an anomalous bag example and normal dress example from one of the Fashion MNIST runs detailed in Section IV-A. The converging errors demonstrate the problem of generalization in unsupervised AE training, while the diverging bold lines illustrate part of our proposed solution.

![Figure 1. Reconstruction error over 500 epochs for a normal dress (blue) and anomalous bag (red). Bold lines show the cumulative error.](image)

Early in training, the reconstruction errors of anomalous examples are well separated; yet over time, the AE learns to reconstruct the anomalous example equally well as the normal one. In the extreme example, where the loss is zero, the AE has no ability to distinguish anomalies. Over-training is obviously problematic, but arbitrarily choosing a stopping epoch may prevent the AE from fully assimilating the normal examples. In order to allow for greater laxity in the number of training steps and to fully leverage the history of the training process, we introduce Cumulative Error Scoring (CES). CES sums the errors of each example across all training epochs.

The CES for each example $x$ is then,

$$CES(x) = \sum_{j=b}^{J} MSE(x)_j,$$  \quad (2)

where $MSE(x)_j$ is the reconstruction error at the end of epoch $j$ and $b$ is the number of burn-in epochs.

The CES metric can be understood as the summation of the reconstruction errors from an ensemble of earlier states of the model during training. CES can be alternatively viewed as an approximation of the integrated error, as shown in Figure 1. CES produces an equivalent ranking to averaging the errors over the training epochs. The bold lines in Figure 1 represent the cumulative errors of both anomalous and normal examples. CES more reliably separates the anomaly over the course of training. Additionally, the summation places less significance on later epochs where the reconstruction errors are smaller and the network is more likely to be over-trained. Specifying a small number of burn-in epochs, $b$, acts to ignore the some of the initial period of training where the reconstruction errors are not reliably indicative of normality.

B. PERCENTILE LOSS

As shown in Figure 4, CES improves accuracy but does not directly address the contamination of training data by anomalies. After the AE is able to generalize the anomalies, continued summation of error degrades performance over time as anomalous reconstruction errors are no longer reliably higher than those of background examples.

Even if the training data contains a relatively low percentage of anomalies ($\alpha$), there is still a significant probability that an anomaly will be present in any given mini-batch, as described by the hypergeometric distribution. The probability of contamination, $P_c$, that a randomly drawn mini-batch of size $N_b$ contains at least one anomaly from a dataset that contains $N$ examples is given by,

$$P_c = 1 - \frac{c(aN) \times (N-cN)}{N \times N_b}.$$ \quad (3)

An example calculation can provide more insight. If $N = 1000$, $\alpha = 1\%$, and $N_b = 100$, then $P_c = 65.3\%$. In this example, most mini-batches will contain at least one anomaly, regardless of the detector’s current ability to distinguish anomalous examples.

If an anomaly is present, it will contribute to the parameter updates. Worse yet, the higher reconstruction errors will cause anomalies to contribute disproportionately. This problem stems from the conventional, dual use of reconstruction error as both the training target and the basis of an anomaly score (1). The training process directly acts to reduce the anomaly scores of anomalous examples.

To better adapt AEs for anomaly detection, we propose percentile loss (PL) training. PL undermines an AE’s ability to learn anomalies while still encouraging the generalization of normal examples. PL leverages the assumption that early in training, the anomalous examples will more often generate...
the highest errors in a given mini-batch. Rather than updating parameters based on the errors of all the examples in a mini-batch, we define an upper percentile \( q \) (e.g. \( q = 95\% \)) and a reconstruction error in each mini-batch corresponding to that percentile, \( P_q \). PL then only performs parameter updates based on the reconstruction errors less than \( P_q \).

Even if we assume that a detector is able to perfectly rank all anomalous examples above normal examples, it is still possible that a randomly drawn mini-batch contains enough anomalies that some number exists below the threshold. The probability of contamination by at least one anomaly less than \( q \) in a perfect detector is given by

\[
P_{c*} = 1 - \sum_{i=0}^{N_b-L} \binom{aN}{i} \binom{N-aN}{N_b-i},
\]

where we define \( L \) to be the position of \( q \) among the ranked scores, \( L = \lfloor N_b(q/100) \rfloor \). Using the same example values as before, we see a massive reduction in the probability compared to training on the full mini-batch, \( P_{c*} = 0.15\% \). The probability increases if the detector is not able to fully separate the anomalies above the threshold.

