Hybrid Discriminator With Correlative Autoencoder for Anomaly Detection

JUNGEON LEE, MUHAMMAD UMAR KARIM KHAN, AND CHONG-MIN KYUNG, (Life Fellow, IEEE)

1Division of Future Vehicle, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea
2Center for Integrated Smart Sensors (CISS), Daejeon 34141, South Korea
3School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea

Corresponding author: Jungeon Lee (minoreon@kaist.ac.kr)

This work was supported in part by the Center for Integrated Smart Sensors funded by the Ministry of Science and ICT as Global Frontier Project (CISS-2013M3A6A6073718), and in part by SK Hynix Inc.

ABSTRACT Advances in deep neural networks (DNNs) have led to impressive results and in recent years many works have exploited DNNs for anomaly detection. Among others, generative/reconstruction model-based methods have been frequently used for anomaly detection because they do not require any labels for training. The anomaly detection performance of these methods, however, varies a lot, due to the change of the intra-class variance and the difference in complexity of input samples. In addition, most previous state-of-the-art works on anomaly detection have empirically adjusted several hyperparameters to heighten their performance of anomaly detection. These sorts of procedures are known to be impractical and create obstacles in real world anomaly detection. To solve these problems, we propose a hybrid discriminator with a correlative autoencoder for anomaly detection. In the proposed framework, the discriminator implicitly estimates the conditional probability density function and the autoencoder has improved ability to control the reconstruction error. We provide theoretical foundation of our method and verify it through various experiments. We also confirm practical benefits of our interpretation of the conditional expectation and the proposed framework by comparing our results with other state-of-the-art methods.

INDEX TERMS Intelligent systems, machine intelligence, neural networks, semisupervised learning, unsupervised learning.

I. INTRODUCTION

The recent explosive growth in machine learning and big data applications has increased the need for techniques that can detect anomalies. During machine learning, samples that are observed during training are typically called inliers, in-distribution or normal samples, while samples not observed during training are called outliers, out-of-distribution (OOD), novel, or abnormal/anomalous samples. Except for comparing with references, in the remaining sections we use the term “in-distribution” for samples observed during training, and use the term “OOD” for samples not observed during training.

The purpose of anomaly detection is pretty simple: to distinguish OOD samples from in-distribution samples. However, the execution can be fairly complex. Some previous works on anomaly detection have assumed there is only one class of in-distribution samples. From that perspective, if all numerical samples labeled as ‘0’ are in-distribution, then all other numerical samples are OOD. This type of problem has been called novelty/outlier detection or one-class classification (OCC). OOD detection was introduced in other works [11]–[13]. In OOD detection, the in-distribution and OOD samples are from different data sets. For instance, if facial samples are in-distribution, then numerical samples are OOD. Another way of categorizing anomaly detection is to consider multiple class samples within the same data set as either in-distribution or OOD. For instance, if numerical samples labeled as ‘0’, ‘1’, and ‘2’ are in-distribution, then the remaining numerical samples are OOD. So far no one has reported experimental results under this setting but we have found that increasing the number of classes for in-distribution samples dramatically decreases the anomaly detection performance, especially in previous generative/reconstruction model-based approaches. It can be presumed that using multiple classes for in-distribution samples without any
labels will increase the intra-class variance of in-distribution samples.

Most previous works detect anomalies by mapping higher-dimensional samples to lower-dimensional space. In this way, if the intra-class variance of samples increases, distributions of in-distribution samples and OOD samples can overlap or get closer to each other in lower-dimensional space. As a result, if the intra-class variance of samples increases, the anomaly detection performance significantly decreases.

Generative/reconstruction model-based approaches [14]–[16] for OOD detection have an advantage because they do not require any labels for training. However, instances [1], [5], [19] have been found where the probability density likelihood of untrained OOD samples is higher than that of trained in-distribution samples, which are inconsistent with the idea [17], [18] that generative models can be used for anomaly detection. For instance, Glow [14], a state-of-the-art normalizing flow, trained on FashionMNIST or CIFAR10 assigns the higher likelihood to untrained MNIST or SVHN than to its trained FashionMNIST or CIFAR10. Unfortunately, in generative models, it has been experimentally observed that the probability density likelihood does not only depend on whether or not samples are trained but also on the difference in complexity of input samples (input complexity) [6], [7]. Estimating the probability density is still an open issue, which is a significant factor of the anomaly detection performance of generative model-based approaches.

In order to solve the performance degradation problem due to the change of the intra-class variance and the difference in input complexity, we propose a framework for anomaly detection which is capable of conditionally discriminating and selectively controlling the reconstruction error by using a hybrid discriminator with a correlative autoencoder. We focus on the anomaly detection problem which uses only in-distribution samples without any labels during training, and verify the proposed method through experiments for OCC, OOD detection, and multiple class anomaly detection (MCAD). In particular, the proposed method does not use any data set-dependent empirical hyperparameters except for essential hyperparameters such as the size of mini-batch or learning rate.

The rest of the paper proceeds in the following order. First, we investigate previous works for OCC and OOD detection in Section II. In Section III, we explain details of the proposed method including the background, the proposed framework, a hybrid discriminator, a correlative autoencoder, and the procedure for training and testing. In Section IV, we provide details of the experiments including experimental setup, experimental results, ablation study, and analysis/discussion. Finally, we conclude the paper in Section V.

II. RELATED WORKS

We categorize previous DNN-based approaches for anomaly detection according to whether they require labels for training or not. Classifier-based approaches [1]–[4], [11], [12] require labels for training, while self-supervision-based approaches [13], [20] generate pseudo-labels for samples and perform pre-text tasks for training. Generative model-based approaches [1], [5]–[7] and reconstruction model-based approaches [8]–[10] do not require labels for training. In this section, we mention only closely related works to our research. For a broader survey on methodologies and applications for OCC and OOD detection, we recommend the reader to refer to [21], [22].

