IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 24, 2020, accepted July 13, 2020, date of publication July 20, 2020, date of current version July 29, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3010274*

# Anomalous Example Detection in Deep Learning: A Survey

**SAIKIRAN BULUSU[1], (Graduate Student Member, IEEE), BHAVYA KAILKHURA[2], (Member, IEEE), BO LI[3], (Member, IEEE), PRAMOD K. VARSHNEY[1], (Life Fellow, IEEE), AND DAWN SONG[4], (Fellow, IEEE)**

[1]EECS Department, Syracuse University, Syracuse, NY 13244, USA
[2]Lawrence Livermore National Laboratory, Livermore, CA 94550, USA
[3]Computer Science Department, University of Illinois at Urbana–Champaign, Champaign IL 61820, USA
[4]EECS Department, University of California at Berkeley, Berkeley CA 94720, USA

Corresponding author: Saikiran Bulusu (sabulusu@syr.edu)

This work was supported by the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

**ABSTRACT** Deep Learning (DL) is vulnerable to out-of-distribution and adversarial examples resulting in incorrect outputs. To make DL more robust, several posthoc (or runtime) anomaly detection techniques to detect (and discard) these anomalous samples have been proposed in the recent past. This survey tries to provide a structured and comprehensive overview of the research on anomaly detection for DL based applications. We provide a taxonomy for existing techniques based on their underlying assumptions and adopted approaches. We discuss various techniques in each of the categories and provide the relative strengths and weaknesses of the approaches. Our goal in this survey is to provide an easier yet better understanding of the techniques belonging to different categories in which research has been done on this topic. Finally, we highlight the unsolved research challenges while applying anomaly detection techniques in DL systems and present some high-impact future research directions.

**INDEX TERMS** Anomaly detection, out-of-distribution, adversarial examples, deep learning, neural network.

## I. INTRODUCTION

Deep Learning (DL) techniques provide incredible opportunities to answer some of the most important and difficult questions in a wide range of applications in science and engineering. Therefore, scientists and engineers are increasingly adopting the use of DL for making potentially important decisions in the context of applications of interest, such as bioinformatics, healthcare, cyber-security, and fully autonomous vehicles. Several of these applications are often high-regret (i.e., incurring significant costs) in nature. In such applications, incorrect decisions or predictions have significant costs either in terms of experimental resources when testing drugs, lost opportunities to observe rare phenomena, or in health and safety when certifying parts. Most DL methods implicitly assume ideal conditions and rely on the assumption that test data comes from the ''same distribution'' as the training data. However, this assumption is not satisfied in many real-world applications and virtually all problems require various levels of transformation of the DL output as test data is typically different from the training data either due to noise, adversarial corruptions, or other changes in distribution possibly due to temporal and spatial effects. These deviant (or out-of-distribution) data samples are often referred to as anomalies, outliers, novelties in different domains. In general, an anomalous example is one which deviates from the other examples so as to introduce uncertainty as to whether it was produced by the genuine source or by an alternate source. Note that the anomalous examples can be classified into many types based on the context and applications which are enumerated in [1]. It is well known that DL models are highly sensitive to such anomalies, which often leads to unintended and potentially harmful consequences due to incorrect results generated by DL. Hence, it is critical to determine whether the incoming test data is so different from the training dataset that the output of the model cannot be trusted (referred to as the *anomaly detection* problem).

Due to its practical importance, anomaly detection has received a lot of attention from statistics, signal processing and machine learning (ML) communities. Recently, there has

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

been a surge of interest in devising anomaly detection methods for DL applications. This survey aims to provide a structured overview of recent studies and approaches to anomaly detection in DL based high-regret applications. To the best of our knowledge, there has not been any comprehensive review of anomaly detection approaches in DL systems. Although a number of surveys have appeared for conventional ML applications, none of these are specifically for DL applications. This has motivated this survey paper especially in light of recent research results in DL. We expect that this review will facilitate a better understanding of the different directions in which research has been carried out on this topic and potential high-impact future directions.

### A. RELATED WORK

Anomaly detection is the subject of various surveys, review articles, and books. In [2], a comprehensive survey of various categories of anomaly detection techniques for conventional ML as well as statistical models is presented. For each category of detection, various techniques and their respective assumptions along with the advantages and disadvantages are discussed. The computational complexity of each technique is also mentioned. A comprehensive survey of the novelty detection techniques is presented in [3]. Various techniques are classified based on the statistical models used and the complexity of methods. Recently, an elaborate survey is presented in [4] where DL based anomaly detection techniques are discussed. Here, two more categories of anomaly detection, namely, hybrid models as well as one-class deep neural network (DNN) techniques are also included. Note that our survey paper is different from [4] as our focus is on discussing unintentional and intentional anomalies specifically in the context of DNNs whereas [4] discusses approaches which use DNN based detectors applied to conventional ML problems. In some sense, our survey paper is much broader in the context of DL applications. In [5], a survey of the data mining techniques used for anomaly detection are discussed. The techniques discussed are clustering, regression, and rule learning. Furthermore, in [6], the authors discuss the models that are adaptive to account for the data coming from the dynamically changing characteristics of the environment and detect anomalies from the evolving data. Here, the techniques account for the change in the underlying data distribution and the corresponding unsupervised techniques are reviewed. In [7], the anomaly detection techniques are classified based on the type of data namely, metric data, evolving data, and multi-structured data. The metric data anomaly detection techniques consider the use of metrics like distance, correlation, and distribution. The evolving data include discrete sequences and time series. In [8], various statistical techniques, data mining based techniques, and ML based techniques for anomaly detection are discussed. In [9], [10], the existing techniques for anomaly detection which include statistical, neural network based, and other ML based techniques are discussed. Various books [11]–[14] also discussed the techniques for anomaly detection.