In practice, training networks using larger mini-batch sizes exhibit a degradation in generalization [31]. This is generally an argument against training with very large batch, but it does not hold for anomaly detection where there is not a distinction between training and testing data. Instead, over-generalization is potentially problematic. Upon the initial inspection of equation 4, PL should benefit from larger mini-batch sizes, as there is a reduced chance that an anomaly falls below the threshold; notwithstanding, too large of a batch size may cause the failure to generalize more challenging examples from the normal class. Additionally, in extreme cases high dimensional (e.g. volumetric) data may be limited to very small batch sizes. In the extreme case of the batch containing just a single example, there is no difference between PL and MSE loss.

A simple experiment shown in Figure 2 illustrates this trade off on the Digit 6/Digit 9 MNIST test set, described in the later section. Anomaly detection performance, as measured by average precision (AP), is tracked over 500 training epochs using four different batch sizes using both the conventional MSE loss and PL. Performance improves for all mini-batch sizes using PL. The results also indicate that larger batch-sizes are not necessarily better. As explained earlier, this is possibly due to difficult normal examples, failing to generalize. Though the best performance was reached with a batch size of 1024, an improvement in average precision was achieved with a batch size of just 32. A potential means to extend PL to high dimensional data where the batch size is extremely limited is to calculate a *ghost* threshold based on the reconstruction errors of many forward passes that are not necessarily part of the batch update [31]. Then, examples can be accepted or rejected based on this cutoff. Though this process is slower, it circumvents the storage limitations of larger batch sizes.

Overall, PL helps to prevent anomalies from contributing to parameter updates when they are present in the training data. One potential issue is that PL will cause normal examples above the percentile threshold to be ignored. However, we assume that by sufficiently training on other normal examples the AE is still able to generalize the more difficult normal examples better than the anomalies. Furthermore, difficult normal examples that fall above the threshold in one-batch, because of the stochastic draws, may appear below the threshold in another batch. MSE (1) serves as the base loss function for all evaluations in this paper; however, the application of PL can easily be extended to other reconstruction-based training objectives.

### C. EARLY STOPPING VIA KNEE DETECTION

Despite the protections afforded by CES and PL, the Modified Training and Scoring Autoencoder (MTS-AE) can degrade slowly over many training steps, as indicated by the slight downward trend of the bold line in Figure 3a. We propose a means of early stopping that is not tied to the magnitude of the loss and reliable stops training near the optimal point. First, the cumulative error for each is divided by the number of epochs to acquire the average error across all training epochs. These individual averages are then together averaged at the end of each epic, creating an *average loss* statistic as seen in Figure 3, which shows a run from the urban-2 dataset described later. The knee in the average loss curve then serves as a criterion to end training. We deploy the Kneedle algorithm [32], with a sensitivity parameter of \( S = 5 \) to determine the knee at the end of each epoch. Attempting to apply the Kneedle algorithm to the standard epoch-to-epoch MSE loss can cause issues, as it does not reliably produce a smooth curve. Additionally, Figure 3a illustrates the sensitivity of detection to the choice of stopping epoch with conventional MSE-AE training.
IV. EXPERIMENTS

Three modifications to AE training for unsupervised anomaly detection have been presented: CES, PL, and early stopping via knee detection. The benefits of our Modified Training and Scoring Autoencoder (MTS-AE) over conventional AE training are evaluated in this section. Creating a fair comparison across all anomaly techniques, especially in the unsupervised setting, is challenging [3]. We do not attempt to reproduce all possible variations, as the proposed, modified approach can be applied to a large number of architectures and combined with other techniques such as sparse, adversarial, memory-augmented, and energy-based AEs [10], [33], [34]. Furthermore, we do not suggest that these modifications alone will broadly produce state-of-the-art results. Instead, through comparison against standard MSE-AE training, we hope to encourage adoption of the proposed methods by other AE variants and DAD methods as a means of improving accuracy and robustness.