In OOD detection, generative/reconstruction model-based approaches have a great advantage because they do not require labels for training but they suffer from the performance degradation problem as shown in Table 1.

| TABLE 1. Advantages and disadvantages of previous and our methods for anomaly detection. |
|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| **Advantages**                    | **Disadvantages**                 | **Advantages**                    | **Disadvantages**                 |
| Classifier-based Approaches [1][2][3][4] | • The anomaly detection performance degrades less due to the change of the intra-class variance and the difference in complexity of input samples (input complexity). | • Class labels are required for training. | • Class labels are required for training. |
| Generator Model-based Approaches [1][5][6][7] | • No class labels are required for training. | • Use a generative model whose network architecture is relatively simple. | • The anomaly detection performance significant degraded due to the difference of input complexity and recent works have been trying to solve it. |
| Reconstruction Model-based Approaches [8][9][10] | • No class labels are required for training. | • Use a reconstruction model whose network architecture is relatively simple. | • The anomaly detection performance significantly degrades due to the change of the intra-class variance and the difference in input complexity. |
| Ours                               | • No class labels are required for training. | • The anomaly detection performance significantly degrades due to the change of the intra-class variance and the difference in input complexity. | • The anomaly detection performance degrades due to the small inter-class variance. |
Most previous generative/reconstruction model-based approaches improve the performance degradation problem resulting from input complexity in the following ways.

- They distinguish OOD samples from in-distribution samples by estimating the Watanabe-Akaike information criterion (WAIC) of the samples [5].
- They first separate image samples into background components and semantic components, and then distinguish OOD samples from in-distribution samples using the likelihood ratio of these components [6].
- They measure input complexity using the compression tool, and distinguish OOD samples from in-distribution samples based on the measured values [7].

Unlike OOD detection, in OCC samples from a certain single class are considered as in-distribution and commonly the rest of the samples are considered as OOD within the same data set. Labels are not given in OCC, and so a generative/reconstruction model which generates more reconstruction ability in OOD samples than in-distribution samples is often required. Most previous generative/reconstruction model-based approaches for OCC used the following ways to increase the anomaly detection performance.

- The probability distribution of in-distribution samples is trained to follow a predetermined probability distribution through an adversarial autoencoder (AAE) [23]. A discriminator discriminates OOD samples from in-distribution samples, which implicitly estimates the probability distribution of reconstructed samples during testing [9].
- Reconstructed samples are generated through an autoencoder (AE) trained with the mean squared error (MSE) loss and the log-likelihood loss. Both the likelihood of autoregression [24], [25] and the MSE [10] are used to discriminate OOD samples from in-distribution sample during testing.

However, the anomaly detection performance of previous generative/reconstruction model-based approaches [8]–[10] degrades a lot when samples from multiple class within the same data set are used as either in-distribution or OOD. In other words, the anomaly detection performance of previous generative/reconstruction model-based approaches are significantly affected by the intra-class variance.

III. PROPOSED METHOD

Here, we provide a high-level analysis of the proposed method of by comparing them with previous methods. We present the overall schema of previous and our methods in Fig. 1.

A. BACKGROUND

It can be seen in Fig. 1(a) and 1(b) that most of the previous generative/reconstruction model-based approaches for anomaly detection use a discriminator (D), and a generator (G) as an AE which consists of an encoder (Ge) and decoder (Gd) since GAN [26] has proposed. They pit G and D against each other. G is trained to reconstruct an output G(x) similar to an input sample x which can fool D. D is trained to distinguish the real sample x and the generated (reconstructed) G(x) from each other. As training proceeds, both D and G improve each other, ultimately resulting in G reconstructing realistic samples which satisfy pd(x) = pg(G(x)), where pd(x) is a probability density function (PDF) for real samples x, and pg(G(x)) is a PDF for generated samples G(x). In most previous generative/reconstruction model-based approaches for anomaly detection only in-distribution samples are available to train given models, and G introduces reconstruction errors for OOD samples during testing. They considered that D can distinguish the difference between the distribution of OOD samples and the distribution of in-distribution samples because pd(x) ≠ pg(G(x)) for OOD samples. D discriminates OOD samples from in-distribution samples by observing the reconstruction ability of G(x).

To exploit generative/reconstruction model for anomaly detection, the previous methods simultaneously applied more than one auxiliary non-parametric loss, such as the mean square error (MSE) and/or the cross entropy (CE) loss, into the log-likelihood loss of GAN because G from original GAN cannot reconstruct G(x) directly from real samples x. These non-parametric losses result in less reconstruction error than the log-likelihood loss of GAN. Therefore, the ratios between these non-parametric losses and the log-likelihood loss have to be adjusted empirically in order to change the reconstruction ability of G(x). Consequently, D implicitly estimates the reconstruction error of G(x) to detect anomalies.

Previously, the hyperparameters used to control the ratio between these non-parametric losses and the log-likelihood...
loss have heavily affected the anomaly detection performance. As can be seen in Fig. 1(a) and 1(b), the previous methods controlled the reconstruction error for \( G(x) \) by empirically adjusting the hyperparameters. Some works [8], [9] measured the reconstruction error using the MSE, while [10] estimated the probability density of latent vectors \( z \) using autoregression (AR). In this way, the MSE and AR were used to detect anomalies. Through our experiments we found that these generative/reconstruction model-based approaches are vulnerable to not only input complexity of samples but also to the large intra-class variance.

**B. PROPOSED FRAMEWORK**

The proposed anomaly detection framework is composed of two main modules, a Hybrid Discriminator (HD), and a Correlative AutoEncoder (CAE) as can be seen in Fig. 1(c). The former performs the discrimination (detection) process, while latter performs the generation (reconstruction) process. These two networks are trained in an adversarial and unsupervised manner within an end-to-end setting. Rather than using the previous discriminators in Fig. 1(a) and 1(b), we combined a conditional discriminator \( D_2(x, \cdot) \) and the original discriminator \( D_1(\cdot) \) from GAN [26] to propose the HD.

The proposed CAE (G) is composed of two encoders \( G_{E1}, G_{E2} \), and a decoder \( G_D \). In the proposed G, \( G_{E1} \) is trained to be adversarial in response to \( D_1(\cdot) \), and \( G_{E2} \) is trained to be adversarial in response to \( D_2(x, \cdot) \). \( G_D \) is trained to be adversarial in response to both \( D_1(\cdot) \) and \( D_2(x, \cdot) \). Consequently, we can see that \( G_{E1}, G_D \), and \( D_1(\cdot) \) form the first adversarial network, while \( G_{E2}, D_2(x, \cdot) \) form the second adversarial network.