### B. OUR CONTRIBUTIONS

To the best of our knowledge, this survey is the first attempt to provide a structured and a broad overview of extensive research on detection techniques spanning both unintentional and intentional anomalies in the context of DNNs. Most of the existing surveys on anomaly detection focus on (i) anomaly detection techniques for conventional ML algorithms and statistical models, (ii) novelty detection techniques for statistical models, (iii) DL based anomaly detection techniques. In contrast, we provide a focused survey on post-hoc anomaly detection techniques for DL. We classify these techniques based on the availability of labels for the training data corresponding to anomalies, namely, supervised, semi-supervised, and unsupervised techniques. We discuss various techniques in each of the categories and provide the relative strengths and weaknesses of the approaches. We also briefly discuss anomaly detection techniques that do not fall in the post-hoc category, e.g., training-based, architecture design, etc.

### C. ORGANIZATION

This survey is organized mainly in three parts: detection of unintentional anomalies, detection of intentional anomalies, and applications. For both unintentional and intentional anomalies, we will discuss different types of approaches (as illustrated in Fig. 1). In Sec. II, we present the methodology that we used to prepare our survey paper. In Sec. III, we present the classification of anomalies and the possible challenges in anomaly detection in our context. In Sec. IV, we present various post-hoc anomaly detection techniques which are used to detect unintentional anomalies. These techniques are classified based on the availability of labels. In Sec. V, we present various post-hoc anomaly detection techniques which are used to detect intentional anomalies (or adversarial examples). The techniques are again classified based on the availability of labels. In Sec. VI, we discuss strengths and weaknesses of different categories of methods. In Sec. VII, we describe various application domains where anomaly detection is applied. Finally, we conclude and present open questions in this area in Sec. VIII.

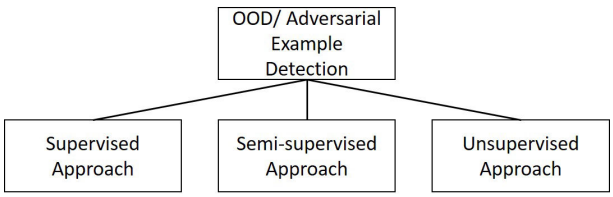


**FIGURE 1. Schematic representation of different types of anomaly detection techniques discussed in this survey.**

## II. SURVEY METHODOLOGY

We performed a comprehensive survey of the existing literature to capture the essence and design aspects of anomaly detection techniques in DL. We used a structured methodology to find anomaly detection relevant research and to

classify the proposed methods in the following sections. For our survey paper, we started by looking for research work in top conferences and journals in fields like computer vision and ML. The conferences from which we selected related papers are CEUR, AINS, ICCV, ICLR, ICAIS, ICML, CIKM, ECCV, IWCBMI, SMC, ICACBD, CSCloud, MILCOM, ICBDA, DMIN, SPW, KDDM, ICDM, ICC, IJCAI, KDD, AAAI, CVPR, and NeurIPS. Furthermore, the journals from which we selected related papers are International Journal of Data Science and Analytics, Artificial intelligence review, Data mining and knowledge discovery, Journal of Ambient Intelligence and Humanized Computing, IEEE Transactions on Signal Processing, IEEE Communication Letters, IEEE Internet of Things. As anomaly detection in deep learning is a rapidly growing research area, we included several *arXiv* preprints as well.

We started with papers related to post hoc anomaly detection in deep learning for out-of-distribution (OOD) test examples (unintentional anomaly detection) and adversarial test examples (intentional anomaly detection). We further classified the research papers based on the availability of labels for the OOD and adversarial examples. The papers are classified based on whether labeled OOD or adversarial examples are available for anomaly detection (supervised approaches), where unlabeled OOD or adversarial examples are available for anomaly detection (semi-supervised approaches), and where only in-distribution examples are utilized for anomaly detection (unsupervised approaches). Specifically, Table 1 shows a digest of fourteen surveyed papers under the supervised, semi-supervised, and unsupervised classification for post hoc OOD detection in deep learning with their respective descriptions. In Section III.D, we have included the discussion on thirteen papers that perform anomaly detection for OOD examples based on either training-based techniques, or architecture design-based techniques that are different from post hoc anomaly detection approaches. Similarly, Table 3 shows a digest of fourteen surveyed papers under the supervised, semi-supervised, and unsupervised classification approaches for post hoc adversarial detection in deep learning with their respective descriptions. To be complete, we have also added some recent papers in Tables 2, 4 which propose techniques different from the post hoc technique for both unintentional and intentional examples. In Section IV.D, we have included eleven papers that perform detection of adversarial examples based on techniques that are different from post hoc anomaly detection approaches. Furthermore, in Section VI, we have included thirty papers from different application domains that perform anomaly detection in deep learning.