In each of our evaluations, we report the Area Under the Receiver Operation Characteristic curve (AUROC). The AUROC quantifies false positive and true rates across all possible threshold settings as set by the anomaly scores assigned to each example. In the unsupervised setting, AUC is most usefully interpreted as the expectation that a randomly drawn, anomalous example will be scored higher than a randomly drawn, normal example. Consequently, AUROC varies between 0.5 (random scoring) and 1.0 (perfect detection). In addition to AUROC, the Average Precision (AP) is also provided. The AP gives the precision of each threshold weighted by the previous increase in recall. AP varies from 0 to 1, and because it does not reward detection of more abundant true negatives, AP can be more informative than AUROC in problems with high class imbalance [35].

The AEs are trained with the Adam optimizer and implemented using the Keras API [36], [37]. Additionally, a \(1 \times 10^{-5}\) \(\ell_2\) regularization is enforced on all layer outputs. ReLU activations are used after each hidden layer with no activation on the final layer. A mini-batch size of 1024 is used throughout. For PL, we set \(q = 95\%\). For CES, we set \(b = 5\). These choices represent heuristics; however, if a small amount of labeled data is available tuning can produce better results. We forgo doing so to preserve the unsupervised framework as best as possible. For the same reason, we use simple AE architectures and learning rates established by other similar studies.

In the strict definition of unsupervised anomaly detection, there is not a distinction between training and testing sets [3]. However, there is no established rule for halting conventional AE training. In order to provide a fair evaluation, some holdout sets are used for choosing reasonable hyperparameters for each architecture and domain. This use of labeled data may be considered a form of weakly supervised learning; however, the use is only intended to provide a reasonable comparison and is not required in practice [38]. Details on hyperparameter tuning are provided in each of the following subsections.

A. EXPERIMENTS ON IMAGE DATA

First, we evaluate our methods on 28 \(\times\) 28 grayscale images from the MNIST and Fashion-MNIST datasets [39], [40]. The Fashion-MNIST data contains examples of clothing items and is more complex than the classic MNIST handwritten digits. As shown in Table 1 and Table 2, different combinations are used to demonstrate the influence of the percentage of anomalies during training, \(\alpha\), and the type of anomalies. Mix indicates the equal combination of anomalies from all other classes, otherwise the anomalies are taken from a single class. We use the same AE structure for all MNIST and Fashion-MNIST evaluations. The images are first flattened...
into 784 dimensional vectors. We follow the architecture used by [41], FC(784, 256)-FC(256, 32) to FC(32, 256)-FC(256, 784), where FC(i, o) indicates the input and output dimensionality of each fully connected (FC) layer. The learning rate is set to $10^{-3}$. The number of steps per epoch was set to 2. Larger choices occasionally caused peak detection performance to be reached before the end of the first training epoch.

As discussed in the previous sections, choosing when to halt training plays a significant role. The first half of the digit 0 class and a random sample of the digit 2 class form the normal and anomaly ($\alpha = 2\%$) classes used for choosing the stopping epoch for the conventional MSE-AE as well the knee-multiple, $M$, for the MTS-AE. Three runs are performed and the choice that produces the best AP are averaged. Based on this process, in the MNIST evaluations the stopping epoch for the MSE-AE is set to 64, while the choice of $M = 6.2$ is used for the MTS-AE. We follow a similar procedure to create a normal pullover and anomalous ($\alpha = 4\%$) boot classes for the Fashion-MNIST evaluations, where the stopping epoch is set to 10, while $M = 3.6$. The images in either of the tuning sets are not used in any other evaluation.

Following the hyperparameter selection, evaluations are first performed on datasets most similar to the tuning-set, where the other half of the normal examples from the tuning set are combined with new random anomalies. Then, other combinations of normal and anomalous classes are used to evaluate how the conventional MSE-AE and the MTS-AE perform under new, but similar conditions using same choices for stopping; however, $\alpha$ is held constant. Each test is repeated three times and the average AUROC and AP are reported. Table 1 reports the results on the MNIST data, Table 2 shows the scores for the Fashion-MNIST tests. The results show that conventional MSE-AE training causes a sensitivity to the choice of stopping epoch due to the over-generalizing of anomalies present in the training data. In all cases, MTS-AE produced higher average AUROC and AP scores.