The proposed conditional discriminator \( D_2(x, \cdot) \) estimates the conditional probability density of \( x \) and \( G(x) \) under the condition that a real sample \( x \) is given. In the next section, we will show that \( D_2(x, \cdot) \) not only replace the function of the MSE in Fig. 1(a) and 1(b), but also distinguishes OOD from in-distribution samples. \( D \) in Fig. 1(c) discriminates OOD samples from in-distribution samples by implicitly estimating the probability density of \( G(x) \). On the other hand, \( D_2(x, \cdot) \) in Fig. 1(c) discriminates OOD samples from in-distribution samples by implicitly estimating the conditional probability density of \( G(x) \) given \( x \). Also, \( D_2(x, \cdot) \) is estimated relatively according to input complexity of samples. The CAE produces generated (reconstructed) samples \( G(x) \) considering the correlation between the outputs of \( G_{E1} \) and \( G_{E2} \).

**C. HYBRID DISCRIMINATORS**

The proposed hybrid discriminator (HD) consists of two different discriminators; one is the discriminator \( D_1(\cdot) \) as in [26], and the other one is the conditional discriminator \( D_2(x, \cdot) \). A previous work [27] suggested the use of the conditional discriminator to conditionally generate an image. Our \( D_2(x, \cdot) \) is somewhat similar to the conditional discriminator model in previous work but unlike them \( D_2(x, \cdot) \) here has no relation with conditional generation. We combine \( D_2(x, \cdot) \) and \( D_1(\cdot) \) into the HD to detect anomalies, as well as implicitly control the reconstruction of \( G \) during training.

We first define the minimax objective for \( D_1(\cdot) \) and \( G \). Since only \( G_{E1} \) and \( G_D \) are adversarially trained with \( D_1(\cdot) \), we denote the generator model for \( D_1(\cdot) \) as \( G_1(x; \theta_{G_{E1}}, \theta_{G_D}) \), where \( G_1 \) is a differentiable function represented by a multi-layer perceptron with parameters \( \theta_{G_{E1}} \) and \( \theta_{G_D} \).

\[
\min_{G_1} \max_{D_1} \mathbb{E}_{x \sim p_{x}(x)} \left[ \log D_1(x) \right] + \mathbb{E}_{G(z) \sim p_{G}(z)} \left[ \log(1 - D_1(G(z))) \right].
\]

Eq. (1) is identical to replacing \( D(G(z)) \) of the minimax objective from GAN [26] with \( D_1(G(x)) \). Thus, \( D_1(\cdot) \) and \( G_1 \)
reach the global optimum when \( p_{data}(x) = p_{d}(G(x)) \), where \( p_{data}(x) \) is the PDF for given data samples \( x \), and \( p_{d}(G(x)) \) is the PDF for generated samples \( G(x) \).

Since only \( G_{E2} \) and \( D_{G} \) are adversarially trained with \( D_{2}(x, \cdot) \), we denote the generator model for \( D_{2}(x, \cdot) \) as \( G_{2}(x; \theta_{G_{E2}}, \theta_{D_{G}}) \), where \( G_{2} \) is a differentiable function represented by a multilayer perceptron with parameters \( \theta_{G_{E2}} \) and \( \theta_{D_{G}} \).

\[
\min_{G_{2}} \max_{D_{2}} L_{2}(D_{2}, G_{2}) = E_{x \sim p_{data}(x)} \left[ \log D_{2}(x, \cdot) \right] + E_{G(x) \sim p_{d}(G(x))} \left[ \log (1 - D_{2}(G(x), \cdot)) \right], \tag{2}
\]

In Eq. (2), \( p_{d}(x) \) is the conditional PDF of \( x \) given \( G(x) \), and \( p_{d}(G(x)) \) is the conditional PDF of \( G(x) \) given \( x \). We found that both \( D_{2}(x, \cdot) \) and \( G_{2} \) reach the global optimum when \( p_{d}(x) = p_{d}(G(x)) \). Since real samples \( x \) are i.i.d. and sample \( x \) is given to \( G(x) \) as a condition, \( x = G(x) \) must be satisfied to satisfy \( p_{d}(x) = p_{d}(G(x)) \). Please refer to the Appendix A for further details including proof.

\( D_{2}(x, \cdot) \) distinguishes between \( x \) and \( G(x) \) by using the conditional PDF \( (p_{d}(G(x))|x) \) and not the PDF \( (p_{d}(G(x))) \). By using \( x \) in \( D_{2}(x, \cdot) \), we expect that anomaly detection using \( D_{2}(x, \cdot) \) is better than anomaly detection using \( D_{1}(\cdot) \), especially when samples are diverse and/or complex. \( D_{2}(x, \cdot) \) is not affected by the number of classes for samples unlike \( D_{1}(\cdot) \) because the real sample \( x \) is always given as a condition. Because \( D_{2}(x, \cdot) \) compares the PDF of the real sample \( x \) and the generated sample \( G(x) \) relatively, \( D_{2}(x, \cdot) \) is less affected by input complexity.

The MSE between \( x \) and \( G(x) \) trained with \( D_{2}(x, \cdot) \) can be expressed as \( E[(x - G(x))|x]^{2} \) [28] because \( D_{2}(x, \cdot) \) learns the conditional PDF of samples. On the other hand, the MSE between \( x \) and \( G(x) \) trained with \( D_{1}(\cdot) \) is \( E[(x - G(x))|x]^{2} \) because \( D_{1}(\cdot) \) learns the PDF of the samples. Since \( E[(x - G(x))|x]^{2} \geq E[(x - G(x))|x]^{2} \) [29], the reconstruction error of \( G(x) \) trained with \( D_{1}(\cdot) \) is greater than that of \( G(x) \), or rather, \( G(x)|x \) trained with \( D_{2}(x, \cdot) \). In other words, the variance of real samples within mini-batch is eliminated for the conditional PDF (conditional expectation) because a real sample \( x \) is given for \( G(x) \). Moreover, the MSE in Fig. 1(a) compares \( G(x) \) and \( x \) in pixel units with \( x \), as in \( D_{2}(x, \cdot) \), so the MSE of \( x \) and \( G(x) \) is \( E[(x - G(x))|x]^{2} \). Therefore, we can use \( D_{2}(x, \cdot) \) instead of the MSE in Fig. 1(a).