**TABLE 1. OOD detection related papers.**

| Classification Type | Reference | Contributions |
|---|---|---|
| Supervised | [19] | Uncertainty measure based on the gradient of the negative log-likelihood is used as a measure of confidence |
| Supervised | [20] | Confidence scores based on Mahalanobis distance from different layers is combined using weighted averaging |
| Supervised | [21] | Invariance of classifier's softmax under various transformations to input image is used as a measure of confidence |
| Supervised | [22] | Ratio of Hausdorff distances between test sample to the nearest non-predicted and the predicted classes is used as the trust score |
| Semi-supervised | [23] | Likelihood ratio-based method is used to differentiate between in-distribution and OOD examples |
| Semi-supervised | [24] | A two-head CNN consisting of a common feature extractor and two classifiers with different decision boundaries is trained to detect OOD examples |
| Unsupervised | [25] | Predicted softmax probability is used to detect OOD examples |
| Unsupervised | [26] | Temperature scaling and by adding small perturbations to the input is used to better separate the softmax score for OOD detection |
| Unsupervised | [27] | GAN based architecture is used to compare the bottleneck features of the generated image with that of the test image |
| Unsupervised | [28] | Degenerated prior network with concentration perturbation algorithm is used to get better uncertainty measure |
| Unsupervised | [29] | Learning to discriminate between geometric transformations is used for learning unique features that are useful in OOD detection |
| Unsupervised | [30] | Mahalanobis distance is applied in the latent space of the autoencoder to detect OOD examples |
| Unsupervised | [31] | Resampling uncertainty estimation approach is proposed as an approximation to the bootstrap |

**TABLE 2. Miscellaneous papers on OOD detection.**

| Classification Type | Reference | Contributions |
|---|---|---|
| Uncertainty Quantification | [43] | Watanabe-Akaike Information Criterion gives an estimate of the gap between the training set and test set expectations |
| Uncertainty Quantification | [44] | Brier score based on posterior probabilities is used to measure the model uncertainty about a test example |
| Architecture design | [45] | Hybrid model combines temporal pattern of user activity using RNN and softmax values for anomaly detection |
| Architecture design | [46] | Class probabilities are modeled using softmax of scaled cosine similarities that are used to differentiate between in-distribution and OOD examples |
| Architecture design | [47] | Deep autoencoder is used to extract essential features that are fed to CNN to perform OOD detection |
| Architecture design | [48] | Ensemble of leave-out-classifiers are trained using in-distribution and OOD examples along with novel loss function that ensures a margin between normal and OOD examples is used for anomaly detection |
| Training-based | [49] | Numenta Anomaly benchmark metric along with an adapted RNN is used for anomaly detection |
| Training-based | [50] | Training the classifier based on OOD examples generated by GAN used to detect anomalous test examples |
| Training-based | [51] | Training the classifier using an auxiliary dataset of diverse outliers called outlier exposure to detect OOD examples |
| Training-based | [52] | Generative model that is trained with batch normalization along with permutation test statistic based on expected conditional likelihood of test example is used to detect OOD examples |
| Training-based | [53] | Training the classifier to learn the features to better separate in-distribution and OOD examples to detect anomalous test examples |
| Training-based | [54] | Robust training is performed by exposing the model to both adversarially crafted in-distribution and OOD examples to make the detector robust |

## III. BACKGROUND

### A. WHAT ARE ANOMALIES?

The problem setup for anomaly detection in DNNs is as follows: the DNN is trained on in-distribution data and is asked to perform predictions on both in-distribution as well as OOD test samples. In-distribution test samples are from the same distribution as the training data and the trained DNN is expected to perform reliably on them. On the other hand,