### TABLE 1. The average AUROC (left) and AP (right) scores on the Fashion-MNIST test sets. The * indicates the tuning set.

| background | anomaly | N  | $\alpha$ | MSE-AE       | MTS-AE       |
|------------|---------|----|----------|---------------|---------------|
| Digit 0*   | Digit 2 | 3451 | 2%       | 0.967 | 0.493 | 0.996 | 0.848 |
| Digit 0    | Digit 2 | 5665 | 2%       | 0.992 | 0.742 | 0.996 | 0.904 |
| Mix        | Digit 6 | 7089 | 2%       | 0.981 | 0.670 | 0.992 | 0.790 |
| Digit 9    | Digit 5 | 7117 | 2%       | 0.976 | 0.557 | 0.990 | 0.777 |

### TABLE 2. The average AUROC (left) and AP (right) scores on the Fashion-MNIST test sets. The * indicates the tuning set.

| background | anomaly | N  | $\alpha$ | MSE-AE       | MTS-AE       |
|------------|---------|----|----------|---------------|---------------|
| Pullover*  | Boot    | 3500 | 4%       | 0.868 | 0.132 | 0.932 | 0.297 |
| Pullover   | Boot    | 3948 | 4%       | 0.984 | 0.737 | 0.990 | 0.846 |
| Mix        | Bag     | 7264 | 4%       | 0.867 | 0.385 | 0.871 | 0.426 |
| Dress      | Coat    | 7264 | 4%       | 0.899 | 0.215 | 0.904 | 0.250 |

B. ABLATION STUDY

In this section, we will conduct several further ablation studies to investigate how the components of the MTS-AE, namely CES and PL, contribute to improved and more reliable anomaly detection performance. Figure 4a illustrates the separate role of PL and CES on one of the runs from the Digit 0 versus Digit 2 dataset, while 4b reports a run from the dress versus bag dataset.

We see that CES provides a buffer from the erratic epoch-to-epoch performance of the MSE training and reduces the impact of the sharp drop around epoch 20, decreasing the sensitivity to the choice of stopping epoch. PL increases performance and helps to reduce the impact of training with anomalies by attempting to exclude them from updates.
and the learning rate is set to 1e-4. Because of the much larger number of examples and lower learning rate, the steps per epoch is increased to 100. Using MTS requires a full evaluation at the end of each epoch, so limiting the total number of epochs during training is important.

We compare MTS-AE to standard AE training and Isolation Forest (IF) [22]. IF is commonly used for unsupervised anomaly detection on large datasets. IF requires the selection of two important parameters: the subsampling size, $\Psi$, and the number of trees, $t$. We follow the author’s suggestions by setting $\Psi = 256$ and $t = 100$ [22]. IF is implemented using the Python toolkit for detecting outlying objects (PyOD) [42].

In a similar procedure to that of the previous section, the normal class is divided in half and a sample of anomalies ($\alpha = 1\%$) are used for tuning. The highest AP after three runs indicates selecting an MSE-AE stopping epoch of 16 and setting $M = 6.72$ for the MTS-AE. The other half of the normal class is used for the remaining evaluations with varying percentages of anomalies. Anomalies used in one set are not repeated in another. Table 3 reports the average AUROC and AP across three runs of IF, MSE-AE, and MTS-AE. MTS-AE outperform IF, and MSE at all levels of $\alpha$.

| dataset | $\alpha$ | IF       | MSE-AE    | MTS-AE    |
|---------|----------|----------|-----------|-----------|
| KDD99*  | 1.0%     | 0.969 | 0.13     | 0.972 | 0.170 | 0.984 | 0.275 |
| KDD99   | 0.1%     | 0.991 | 0.082 | 0.994 | 0.130 | 0.997 | 0.214 |
|         | 1.0%     | 0.994 | 0.573 | 0.993 | 0.466 | 0.996 | 0.661 |
|         | 2.0%     | 0.990 | 0.525 | 0.990 | 0.579 | 0.992 | 0.658 |

D. EXPERIMENTS ON REMOTE SENSING DATA

Hyperspectral imagery (HSI) produces rich data in both the spatial and spectral domains, opening up a wide range of important remote sensing applications including environmental monitoring and accurate material detection [43]. Atmospheric conditions are highly dynamic and accessing a complete spectral database across all possible signatures is impractical. For these reasons, unsupervised data-driven techniques for detection in HSI are of particular interest. AEs have shown promising results in this task [44]–[46].