\( D_{2}(x, \cdot) \) can implicitly estimate not only the conditional PDF of \( G(x) \) but also that of \( x \). While \( D_{2}(x, G(x)) \) is trained with adversarial learning, \( D_{2}(x, x) \) is trained with the maximum log-likelihood as is a generative model which explicitly estimates probability density. Although \( D_{2}(x, \cdot) \) implicitly estimates the probability density, \( D_{2}(x, x) \) can relatively measure the difference in input complexity. When input complexity of the samples is similar to each other, \( D_{2}(x, x) \) for in-distribution samples is higher than that for OOD samples like generative models such as Glow [14] or PixelCNN++ [15]. Accordingly, \( D_{2}(x, \cdot) \) basically functions as discriminator of GAN, \( D_{2}(x, G(x)) \) discriminates OOD samples from in-distribution samples, and \( D_{2}(x, x) \) can be used to measure input complexity.

\section*{D. Correlative Autoencoder}

As we would like \( G \) to increase reconstruction errors for OOD samples regardless of input complexity, we propose a correlative autoencoder (CAE) which cooperatively works with the HD. The CAE consists of two encoders \( G_{E1}, G_{E2} \), and a decoder \( G_{D} \). In the proposed CAE, \( G_{E1} \) is trained to be adversarial in response to \( D_{1}(\cdot) \), and \( G_{E2} \) is trained to be adversarial in response to both \( D_{1} \) and \( D_{2} \). \( G_{E1} \) and \( G_{E2} \) are adversarially trained with \( D_{1}(\cdot) \) and \( G_{D} \). We use the Pearson correlation to both \( z_{1} \) and \( z_{2} \) because we would like them to be consistent regardless of the data sets. Thus, \( z_{1} \) and \( z_{2} \) will be normalized as follows.

\[
\begin{align*}
z_{N} &= z_{N12} + z_{N2}, \\
where \quad z_{N12} &= \frac{(z_{1} - E[z_{1}])(z_{2} - E[z_{2}])}{\sigma_{z_{1}} \sigma_{z_{2}}}, \\
and \quad z_{N2} &= \left(\frac{z_{2} - E[z_{2}]}{\sigma_{z_{2}}}\right)^{2}, \tag{3}
\end{align*}
\]

For \( E[z_{1}] \) and \( E[z_{2}] \), we use the mean of \( z_{1} \) and \( z_{2} \) from the training mini-batch, respectively. In the same way, we use the standard deviations of \( z_{1} \) and \( z_{2} \) from the training mini-batch for \( \sigma_{z_{1}} \) and \( \sigma_{z_{2}} \). This policy will not change, either during training or testing. In other words, all of the statistical parameters in Eq. (3) are only set during training. The CAE generates \( G(x) \) whose varying reconstruction error depends on whether \( x \) was observed during training or not. To match the order of \( z_{N12}, z_{2} \) is normalized as \( z_{N2} \) using expectation and variance of \( z_{2} \). We already know the reconstruction error of \( z_{2} \) is less than that of \( z_{1} \). Thus, the reconstruction error of \( z_{N2} \) will be less than that of \( z_{N12} \).

If \( G_{D} \) uses only \( z_{N12} \), rarely seen samples during training will introduce large reconstruction errors, and this will lead to false negatives during testing. On the other hand, if \( G_{D} \) only uses \( z_{N2} \), generalization becomes too high, which leads false positives during testing. By using \( z_{N} \) as \( z_{N12} + z_{N2} \), \( z_{N} \) maintains basic reconstruction functions of \( z_{N2} \), and at the same time it introduces a reconstruction error from \( z_{N12} \), which varies depending on whether \( x \) was observed during
Algorithm 1 Training the Proposed Framework

for $k$ steps do

- Sample $m$ samples of mini-batch $\{x_1, \ldots, x_m\}$ from a training samples set.
- Update $\theta_{D_2(x, \cdot)}$ (parameters of $D_2(x, \cdot)$) by ascending its stochastic gradient:
  \[
  \nabla_{\theta_{D_2(x, \cdot)}} \frac{1}{m} \sum_{i=1}^{m} \left[ \log(D_2(x_i, G(x_i))) + \log(1 - D_2(x_i, G(x_i))) \right].
  \]
- Update $\theta_{G_E2}$ (parameters of $G_{E2}$) and $\theta_{G_D}$ (parameters of $G_D$) by ascending its stochastic gradient:
  \[
  \nabla_{\theta_{G_E2}, \theta_{G_D}} \frac{1}{m} \sum_{i=1}^{m} \left[ \log(D_2(x_i, G(x_i))) \right].
  \]
- Update $\theta_{D_1(\cdot)}$ (parameters of $D_1(\cdot)$) by ascending its stochastic gradient:
  \[
  \nabla_{\theta_{D_1(\cdot)}} \frac{1}{m} \sum_{i=1}^{m} \left[ \log(D_1(x_i)) + \log(1 - D_1(G(x_i))) \right].
  \]
- Update $\theta_{G_E1}$ (parameters of $G_{E1}$) and $\theta_{G_D}$ by ascending its stochastic gradient:
  \[
  \nabla_{\theta_{G_E1}, \theta_{G_D}} \frac{1}{m} \sum_{i=1}^{m} \left[ \log(D_1(G(x_i))) \right].
  \]
- Update $E[z_1], E[z_2], \alpha_{z_1}$ and $\alpha_{z_2}$ for a mini-batch of $m$ normal samples for training.

end

// The gradient-based updates can use any standard gradient-based learning rule.
// We used Adam with betas = (0.9, 0.999), eps = $10^{-8}$, and weight decay = 0.

| Table 2. Data set setting for OCC, MCAD, OOD detection. |
|---------------------------------------------------------|
| The number of classes for in-distribution | One-Class Classification (OCC) | Multiple Class Anomaly Detection (MCAD) | OOD Detection |
|------------------------------------------|--------------------------|--------------------------------|-------------|
| Single | Multiple | Multiple | Multiple |
| The number of classes for OOD | Single/Multiple | Single/Multiple | Multiple |
| Source data set for in-distribution and OOD samples | Identical | Identical | Different |
| Data set for in-distribution samples | MNIST, FashionMNIST, COIL100, CIFAR10 | FashionMNIST | MNIST |
| Data set for OOD samples | MNIST | |

As shown in Eq. (4), the total loss $L(D_1, D_2, G) = \min \max V_1(D_1, G_1) + \min \max V_2(D_2, G_2)$

\[
L(D_1, D_2, G) = \min \max V_1(D_1, G_1) + \min \max V_2(D_2, G_2) \quad G_1 \text{ } D_1 \quad G_2 \text{ } D_2 \tag{4}
\]

As shown in Eq. (4), the total loss $L(D_1, D_2, G)$ is the sum of Eq. (1) and Eq. (2). In the proposed framework, note that the parameters of $G_{E2}$ will not be updated when $G_1$ and $D_1(\cdot)$ are trained. Also, the parameters of $G_{E1}$ will not be updated when $G_2$ and $D_2(x, \cdot)$ are trained. The proposed framework is composed of two adversarial networks, the $G_1$ and $D_1(\cdot)$ networks, and the $G_2$ and $D_2(x, \cdot)$ networks. These two adversarial networks are trained alternatively. We describe the detailed training procedure in Algorithm 1.