**TABLE 3. Adversarial example detection related papers.**

| Classification Type | Reference | Contributions |
|---|---|---|
| Supervised | [57] | Binary detector trained on intermediate feature representations is proposed to detect adversarial examples |
| Supervised | [58] | Logistic regression based detector trained with two features: the uncertainty and the density estimate is used |
| Supervised | [59] | LSTM based binary detector is trained to analyze the sequence of deep features embedded in a distance space |
| Supervised | [60] | Local Intrinsic Dimensionality is used to characterize the dimensional properties of the regions where the adversarial examples lie |
| Supervised | [61] | Three layer regression NN used as the detector to predict confidence score |
| Unsupervised | [62] | Rank based statistics with generative models is used for detecting adversarial examples |
| Unsupervised | [63] | KL distance based metric is applied on the posterior distributions to detect the adversarial examples |
| Unsupervised | [64] | Nearest neighbor classification score based on deep features is used as to detect adversarial examples |
| Unsupervised | [65] | Adversarial examples are detected by modeling output distribution of hidden layers of the DNN given normal examples |
| Unsupervised | [66] | Provenance and activation invariance is used to detect adversarial examples |
| Unsupervised | [67] | Mutual Information is used to detect adversarial examples by minimizing uncertainty over sampling probabilities |
| Unsupervised | [68] | Detection by nearest neighbor search based projections of adversarial examples onto in-distribution image manifold is used |
| Unsupervised | [69] | Detection by gradient search based projections of adversarial examples onto in-distribution image manifold is used |
| Unsupervised | [70] | Sensitivity of adversarial examples under compression based transformations is used as a measure of confidence |

anomalous test samples are samples which do not conform to the distribution of the training data. Therefore, predictions of DNNs based on these anomalous samples should not be trusted. The goal of the anomaly detection problem is to design post-hoc detectors to detect these nonconforming test samples (see Fig. 2).

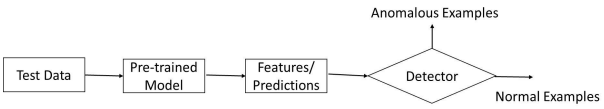


**FIGURE 2. Schematic of anomaly detection in DL.**

Next, we discuss the types of anomalies, and present their respective differences. We classify anomalies into (a) unintentional and (b) intentional (see Fig. 3) types. Unintentional anomalies are independent of the DNN model, as opposed to, intentional anomalies which are intentionally designed by an attacker to force the DNN model to yield incorrect results, and are model dependent.

#### 1) UNINTENTIONAL: NOVEL AND OUT-OF-DISTRIBUTION EXAMPLES

The unintentional anomalies are further classified into novel and OOD examples.[1] Novelty detection is the identification of new or unknown in-distribution data that a ML system is not aware of during training. However, the OOD example comes from a distribution other than that of the training data. The distinction between novelties and OOD data is that the novel data samples are typically incorporated into the normal model after being detected, however, OOD samples are usually discarded. In Fig. 3, the blue circles outside class boundaries are OOD examples. The OOD examples do not belong to any of the classes. In other words, the classifier is either unaware or does not recognize the OOD examples.

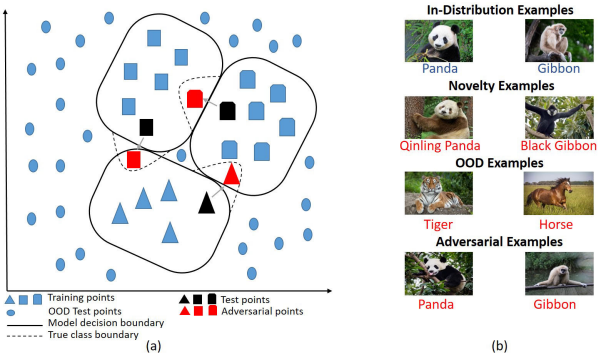


**FIGURE 3. (a) A simple example of anomalies in a 2-dimensional data set. (b) Different variants of anomalous examples for the Panda vs. Gibbon classification problem. Captions indicate the true label and the color indicates whether the prediction is correct or wrong (blue for correct and red for wrong).**

A related problem arises in Domain adaptation (DA) and transfer learning [15] which deal with scenarios where a model trained on a source distribution is used in the context of a different (but related) target distribution. The difference between the DA and OOD problems is that DA techniques assume that the test/target distribution is related to the task (or distribution) of interest (thus, utilized during training). On the other hand, OOD techniques are designed to detect if incoming data is so different (and unrelated) from the training data that the model cannot be trusted.

#### 2) INTENTIONAL: ADVERSARIAL EXAMPLES

The intentional anomalies (also known as the adversarial examples) are the test inputs that are intentionally designed by an attacker to coerce the model to make a mistake. For example, an attacker can modify the input image to fool the DNN classifier which could lead to unforeseen consequences, such as, accidents of autonomous cars or possible bank frauds. In Fig. 3, the examples in red are adversarial in nature. Via small perturbation at the input, these examples have been moved to other class regions leading to misclassification. The classifier may or may not have access to some of the labels of these examples leading to different techniques in the literature.