A hyperspectral image can be understood as a data cube with two spatial dimensions and one spectral dimension. Anomaly detection is the task of identifying interesting pixels among the background points. The proposed methods are evaluated using the publicly available Airport-Beach-Urban (ABU) dataset. We select the three scenes from each of the airport, beach, and urban categories, with the fourth airport scene used for hyperparameter tuning. All scenes were recorded by the recorded by the

\[\text{http://xudongkang.weebly.com/data-sets.html}\]
FIGURE 6. The (a) false color image, (b) anomaly mask, and (c) GRX, (d) MSE-AE, and (e) MTS-AE anomaly score heat maps for the urban-1 (top), urban-2 (middle), and urban-3 (bottom) HSI scenes. Brighter indicates a higher anomaly score.

Airborne Visible/Infared Imaging Spectrometer (AVIRIS) sensor [47]. The sensor features 224 spectral bands, but noise bands have been removed reducing the dimensionality. All scenes fill a 100 × 100 pixel area with the exception of beach-1, which spans 150 × 150 pixels. We refer to Tu et al. for more details on the spatial resolution and location of each scene [48].

For this evaluation, we ignore any spatial information and treat each dataset as a bag-of-pixels, with each pixel representing a data vector. Prior to applying each method, the hyperspectral pixels are first divided by their $l_2$ norm to give each a unit length, a common preprocessing step for HSI data [49]. This step removes the relative intensity information so that anomaly detection is based only on spectral signatures. The vector is then fed into an AE: $\text{FC}(D, 100) - \text{FC}(100, 50) - \text{FC}(50, 25)$ to $\text{FC}(25, 50) - \text{FC}(50, 100) - \text{FC}(100, D)$, where $D$ is the pixel vector dimensionality. This architecture and learning rate of $1e^{-3}$ is based on the AE feature extractor network used by Windrim et al. [45]. Again, due to the larger size, the number of steps per epoch is set to 100. Three runs using the airport-4 dataset are used for setting the MSE-AE stopping epoch at 9 and the MTS-AE parameter to $M = 4.2$.

We compare the standard MSE-AE and our MTS-AE to a baseline algorithm widely used for hyperspectral anomaly detection, global RX (GRX), as introduced by Reed and Yu [50]. GRX produces anomaly scores by generating a multi-variate Gaussian model with a mean vector, $\mu$, and covariance matrix, $\Sigma$, from all the pixels in a scene. The algorithm then calculates an anomaly score for each pixel $x$ from the Mahalonobis distance,

$$D_{RX}(x) = (x - \mu)^T \Sigma^{-1} (x - \mu)$$  \hspace{1cm} (5)

This method often yields acceptable results but relies on the assumption that the spectral band values are Gaussian distributed, which is frequently not the case. The primary benefit of using GRX in the unsupervised setting is that it does not require parameter selection.

We compare our methods to a baseline algorithm widely used for hyperspectral anomaly detection, global RX (GRX), as introduced by Reed and Yu [50]. GRX produces anomaly scores by generating a multi-variate Gaussian model with a mean vector, $\mu$, and covariance matrix, $\Sigma$, from all the pixels in a scene. The algorithm then calculates an anomaly score for each pixel $x$ from the Mahalonobis distance,

$$D_{RX}(x) = (x - \mu)^T \Sigma^{-1} (x - \mu)$$  \hspace{1cm} (5)

This method often yields acceptable results but relies on the assumption that the spectral band values are Gaussian distributed, which is frequently not the case. The primary benefit of using GRX in the unsupervised setting is that it does not require parameter selection.