During testing, we exploit the proposed conditional discriminator $D_2(x, \cdot)$ for anomaly detection. We know $D_2(x, G(x))$ implicitly estimates the conditional PDF for the generated $G(x)$ under the condition that the real sample $x$ is given. We can expect that $D_2(x, G(x))$ depends on whether sample was observed during training or not, so it can be used to discriminate OOD samples from in-distribution samples using the following policy.

\[
f_{AD} = \begin{cases} 
0 \text{ (OOD)} & \text{if } D_2(x, G(x)) < \lambda_{th} \\
1 \text{ (in-distribution)} & \text{otherwise,}
\end{cases}
\tag{5}
\]

where $\lambda_{th}$ is a threshold for binary classification.

IV. EXPERIMENTS

To evaluate the anomaly detection performance of the proposed method, we compared it with previous methods mentioned in Sec. II and Table 1. We set various test conditions for anomaly detection in Table 2. To prove whether the proposed method eradicates the performance degradation problems of anomaly detection due to differences in input complexity of samples, we use combinations of data sets that have been reported to cause this situation.

A. EXPERIMENTAL ENVIRONMENT

In One-Class Classification (OCC), samples from a single class(or category) are used as in-distribution during training, and samples from another single class or other multiple classes are used as OOD during testing. The rest of the samples except in-distribution samples are used as OOD for OCC on MNIST, FashionMNIST, and CIFAR10. For OCC on COIL100, we followed the experimental setting from [9], [31].

OOD detection evaluates the anomaly detection performance on inter-data sets. We use entirety of a data set for in-distribution samples during training and entirety of another
data set for OOD samples during testing. Commonly, input complexity of in-distribution samples are more complex than that of OOD samples. For instance, using FashionMNIST as in-distribution samples and MNIST as OOD samples is very common for OOD detection.

In multiple class anomaly detection (MCAD), we use multiple classes for in-distribution samples during training, as well as other multiple classes for OOD samples during testing. Both the in-distribution and OOD samples are from the same data set. The number of classes for in-distribution samples and the number of classes for OOD samples are configured to be the same. We randomly select 50% of all classes in a certain data set and allocated them to in-distribution. The remaining 50% are allocated to OOD samples. Only in-distribution samples with no class labels are used during training. Both in-distribution and OOD samples are used to evaluate the anomaly detection performance during testing. Samples for training and testing are completely separated, and so there are no common samples between them. We assumed that both the in-distribution samples used for training and the in-distribution samples used for testing are from the same distribution ($p_d(x)$). We used the area under the receiver operating characteristic (ROC) curve (AUC) as a metric for the anomaly detection performance. Commonly, the anomaly detection performance for MCAD is lower than that for OCC and OOD detection because in the MCAD experiment samples have the larger intra-class variance and the higher input complexity.

Including previous works listed in Table 1 no one has reported experimental results for MCAD so far. Therefore, we test [8]–[10] for MCAD using their source code, and compare their results with ours. The proposed framework is implemented in the PyTorch 0.3.1 environment with multiple NVIDIA GTX 1080Ti and RTX Quadro 8000. If a data set has its own preset for training and testing, we follow the preset. If a data set has no preset for training and testing such as COIL100, we randomly draw samples for training and testing with no duplication. Consequently, we set our experimental setting to be identical to previous works [9], [31] for fair comparisons. We present details of the data sets used for our experiments in Table 2.

Although we confirm that the change of the number of layers (or other arguments such as kernel size, output channel, and etc. for convolution layers) is slightly affected to anomaly detection performance within average range of 1%, we decide to use the same network architectures of which the number of layers are all five in Table 6 for OCC, MCAD, and OOD detection. The proposed framework shows superiority under various anomaly detection scenarios, including OCC, MCAD, and OOD detection on various data sets.

### B. EXPERIMENTAL RESULTS

We conduct experiments for OCC, MCAD, and OOD detection using various works of anomaly detection including ours. Our results in all the tables are the average of at least five test runs. We plotted all the figures by selecting certain AUCs and ROC curves close to the average. Since the difference between OCC and MCAD is the number of classes for OOD samples, we present the comparison of the AUCs for both OCC and MCAD in Table 3. In Table 3, we report the AUCs of previous methods for OCC by referring to their original references and also we report the AUCs of previous methods by using their code [8].
The overall ROC curves of previous and our methods on MNIST can be seen in Fig. 3. The AUCs of the proposed method are all higher than that from other previous methods, regardless of OCC (except FashionMNIST) or MCAD. In particular, since FashionMNIST has many visually similar samples between classes such as FashionMNIST samples labeled as ‘T-shirt/top’ and ‘Shirt’ which have the small inter-class variance, the anomaly detection performance of the proposed method for MCAD on FashionMNIST is higher than that for OCC.

We provide the comparison of the AUCs for OOD detection in Table 4. Results of previous methods are from their original references. We use a relatively complex data set as in-distribution samples; for instance, if FashionMNIST samples are used as in-distribution, then MNIST samples are used as OOD, which is very common for OOD detection. In Table 4, FashionMNIST samples are used as in-distribution, MNIST samples are used as OOD, and the anomaly detection performance degrades due to the difference in input complexity. Note that the previous generative/reconstruction model-based approaches used pre-processing, compression tool, and adjusted hyperparameters to improve their performance but the anomaly detection performance of the proposed method is comparable to that of theirs without using any pre-processing or other minor adjustments. It means that the proposed method can solve the performance degradation problem due to the difference in input complexity.

C. ANALYSIS AND DISCUSSION

As can be seen in Table 3, the proposed method shows high AUCs for OCC and MCAD. In particular, the proposed method exhibits higher AUCs than previous methods for MCAD, which means that the proposed method is less affected by the intra-class variance. In Fig. 3(a), the AUC is the lowest for samples labeled as ‘8’ and we believe samples labeled as ‘8’ have many similar features which other samples also have. In Fig. 3(b), the AUCs are low when samples labeled as ‘T-shirt/top’ and ‘Shirt’ are used as in-distribution because they have the small inter-class variance. In Fig. 3(c), the AUCs are low when samples which have with a complex background are used as in-distribution. In Fig. 3(d) the AUC of the proposed method is much higher than that of previous methods.