### *B. CHALLENGES*

As mentioned above, anomalies are data samples that do not comply with the expected normal behavior. Hence, a naive

[1]Note that we refer to samples as examples.

approach for detecting anomalies is to define a region in the data space that represents normal behavior and declare an example as anomaly if it does not lie in this region. However, there are several factors that make this seemingly simple method ineffective:

- The boundary between the normal and anomalous regions is very difficult to define, especially, in complex DNN feature spaces.
- Based on the type of applications, the definition of an anomaly changes. For certain applications, a small deviation in the classification result from that of the normal input data may have far reaching consequences and thus may be declared as anomaly. In other applications, the deviation needs to be large for the input to be declared as an anomaly.
- The success of some anomaly detection techniques in the literature depends on the availability of the labels for the training and/or testing data.
- Anomaly detection is particularly difficult when the adversarial examples tend to disguise themselves as normal data.

The aforementioned difficulties make the anomaly detection problem difficult to solve in general. Therefore, most of the techniques in the literature tend to solve a specific instance of the general problem based on the type of application, type of input data and model, availability of labels for the training and/or testing data, and type of anomalies.

## IV. UNINTENTIONAL ANOMALY DETECTION

In this section, we discuss the detection techniques which detect the OOD examples given a pre-trained neural network. Most DL approaches assume that the test examples belong to the same distribution as the training examples. Consequently, the neural networks are vulnerable to test examples which are OOD. Hence, we need techniques to improve the reliability of the predictions or determine whether the test example is different in distribution from that of the training dataset. Here, we concentrate on the techniques that determine whether the test example is different in distribution from that of the training dataset, using the pre-trained DNN followed by a detector. We refer to this architecture as post-hoc anomaly detection. A topic related to OOD example detection is novelty detection [16]–[18] which aims at detecting previously unobserved (emergent, novel) patterns in the data. It should be noted that solutions for novelty detection related problems are often used for OOD detection and vice-versa, and hence we use these terms interchangeably in this survey. Based on the availability of labels for OOD data, techniques are classified as supervised, semi-supervised, and unsupervised which are discussed next and summarized in Table 1.

### A. SUPERVISED APPROACHES

In this section, we review the anomaly detection approaches when the labels of both the in-distribution and the OOD examples are available to enable differentiation between them as the supervised anomaly detection problem. Any unseen test data sample is compared with the detector to determine which class (in-distribution vs. OOD) it belongs to.

In [19], an approach to measure uncertainty of a neural network based on gradient information of the negative log-likelihood at the predicted class label is presented. The gradient metrics are computed from all the layers in this method and scalarized using norm or min/max operations. A large value of the gradient metrics indicates incorrect classification or OOD example. A convolutional neural network (CNN) is used as the classifier trained on Extended MNIST digits [32]. EMNIST letters, CIFAR10 [33] images as well as different types of noise are used as OOD data. The authors found that such an unsupervised scheme does not work well on all types of OOD data. Therefore, a supervised variant of this scheme where one allows an anomaly detector to be trained on uncertainty metrics of some OOD samples is proposed. It is shown that the performance is improved considerably by utilizing the labeled OOD data.

In [20], the high-level idea is to measure the probability density of test sample on DNN feature spaces. Specifically, the authors fit class-conditional Gaussian distributions to pre-trained features. This is possible since the posterior distribution can be shown to be equivalent to the softmax classifier under Gaussian discriminant analysis. Next, a confidence score using the Mahalanobis distance with respect to the closest class conditional distribution is defined. Its parameters are chosen to be empirical class means and tied empirical covariance of training samples. To further improve the performance, confidence scores from different layers of DNN is combined using weighted averaging. Weight of each layer is learned by training a logistic regression detector using labeled validation samples comprising of both in-distribution and OOD data. The method is shown to be robust to OOD examples.

In [21], a detector is trained on representations derived from a set of classifier responses generated from applying different natural transformation to a given image. Analyzing the invariance of classifier's decision under various transformations establishes a measure of confidence in its decision. In other words, the softmax values of the OOD input should fluctuate across transformed versions, while those of the in-distribution image should be relatively stable. The authors trained a binary OOD detector on confidence scores under various transformations for in-distribution vs. OOD training data. ResNet based architecture is used as the classifier and the Self-Taught Learning (STL-10) dataset [34] is used as the in-distribution data and the Street View House Numbers (SVHN) dataset [35] is used as the OOD data. The approach is shown to outperform other baselines.

In [22], a trust score is proposed to know whether the prediction of a test example by a classifier can be trusted. This score is defined as the ratio of the Hausdorff distances between the distance from the testing sample to the nearest class different from the predicted class (e.g., OOD class) and the distance to the predicted class. To compute the trust score,

the training data is pre-processed to find a high density set of each class to filter outliers. The trust score is estimated based on this high density set. The idea behind the approach is that if the classifier predicts a label that is considerably farther than the closest label, then it may be an OOD or unreliable example. For the task of identifying correctly/incorrectly classified examples, it is shown that the trust score performs well in low to medium dimensions. However, it performs similar to classifiers' own reported confidence (i.e., probabilities from the softmax layer) in high dimensions.

### B. SEMI-SUPERVISED APPROACHES

We refer to the anomaly detection techniques as semi-supervised if they utilize unlabeled contaminated data (or information) in addition to labeled instances of in-distribution class. Since, these techniques do not require to know whether unlabeled instance is in-distribution or OOD, they are more widely applicable than supervised techniques.