We compare our methods to a baseline algorithm widely used for hyperspectral anomaly detection, global RX (GRX), as introduced by Reed and Yu [50]. GRX produces anomaly scores by generating a multi-variate Gaussian model with a mean vector, $\mu$, and covariance matrix, $\Sigma$, from all the pixels in a scene. The algorithm then calculates an anomaly score for each pixel $x$ from the Mahalonobis distance,

$$D_{RX}(x) = (x - \mu)^T \Sigma^{-1} (x - \mu)$$  \hspace{1cm} (5)

This method often yields acceptable results but relies on the assumption that the spectral band values are Gaussian distributed, which is frequently not the case. The primary benefit of using GRX in the unsupervised setting is that it does not require parameter selection.

We compare our methods to a baseline algorithm widely used for hyperspectral anomaly detection, global RX (GRX), as introduced by Reed and Yu [50]. GRX produces anomaly scores by generating a multi-variate Gaussian model with a mean vector, $\mu$, and covariance matrix, $\Sigma$, from all the pixels in a scene. The algorithm then calculates an anomaly score for each pixel $x$ from the Mahalonobis distance,

$$D_{RX}(x) = (x - \mu)^T \Sigma^{-1} (x - \mu)$$  \hspace{1cm} (5)

This method often yields acceptable results but relies on the assumption that the spectral band values are Gaussian distributed, which is frequently not the case. The primary benefit of using GRX in the unsupervised setting is that it does not require parameter selection.

We compare our methods to a baseline algorithm widely used for hyperspectral anomaly detection, global RX (GRX), as introduced by Reed and Yu [50]. GRX produces anomaly scores by generating a multi-variate Gaussian model with a mean vector, $\mu$, and covariance matrix, $\Sigma$, from all the pixels in a scene. The algorithm then calculates an anomaly score for each pixel $x$ from the Mahalonobis distance,

$$D_{RX}(x) = (x - \mu)^T \Sigma^{-1} (x - \mu)$$  \hspace{1cm} (5)

This method often yields acceptable results but relies on the assumption that the spectral band values are Gaussian distributed, which is frequently not the case. The primary benefit of using GRX in the unsupervised setting is that it does not require parameter selection.
TABLE 4. The average AUROC (left) and AP (right) on nine HSI datasets. The * indicates the tuning set.

| scene       | α     | GRX        | MSE-AE | MTS-AE       |
|-------------|-------|------------|--------|--------------|
| airport-4*  | 0.6%  | 0.900 ± 0.170 | 0.954 ± 0.511 | 0.934 ± 0.517 |
| airport-1   | 1.44% | 0.589 ± 0.017 | 0.752 ± 0.027 | 0.783 ± 0.032 |
| airport-2   | 0.87% | 0.722 ± 0.026 | 0.845 ± 0.028 | 0.876 ± 0.034 |
| airport-3   | 1.70% | 0.457 ± 0.032 | 0.727 ± 0.067 | 0.863 ± 0.067 |
| beach-1     | 0.08% | 0.976 ± 0.285 | 0.903 ± 0.007 | 0.944 ± 0.009 |
| beach-2     | 2.02% | 0.927 ± 0.174 | 0.886 ± 0.078 | 0.899 ± 0.088 |
| beach-3     | 0.11% | 0.999 ± 0.874 | 0.921 ± 0.019 | 0.962 ± 0.053 |
| urban-1     | 0.69% | 0.898 ± 0.036 | 0.958 ± 0.074 | 0.977 ± 0.458 |
| urban-2     | 1.55% | 0.694 ± 0.135 | 0.949 ± 0.559 | 0.992 ± 0.798 |
| urban-3     | 0.52% | 0.650 ± 0.007 | 0.907 ± 0.029 | 0.992 ± 0.471 |

(a) False Color (b) MTS-AE

FIGURE 7. A run from the beach-3 dataset highlights a potential problem of the MTS-AE. PL prevents the rare, but normal mixed pixels along the shoreline from being learned during training.