In order to verify that the proposed method solved the performance degradation problem due to the intra-class variance and input complexity, we plotted the histogram of $D_2(x, \cdot)$ values and present them in Fig. 4, along with images of $x$ and $G(x)$ in Fig. 5.
The histograms of $D_2(x, x)$ and $D_2(x, G(x))$ of the proposed method for OOD detection are shown in Fig. 4(a) and 4(b) respectively, where FashionMNIST samples are used as in-distribution and MNIST samples are used as OOD. Although input complexity of MNIST is lower than that of FashionMNIST and $D_2(x, \cdot)$ is trained on FashionMNIST samples, it can be seen that Fig. 4(a) $D_2(x, x)$ for MNIST are located near the high points of $D_2(x, x)$ for FashionMNIST. Since $D_2(x, x)$ implicitly estimates the probability density log-likelihood of samples, it barely discriminates OOD samples from in-distribution samples. In Fig. 4(b), however, there is little overlap between $D_2(x, G(x))$ for reconstructed samples of MNIST and $D_2(x, G(x))$ for reconstructed samples of FashionMNIST. The histograms of $D_2(x, x)$ and $D_2(x, G(x))$ of the proposed method for MCAD are shown in Fig. 4(c) and Fig. 4(d) respectively, where MNIST samples from five classes are used as in-distribution and the rest of the MNIST samples are used as OOD. Although $D_2(x, x)$ did not observe any of OOD samples during training, $D_2(x, x)$ for in-distribution and OOD samples have the similar distributions. It means that discriminating OOD samples from in-distribution samples by simple estimation of the probability density log-likelihood of samples is impossible because of the large intra-class variance and the difference in input complexity. In Fig. 4(d), however, $D_2(x, G(x))$ for OOD samples are mostly concentrated near 0. Thus, $D_2(x, G(x))$ can discriminate OOD samples from in-distribution samples.

Fig. 5(a) and 5(b) show visualization of the real samples $x$ and reconstructed samples $G(x)$ for MCAD and OOD detection. (a) MNIST samples labeled as '3', '4', '7', '8', and '9' are used as in-distribution (top) and MNIST samples labeled as '0', '1', '2', '5', and '6' are used as OOD (bottom) (b) FashionMNIST samples are used as in-distribution (top) and MNIST samples are used as OOD (bottom).

**D. ABLATION STUDY**

In order to verify superiority of the proposed CAE and HD, we conduct several experiments that substitute the MSE for $D_2(x, \cdot)$, and substitute an AE for the CAE. The overall experiment results for this ablation study are present in Table 5. If both $G_{E1}$ and $G_{E2}$ in Eq. (4) are replaced on $G_E$, the total loss for the AE with the HD framework will be $V(D_1, D_2, G) = \min \max V_1(D_1, G) + \beta \min \max V_2(D_2, G)$ which is similar to Eq. (4), where $\beta$ is an empirical hyper-parameter to control the reconstruction error. Since the CAE is replaced by the AE, $\beta$ should be set empirically in the way that other SOTA approaches [8]–[10] did. In this ablation study, $\beta$ was empirically set to 0.1 for OCC and MCAD, and 0.02 for OOD detection.

We presented the AUC traces by increasing training steps in Fig. 6 in order to confirm stability of the proposed framework. It can be seen that substituting the CAE for the AE and using $D_2(x, \cdot)$ as the anomaly detector instead of $D_1(\cdot)$ results in the best performance, where the traces remain flat and high across all training steps. Note that the framework composed of the CAE, $D_1(\cdot)$, and the MSE still exhibited decent performance, although it was not as good as the proposed framework including the CAE and the HD. $D_2(x, \cdot)$ acts as the parametric MSE, merely substituting the MSE for $D_2(x, \cdot)$ will not degrade the anomaly detection performance significantly. Once the AE is substituted for the CAE, however, it degrades the anomaly detection performance a lot, and the AUC traces no longer remain flat and high across all training steps.

We presented the estimated Pearson correlation coefficients for in-distribution and OOD samples in Fig. 7, in order
to verify validity of the correlations for $\mathbf{z}_{N12}$ which was discussed in Sec. III-C. $\mathbb{E}[\mathbf{z}_{N12}]$ which is the mean vector in the Pearson correlation coefficients batch is as follows.

$$\mathbb{E}[\mathbf{z}_{N12,k}] = \frac{1}{m} \sum_{i=1}^{m} \mathbf{z}_{N12,i,k},$$

where $m$ is batch size, $i$ is an index for the samples in the batch, and $k$ is an index for the latent vector. $\mathbb{E}[\mathbf{z}_{N12}]$ is the mean of $\mathbb{E}[\mathbf{z}_{N12}]$, as follows

$$\mathbb{E}[\mathbf{z}_{N12}] = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[\mathbf{z}_{N12,k}],$$

where $K$ is the dimension for the latent vector. In Fig. 7, it can be seen that $\mathbb{E}[\mathbf{z}_{N12,k}]$ for in-distribution samples is almost higher than that of OOD samples, and $\mathbb{E}[\mathbf{z}_{N12}]$ for in-distribution samples is substantially higher than that of OOD samples because samples can be characterized by certain dominant latent vectors. Each $\mathbb{E}[\mathbf{z}_{N12,k}]$ and $\mathbb{E}[\mathbf{z}_{N12}]$ in Fig. 7 are from the training step whose AUC is the highest and we can see similar observations in almost all of the results. Consequently, in-distribution samples have a greater correlation between $\mathbf{z}_1$ and $\mathbf{z}_2$ than OOD samples. To maximize the effect of the correlation between $\mathbf{z}_1$ and $\mathbf{z}_2$, we used encoders $G_{ER}$, $G_{EG}$, $G_{EB}$ which are assigned to each of red, blue, and green channels of color images such as the CIFAR10 and COIL100 (please refer to Table 6 in Appendix B for details).