In [23], a likelihood ratio-based method using deep generative models is presented to differentiate between in-distribution and OOD examples. The authors assumed that the in-distribution data is comprised of both semantic and background parts. The authors found that the likelihood can be confounded by the background (e.g. OOD input with the same background but different semantic component). Using this information about OOD data, they propose to use a background model to correct for the background statistics and enhance the in-distribution specific features for OOD detection. Specifically, background model is trained by adding the right amount of perturbations to inputs to corrupt the semantic structure in the data. Hence, the model trained on perturbed inputs captures only the population level background statistics. This likelihood ratio is computed from the in-distribution data and the background statistics. If the likelihood ratio is larger than a pre-specified threshold, it is highly likely that the test example is OOD. The National Center for Biotechnology Information microbial genome dataset is utilized in [23] in the following manner. Various bacteria are grouped into classes which were discovered over the years. Specifically, the classes discovered before a given cutoff year are considered as in-distribution classes and those discovered after the cutoff year are considered OOD classes. The proposed test improves the accuracy of OOD detection compared to the accuracy of the state-of-the-art detection results.

In [24], a semi-supervised OOD detection technique based on two-head CNN is proposed. The idea is to train a two-head CNN consisting of one common feature extractor and two classifiers which have different decision boundaries but can classify in-distribution samples correctly. Further, unlabeled contaminated data is used to maximize the discrepancy between two classifiers to push OOD samples outside in-distribution manifold. This enables the detection of OOD samples that are far from the support of the in-distribution samples.

### C. UNSUPERVISED APPROACHES

We refer to the detection techniques as unsupervised if they only utilize in-distribution data for OOD detection.

In [25], as the statistics derived from the softmax distributions are helpful, a baseline method based on softmax to determine whether or not a test example is OOD is proposed. The idea is that a well trained network tends to assign higher predicted probability to in-distribution examples than to OOD examples. Hence, the OOD example can be detected by comparing the predicted softmax class probabilities of the examples to a threshold. Specifically, the authors generated the training data by separating correctly and incorrectly classified test set examples and, for each example, computing the softmax probability of the predicted class which is used to compute the threshold. The performance of this approach is evaluated on computer vision, natural language processing and speech recognition tasks. The technique fails if the classifier does not separate the maximum values of the predictive distribution well enough with respect to in-distribution and OOD examples. Therefore, the authors in [26] proposed a method based on the observation that using temperature scaling and adding small perturbations to the input can better separate the softmax score distributions between in- and out-of-distribution images. Wide ResNet [36] and DenseNet [37] architectures are used and trained using the CIFAR-10 and CIFAR-100 [33] as in-distribution datasets. The OOD detector is tested on several different natural image datasets and synthetic noise datasets. Similarly, the authors in [38] showed that self-supervised learning with rotation prediction task enables the detection of harder OOD examples. It is shown that the approach significantly improves the detection performance and outperforms the baseline in [25].

In [27], a generative adversarial network (GAN) [39] based architecture is used in reconstruction error based OOD detection method. The motivation is that the GAN will perform better when generating images from previously seen objects (i.e., in-distribution data) than it will when generating images of objects it has never seen before (i.e., OOD data). In this approach, the test image is first passed through the generator of the GAN, which produces bottleneck features and a reconstructed image. Next, the reconstructed image is passed through the encoder producing another set of bottleneck features. The Euclidean distance between these two feature sets represents a measure of how much the generated image deviates from the original image and is used as an anomaly score.

In [28], the authors propose a degenerated prior network architecture, which can efficiently separate model-level uncertainty from data-level uncertainty via prior entropy. To better separate in-distribution and OOD images, they propose a concentration perturbation algorithm, which adaptively adds noise to concentration parameters of prior network. Through comprehensive experiments, it is shown that this method achieves state-of-the-art performance especially on the large-scale dataset. However, this method is found to

be sensitive to different neural network architectures, which could sometimes lead to inferior performance.

In [29], the intuition is that learning to discriminate between geometric transformations applied to images help in learning of unique features of each class that are useful in anomaly detection. The authors train a multi-class classifier over a self-labeled dataset created by applying various geometric transformations to in-distribution images. At test time, transformed images are passed through this classifier, and an anomaly score derived from the distribution of softmax values of the in-distribution training images is used for detecting OOD data. The classifier used is the Wide Residual Network model [36] trained on CIFAR dataset. The Catsvs-Dogs dataset [40], that contains 12,500 images of cats and dogs each, is treated as the OOD data. The method performs better compared to the baseline approaches in [25] for the larger-sized images and is robust to the OOD examples. The method is able to distinguish between the normal and OOD examples with a significant margin compared to the baseline methods.