V. CONCLUSION

Conventional autoencoders (AE) used for unsupervised anomaly detection are prone to over-generalize to the anomalies present in the training data. This reduces the ability of AEs to identify abnormal data based on measures of reconstruction error. To address this shortcoming, we propose several novel methods, namely cumulative error scoring (CES), percentile loss (PL), and early stopping via knee detection. CES leverages the history of training errors to better separate anomalous and background points. PL diminishes the influence of anomalies on parameter updates, undermining the ability of AEs to generalize anomalous examples. Lastly, we show how the smooth cumulative loss statistic provides a reliable means of early stopping. In a number of evaluations on image, cybersecurity, and remote sensing data, we show considerable improvement in both detection accuracy and robustness across a number of applications. Notably, the techniques presented can be readily applied on top of other existing deep anomaly detection architectures and methods.

However, the MTS-AE presented in this work does not yield a final trained model that can be applied to new unseen data. Instead, the MTS-AE can only be applied to identifying anomalies as part of a full dataset during the act of training. One possible, but costly solution is to retain the historical states of the model during training and evaluating new examples with each of these states. A possible avenue of future research is to incorporate the training history into a single model instance, perhaps through some means of distillation [52].

REFERENCES

[1] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, “Detection of anomalies in large scale accounting data using deep autoencoder networks,” 2017, arXiv:1709.05254. [Online]. Available: http://arxiv.org/abs/1709.05254
[2] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, “A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study,” Sensors, vol. 18, no. 8, p. 2491, Aug. 2018.
[3] M. Goldstein and S. Uchida, “A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data,” PLoS ONE, vol. 11, no. 4, Apr. 2016, Art. no. e0152173.
[4] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” 2019, arXiv:1901.03407. [Online]. Available: http://arxiv.org/abs/1901.03407
[5] V. Škvára, T. Pevný, and V. Šmíd, “Are generative deep models for novelty detection truly better?” 2018, arXiv:1807.05027. [Online]. Available: http://arxiv.org/abs/1807.05027
[6] D. D’Avino, D. Cozzolino, G. Poggi, and L. Verdoliva, “Autoencoder with recurrent neural networks for video forgery detection,” 2017, arXiv:1708.08754. [Online]. Available: http://arxiv.org/abs/1708.08754
[7] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” 2016, arXiv:1604.04574. [Online]. Available: http://arxiv.org/abs/1604.04574
[8] M. Ribeiro, A. E. Lazzarette, and H. S. Lopes, “A study of deep convolutional auto-encoders for anomaly detection in videos,” Pattern Recognit. Lett., vol. 105, pp. 13–22, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865517302489
[9] L. Beggela, M. Pfeiffer, and B. Bischl, “Robust anomaly detection in images using adversarial autoencoders,” Jan. 2019, arXiv:1901.06355. [Online]. Available: https://arxiv.org/abs/1901.06355
[10] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, “Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection,” 2019, arXiv:1904.02639. [Online]. Available: http://arxiv.org/abs/1904.02639
[11] N. Merrill and C. C. Olson, “A new autoencoder training paradigm for unsupervised hyperspectral anomaly detection,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Jul. 2020.
[12] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi, “A survey on GANs for anomaly detection,” 2019, arXiv:1906.11632. [Online]. Available: http://arxiv.org/abs/1906.11632
[13] T. Daniel, T. Kurutach, and A. Tamar, “Deep variational semi-supervised novelty detection,” 2019, arXiv:1911.04971. [Online]. Available: https://arxiv.org/abs/1911.04971
[14] J. T. Andrews, E. J. Morton, and L. D. Griffin, “Detecting anomalous data using auto-encoders,” Int. J. Mach. Learn. Comput., vol. 6, p. 21, Feb. 2016.
[15] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, “Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance,” Dec. 2018, arXiv:1812.02765. [Online]. Available: https://arxiv.org/abs/1812.02765
[16] M. S. Minhas and J. Zelek, “Anomaly detection in images,” 2019, arXiv:1905.13147. [Online]. Available: http://arxiv.org/abs/1905.13147
[17] E. Norlander and A. Sopapasakis, “Latent space conditioning for improved classification and anomaly detection,” 2019, arXiv:1911.10599. [Online]. Available: http://arxiv.org/abs/1911.10599
[18] R. Lukas, R. A. Vandermeulen, N. Gørnitz, A. Binder, E. Müller, and M. Kloft, “Deep support vector data description for unsupervised and semi-supervised anomaly detection,” in Proc. ICML Workshop Uncertainty Robustness Deep Learn., 2019.
[19] L. Ruff, R. Vandermeulen, N. Goenitz, M. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, “Deep support vector data description for unsupervised and semi-supervised anomaly detection,” in Proc. ICML Workshop Uncertainty Robustness Deep Learn., 2019.
[20] L. Xiong, B. Póczos, and J. Schneider, “Group anomaly detection using flexible genre models,” in Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS), Red Hook, NY, USA: Curran Associates, 2011, pp. 1071–1079.
Y. Zhao, Z. Nasrullah, and Z. Li, “PyOD: A Python toolbox for scalable anomaly detection,” in Proc. 10th Workshop Hyperspectral Imag. Signal Process., Evol. Remote Sens., Sep. 2019, pp. 1–5.