V. CONCLUSION

In this work we proposed the CAE and the HD that provide the solution for problems of previous generative/reconstruction model-based approaches for anomaly detection. Based on the theoretical analysis and experimental results, we confirm that the proposed CAE and HD was superior to the previous autoencoder, discriminator, and MSE in despite of the large intra-class variance and free-hyperparameters. In addition, the proposed method was comparable to state-of-the-art generative/reconstruction model-based approaches for OOD detection with no requiring any data-sets-dependent and empirical hyperparameters. Although the CAE and the HD here were implemented using a naive autoencoder and discriminator, the CAE and the HD can be applied to other generative and discriminative models. We believe that the proposed CAE and HD can be employed not only for anomaly detection, but also in other fields where selective reconstruction error and conditional/unconditional discrimination are important.

APPENDIX A

PROOF OF GLOBAL OPTIMUM FOR $D_2$

The losses for the networks including $D_2(\mathbf{x}, \cdot)$ and $G$ are given in Eq. (2). The concatenation of a given sample $\mathbf{x}$ and a generated sample $G(\mathbf{x})$ can be modeled as the conditional distribution. The conditional PDF for the first term in Eq. 2 is $p_d(\mathbf{x})$, and the conditional PDF for the second term in Eq. (2) is $p_G(G(\mathbf{x})|\mathbf{x})$. Then, applying the definition of expectation for continuous random variables and Radon-Nikodym’s theorem to Eq. (2) gives

$$\min_{G} \max_{D} V(D, G) = \int p_d(\mathbf{x}) \left[ \log D(\mathbf{x}, \mathbf{x}) \right] d\mathbf{x} + \int p_G(G(\mathbf{x})|\mathbf{x}) \left[ \log(1 - D(\mathbf{x}, G(\mathbf{x}))) \right] d\mathbf{x}. \tag{8}$$

by minimizing $\mathbb{E}[\mathbf{z}_{N12}]$ for $\mathbf{z}_{N12}$ which was 16.
TABLE 6. Network architecture details of the proposed framework.

| Data Set  | Network Architecture | Input Size: \((C \times H \times W)\) | Batch Normalization | Layer: (Kernel Size, Output Channel, Stride, Padding) | Activation | Output Size: \((C \times H \times W)\) |
|-----------|----------------------|--------------------------------------|---------------------|-----------------------------------------------|------------|--------------------------------------|
| MNIST (2048) | \(G_{B1}, G_{B2}\) \((0.001)\) | \(1 \times 32 \times 32\) | False, True | Conv2d\((4 \times 4), 32, 2, 1\) | ReLU | \(32 \times 16 \times 16\) |
| Fashion MNIST (1200) | \(G_{B}\) \((0.001)\) | \(32 \times 18 \times 16\) | False, True | Conv2d\((4 \times 4), 64, 2, 1\) | ReLU | \(64 \times 8 \times 8\) |
| COIL100 (Greyscales) (54) | \(D_1, D_2\) \((0.00005)\) | \(128 \times 5 \times 5\) | False, True | Conv2d\((3 \times 3), 256, 1\) | ReLU | \(256 \times 2 \times 3\) |
| COIL100 (54) | \(G_{B1}, G_{B2}\) \(G_{B1}, G_{B2}\) \(G_{B1}, G_{B2}\) \((0.0001)\) | \(1 \times 32 \times 32\) | False, True | Conv2d\((4 \times 4), 64, 2, 1\) | ReLU | \(64 \times 8 \times 8\) |
| CIFAR10 (400) | \(D_1, D_2\) \((0.00002)\) | \(3 \times 32 \times 32\) | True | Conv2d\((4 \times 4), 32, 2, 1\) | ReLU | \(32 \times 16 \times 16\) |

The optimal discriminator \(D^*(x, \cdot)\) is arg max \(V(D, G)\). As in [26], if we change \(a = p_d(x), b = p_g(G(x)|x)\) and find \(D\) with \(\frac{\partial V}{\partial D} = 0\), the optimal discriminator \(D^*(x, \cdot)\) is

\[
D^*(x, \cdot) = \frac{p_d(x)}{p_d(x) + p_g(G(x)|x)}.
\]

(9)

Substituting Eq. (9) into Eq. (8)

\[
V(D^*, G) = \int p_d(x)\log \frac{p_d(x)}{p_d(x) + p_g(G(x)|x)} dx + \int p_g(G(x)|x)\log \frac{p_g(G(x)|x)}{p_d(x) + p_g(G(x)|x)} dx.
\]

(10)

If we change \(p_g(G(x)|x)\) in Eq. (10) to \(p_d(x)\), Eq. (10) is equal to \(V(D^*, G)\) in [26]. Therefore, the global optimum that satisfies arg max \(V(D, G)\) and arg min \(V(D, G)\) simultaneously is achieved at \(p_d(x) = p_g(G(x)|x)\).

APPENDIX B

IMPLEMENTATION DETAILS

We present the network architecture details of the proposed framework in Table 6. For other hyperparameters and algorithm details please refer to the Algorithm 1.

REFERENCES

[1] D. Hendrycks, M. Mazeika, and T. G. Dietterich, “Deep anomaly detection with outlier exposure,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2019, pp. 1–18.
[2] S. Liang, Y. Li, and R. Srikant, “Enhancing the reliability of out-of-distribution image detection in neural networks,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2018, pp. 1–15.
[3] A. A. Alemi, I. Fischer, and J. V. Dillon, “Uncertainty in the variational information bottleneck,” in Proc. Conf. Uncertainty Artif. Intell. (UAI)-Workshops, 2018, pp. 1–10.
[4] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2018, pp. 14707–14718.
[5] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, “Likelihood ratios for out-of-distribution detection,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2019, pp. 14707–14718.
[6] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque, “Input complexity and out-of-distribution detection with likelihood-based generative models,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2020, pp. 1–15.
[7] S. Abati, A. Porrello, S. Calderara, and R. Cucchiara, “Deep anomaly detection with adversarial autoencoders,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3379–3388.
[8] J. M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, “Adversarially learned one-class classifier for novelty detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3379–3388.
[9] P. Pidhorskyi, R. Almohsen, and G. Doretto, “Generative probabilistic novelty detection with adversarial autoencoders,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2019, pp. 14707–14718.
[10] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, “Likelihood ratios for out-of-distribution detection,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2019, pp. 14707–14718.
[11] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque, “Input complexity and out-of-distribution detection with likelihood-based generative models,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2020, pp. 1–15.
[12] S. Abati, A. Porrello, S. Calderara, and R. Cucchiara, “Latent space autoregression for novelty detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 481–490.
D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017, pp. 1–12.