The approach in [30] (and references therein) considers the problem of detecting OOD samples based on the reconstruction error. These methods assume that OOD data is composed of different factors than in-distribution data. Therefore, it is difficult to compress and reconstruct OOD data based on a reconstruction scheme optimized for in-distribution data. Specifically, [30] proposes to incorporate the Mahalanobis distance in latent space to better capture these OOD samples. They combined the Mahalanobis distance between the encoded test sample and the mean vector of the encoded training set with the reconstruction loss of the test sample to construct an anomaly score. Single digit class from MNIST [41] is used as in-distribution and the other classes of MNIST are treated as OOD samples. The authors illustrate that by including the latent distance helps in improving the detection of in-distribution and OOD examples.

In [31], the predictions of a pre-trained DNN are audited to determine their reliability. Resampling uncertainty estimation (RUE) approach is proposed as an approximation to the bootstrap procedure. Intuitively, RUE estimates the amount that a prediction would change if different training data is used from the same distribution. It quantifies uncertainty using the gradients and Hessian of the model's loss on training data and bootstrap samples to produce an ensemble of predictions for a test input. This uncertainty score is compared to a threshold for detecting correct and incorrect predictions. A single hidden layer feedforward neural network architecture is trained using eight common benchmark regression datasets [42] from the UCI dataset repository. The authors show that the uncertainty score detects inaccurate predictions for auditing reliability compared to existing techniques more effectively. This approach can also be used to detect OOD samples.

Note that the unsupervised methods discussed above require comparing proposed anomaly scores with a threshold. Although thresholds are computed solely based on in-distribution data, one can further improve the performance by optimally choosing thresholds based on OOD validation samples (if available).

### D. OTHER MISCELLANEOUS TECHNIQUES

In this section, we discuss various approaches that are different from the post-hoc anomaly detection techniques, e.g., uncertainty quantification, architecture design, and training-based that are summarized in Table 2. Here, uncertainty quantification based techniques aim to provide meaningful quantification of predictive uncertainty by providing accurate confidence values of the model in addition to its class predictions for the test examples. The idea is that well calibrated uncertainty estimates inform whether a model's output can be trusted. The architecture design-based techniques propose novel architectures for the neural networks to perform anomaly detection. Finally, training-based techniques discuss novel training procedures for the neural networks that include modifying the training loss function to improve the ability of the neural network to recognize anomalies.

The key idea in [43] is that the likelihood models assign higher density values to the OOD examples than the in-distribution examples. The authors propose generative ensembles to detect OOD examples by combining a density evaluation model with predictive uncertainty estimation on the density model via ensemble variance. The generative ensembles compute likelihoods that are used to estimate the Watanabe-Akaike Information Criterion (WAIC). WAIC estimates the gap between the training and test set expectations. Specifically, the generative ensembles use uncertainty estimation on randomly sampled GAN discriminators to de-correlate the OOD classification errors made by a single discriminator.

In [44], the effect of OOD examples on the accuracy and calibration for the classification tasks is investigated. The authors evaluate uncertainty not only for in-distribution examples but also for OOD examples. They utilize metrics such as negative log-likelihood and Brier scores to evaluate the model uncertainty or accuracy of computed predicted probabilities. Using large-scale experiments, the authors show that the calibration error increases with increasing distribution shift and post-hoc calibration does indeed fall short in detecting OOD examples.

A hybrid model for fake news detection in [45] consists of three steps which capture the temporal pattern of user activity on a given article using a recurrent neural network (RNN), checking the credibility of the media source, and classifying the article as fake or not.

The method presented in [46] proposes to modify the output layer of DNNs. Specifically, instead of using logit scores for computing class probabilities, the cosine of the angle between the weights of a class and the features of the class are used. In other words, the class probabilities are obtained using the softmax of scaled cosine similarity. The detection of OOD samples is done by comparing the maximum of cosine values across classes to a threshold. The

method is hyperparameter-free and has high OOD detection performance. However, the trade-off is the degradation of the classification accuracy.

In [47], a deep autoencoder is combined with CNN to perform supervised OOD detection. Autoencoder is used as a pre-training method for supervised CNN training. The idea is to reconstruct high-dimensional features using the deep autoencoder and detect anomalies using CNNs. It is shown that this combination can improve the accuracy and efficiency of large-scale Android malware detection.

In [48], the algorithm comprises of an ensemble of leave-out-classifiers. Each classifier is trained using in-distribution examples as well as OOD examples. Here, the OOD examples are obtained by designating a random subset from the training dataset as OOD and the rest are in-distribution. A novel margin-based loss function is presented that maintains a margin $m$ between the average entropy of the OOD and in-distribution samples. Hence, the loss function is the cross-entropy loss along with the margin-based loss. The loss function is minimized to train the ensemble of classifiers. The OOD detection score is obtained by combining the softmax prediction score and the entropy with temperature scaling. The score is shown to be high for in-distribution examples and low for OOD examples. In [49], an RNN network is used to detect anomalous data where the Numenta Anomaly Benchmark metric is used for early detection of anomalies.