[45] L. Windrim, R. Ramakrishnan, A. Melkumyan, R. J. Murphy, and A. Chilingaryan, “Unsupervised feature-learning for hyperspectral data with autoencoders,” Remote Sens., vol. 11, no. 7, p. 864, Apr. 2019.

[46] C. Zhao, X. Li, and H. Zhu, “Hyperspectral anomaly detection based on stacked denoising autoencoders,” J. Appl. Remote Sens., vol. 11, no. 4, p. 1, Sep. 2017.

[47] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, “The airborne visible/infrared imaging spectrometer (AVIRIS),” Remote Sens. Environ., vol. 44, nos. 2–3, pp. 127–143, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/003442579390012M

[48] 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, Nov. 2019.

[49] N. Audebert, B. Le Saux, and S. Lefevre, “Deep learning for classification of hyperspectral data: A comparative review,” IEEE Geosci. Remote Sens. Mag., vol. 7, no. 2, pp. 159–173, Jun. 2019.

[50] I. S. Reed and X. Yu, “Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution,” IEEE Trans. Acoust., Speech, Signal Process., vol. 38, no. 10, pp. 1760–1770, Oct. 1990.

[51] T. H. Emerson, T. Doster, and C. C. Olson, “Path-based background model augmentation for hyperspectral anomaly detection,” in Proc. 9th Workshop Hyperspectral Imag. Signal Process. Evol. Remote Sens. (WHISPERS), Sep. 2018, pp. 1–5.

[52] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, arXiv:1503.02531. [Online]. Available: https://arxiv.org/abs/1503.02531

NICHOLAS MERRill received the B.Sc. degree in physics and the M.A. degree in education from Virginia Tech (VT), in 2012 and 2013, respectively, where he is currently pursuing the M.Sc. degree in mechanical engineering with the Autonomous Systems and Intelligent Machines Laboratory (ASIM). He worked as a High School Physics Instructor for several years before returning to VT. His current research interest includes unsupervised anomaly detection.

AZIM ESKANDARIAN (Senior Member, IEEE) received the B.S. degree from The George Washington University (GWC), the M.S. degree from Virginia Tech (VT), and the D.Sc. degree from GWU, all in mechanical engineering. He has been a Professor and the Head of the Department of Mechanical Engineering, VT, since August 2015. He became the Nicholas and Rebecca Des Champs Chaired Professor, in April 2018. He established the Autonomous Systems and Intelligent Machines Laboratory, VT, to conduct research in intelligent and autonomous vehicles and mobile robotics. Prior to that, he was a Professor of engineering and applied science with GWU and the Founding Director of the Center for Intelligent Systems Research, from 1996 to 2015; the Director of the Transportation Safety and Security University Area of Excellence, from 2002 to 2015; and the Co-Founder of the National Crash Analysis Center, in 1992, where he was the Director from 1998 to 2002 and from May 2013 to July 2015. Earlier, he was an Assistant Professor with Pennsylvania State University, York, PA, USA, from 1989 to 1992, and worked as an Engineer/Project Manager in industry, from 1983 to 1989.

Dr. Eskandarian is a fellow of ASME and a member of SAE professional societies. He was awarded the IEEE ITS Society’s Outstanding Researcher Award, in 2017, and the GWUs School of Engineering Outstanding Researcher Award, in 2013. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.