K. Lee, H. Lee, K. Lee, and J. Shin, “Training confidence-calibrated classifiers for detecting out-of-distribution samples,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2018, pp. 1–16.

D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, “Using self-supervised learning can improve model robustness and uncertainty,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2019, pp. 15663–15674.

D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2018, pp. 10215–10224.

T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, “PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017, pp. 1–10.

D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2013, pp. 1–14.

C. M. Bishop, “Novelty detection and neural network validation,” IEE Proc.-Vis. Image Signal Process., vol. 141, no. 4, p. 217, 1994.

D. Blei, K. Heller, T. Salimans, M. Welling, and Z. Ghahramani, “Advances in approximate Bayesian inference,” in Proc. NeurIPS Work- shop, Dec. 2017. [Online]. Available: https://github.com/bleilab/Hierarchical-Gaussian-Processes.

E. Nalisnick, A. Matsukawa, A. G. W. W. Teh, D. Gorur, and B. Lakshminarayanan, “Do deep generative models know what they don’t know?” in Proc. Int. Conf. Learn. Represent. (ICLR), 2019, pp. 1–19.

S. Mohtemsi, M. Pitale, J. Yadawa, and Z. Wang, “Self-supervised learning for generalizable out-of-distribution detection,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 5216–5223.

R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” 2019, arXiv:1901.03407. [Online]. Available: http://arxiv.org/abs/1901.03407.

S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song, “Anomalous instance detection in deep learning: A survey,” in Proc. IEEE Symp. Secur. Privacy, 2018. [Online]. Available: https://www.osti.gov/servlets/purl/1631092 and https://www.osti.gov/biblio/20036979.

A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, “Adversarial autoencoders,” in Proc. Int. Conf. Learn. Represent. (ICLR)-Workshops, 2016, pp. 1–16.

P. Luc, N. Neverova, C. Couppie, J. Verbeek, and Y. LeCun, “Predicting deeper into the future of semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 648–657.

Y. Yazici, K.-H. Yap, and S. Winkler, “Auto-regressive generative adversarial networks,” in Proc. Int. Conf. Learn. Represent. (ICLR)-Workshops, 2018, pp. 1–12.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2014, pp. 2672–2680.

Y. Ganin, T. Kulkarni, I. Babuschkin, S. M. A. Eslami, and O. Vinyals, “Synthesizing programs for images using reinforced adversarial learning,” in Proc. Int. Conf. Mach. Learn. (ICML), 2018, pp. 1–12.

R. M. Gray and L. D. Davison, An Introduction to Statistical Signal Processing, Cambridge, U.K.: Cambridge Univ. Press, 2004.

R. Steyer and W. Nagel, Probability and Conditional Expectation: Fundamentals for the Empirical Sciences. Hoboken, NJ, USA: Wiley, 2017.

L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, “Deep one-class classification,” in Proc. Mach. Learn. Res., 2018, pp. 4393–4400.

C. You, D. P. Robinson, and R. Vidal, “Provable self-representation based outlier detection in a union of subspaces,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3395–3404.

S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravidran, “Correlational neural networks,” Neural Comput., vol. 28, no. 2, pp. 257–285, Feb. 2016.

Y. Li and N. Vasconcelos, “Efficient multi-domain learning by covariance normalization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 5424–5433.

H. Gao, Z. Shou, A. Zareian, H. Zhang, and S.-F. Chang, “Low-shot learning via covariance-preserving adversarial augmentation networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2018, pp. 975–985.

W. Li, J. Xu, J. Heo, L. Wang, Y. Gao, and J. Lu, “Distribution consistency based covariance metric networks for few-shot learning,” in Proc. AAAI Conf. Artif. Intell., 2019, pp. 8642–8649.

JUNGEON LEE received the B.S. degree in electronic engineering (major) and computer science and engineering (minor) from Jeonbuk National University, Jeonju, South Korea, in 2011, and the M.S. degree in future vehicle from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2013. He is currently pursuing the Ph.D. degree in future vehicle with the Korea Advanced Institute of Science and Technology, Daejeon.

His current research interests focus on trustworthy machine learning for real-world deployment. He mainly focuses on the development of algorithm for anomaly detection and accident anticipation using deep neural networks. However, he also has a broad interest in optimizing the machine learning algorithm such as neural architecture search (NAS) methods.

MUHAMMAD UMAR KARIM KHAN received the B.S. degree in electrical engineering from the University of Engineering and Technology (UET) Peshawar, in 2009, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), in 2014 and 2017, respectively.

He is currently working with the Center for Integrated Smart Sensors, KAIST, as a Researcher. He is also working with QiSens, a startup company which provides AI and vision-based solutions. His research interests include computer vision, artificial general intelligence, deep learning, and optimal implementations.

CHONG-MIN KYUNG (Life Fellow, IEEE) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1975, 1977, and 1981, respectively.

He was with Bell Telephone Laboratories, Murray Hill, NJ, USA, from 1981 to 1983. He has been at KAIST since 1983, involved in CAD algorithms, 3-D graphics, and system-on-a-chip design and verification methodology, development of various RISC/CISC microprocessors, very long instruction word and reconfigurable digital signal processing cores. He has also served as the Hyundai Chair Professor at KAIST from 2008 to 2013, where he is currently the Founding Director of the Integrated Circuit Design Education Center, and currently leads the Center for Integrated Smart Sensors, funded as Global Frontier Project by the Korean Government. His current research interests include system-level low-power design, electrical/thermal co-design in 3-D IC, architectures of H.264 video CODEC, and energy-rate-distortion-power optimization.

Dr. Kyung is a member of the National Academy of Engineering Korea and the Korean Academy of Science and Technology. He received the Best Paper Awards in the 36th DAC held in New Orleans, LA, USA, the 10th International Conference on Signal Processing Applications and Technology, Orlando, FL, USA, in 1999; and the 1999 IEEE International Conference on Computer Design Austin, TX, USA. In 2000, he received the National Medal from the Korean Government for his contribution to research and education in IC design. He was a recipient of the Most Excellent Design Award and Special Feature Award in the University Design Contest at ASP-DAC in 1997 and 1998, respectively. He has served as the General Chair in the 2002 Korean Semiconductor Conference, the 2004 International Soc. Design Conference, the 2007 IEEE Asian Solid-State Circuits Conference; and the 2008 Asia South Pacific Design Automation Conference (ASP-DAC).