A novel training method is presented in [50] where two additional terms are added in the cross entropy loss that minimize the Kullback-Leibler (KL) distance between the predictive distribution on OOD examples and the uniform distribution to assign less confident predictions to the OOD examples. Then, in-distribution and OOD samples are expected to be more separable. However, the loss function for optimization requires OOD examples for training which are generated by using a GAN architecture. Hence, the training involves minimizing the classifier's loss and the GAN loss alternately.

Furthermore, [51] proposes leveraging alternative data sources to improve OOD detection by training anomaly detectors against an auxiliary dataset of outliers, an approach they call Outlier Exposure. The motivation is that while it is difficult to model every variant of anomaly distribution, one can learn effective heuristics for detecting OOD samples by exposing the model to diverse OOD datasets. Thus, learning a more conservative concept of the in-distribution and enabling anomaly detectors to generalize and detect unseen anomalies.

The authors in [52] propose a permutation test statistics to detect OOD samples using deep generative models trained with batch normalization. They show that the training objective of generative models with batch normalization can be interpreted as maximum pseudo-likelihood over a different joint distribution. Over this joint distribution, the estimated likelihood of a batch of OOD samples is shown to be much lower than that of in-distribution samples.

In [53], [54], benchmarking of some of the existing OOD detection techniques is performed. In [53], the challenges in adopting OOD detection methods for large-scale datasets are discussed. The authors compare post hoc methods and training-based strategies for OOD detection to measure the benefit gained by combining these methods. Here, the post hoc methods modify how the DNN outputs are used and the training-based strategies learn the feature representations to enable better OOD example detection. Hence, combining the two types of techniques improves detection performance. It is shown that the performance benefit in using training-based strategies decreases for large scale datasets. This is observed from the ROC curves for the ImageNet Intra-Dataset problem that shows, there is very little benefit from background class regularization compared to standard cross-entropy training. In [54], the susceptibility of OOD detectors to adversarial examples is explored. The authors study the problem of robust OOD detction and show that the existing detection methods are vulnerable to small perturbations to the inputs. Thus, the authors propose an algorithm that performs robust training by exposing the model to both adversarially crafted in-distribution and OOD examples. The training helps in calibrating the error on examples from unknown distribution. The authors show a significant improvement by using robust OOD detection compared to other existing techniques.

## V. INTENTIONAL ANOMALY DETECTION

In this section, we discuss the detection techniques for detecting intentionally designed adversarial test examples given a pre-trained neural network. It is well known that DNNs are highly susceptible to test time adversarial examples – small perturbations that, when added to any data, causes it to be misclassified with high probability [55], [56]. The small perturbation constraint ensures that the test example belongs to the data manifold yet gets misclassified. Hence, we need techniques to improve the reliability of the predictions or determine whether the test example is adversarial or normal. Here, we focus on the latter with the availability of a pre-trained DNN followed by a detector. Based on the availability of labels, the techniques are classified as supervised, semi-supervised, and unsupervised which are elaborated as follows and summarized in Table 3.

### A. SUPERVISED APPROACHES

In this section, we discuss the detection techniques that require the labels of both in-distribution and adversarial examples and refer to them as supervised anomaly detection techniques. The test examples are compared against the detector to determine whether they are normal or adversarial.

In [57], a binary adversarial example detector is proposed. The detector is trained on intermediate feature representations of a pre-trained classifier on the original data set and adversarial examples. Although it may seem very difficult to train such a detector, their results on CIFAR10 and a 10-class subset of ImageNet datasets show that training such a detector is indeed possible. In fact, the detector achieves high accuracy in the detection of adversarial examples. Moreover, while the detector is trained on adversarial examples generated using a

specific attack method, it is found that the detector generalizes to similar and weaker attack methods. Similar strategy is employed in [71] where ML model is augmented with an additional class in which the model is trained to classify all adversarial inputs using labeled data.

The authors in [58] propose three methods to detect adversarial examples. First, method which is based on the density estimation uses estimates from the kernel density estimation of the training set in the feature space of the last hidden layer to detect adversarial examples. This method is meant to detect points that lie far from the data manifold. However, this strategy may not work well when adversarial example is very near the benign submanifold. Therefore, the authors propose a second approach which uses Bayesian uncertainty estimates from the dropout neural networks when points lie in low-confidence regions of the input space. They show that dropout based method can detect adversarial samples in situations where density estimates cannot. Finally, they also build a combined detector which is a simple logistic regression classifier with two features as input: the uncertainty and the density estimate. The combined detector is trained on a labeled training set which comprises of uncertainty values and density estimates for both benign and adversarial examples generated using different adversarial attack methods. The authors report that the performance of the combined detector (detection accuracy of 85-93%) is better than detectors trained either on uncertainty or on density values, demonstrating that each feature is able to detect different qualities of adversarial features.

In [59], the idea is that the trajectory of the internal representations in the forward pass for the adversarial examples are different from that of the in-distribution examples. The internal representations of an input is embedded into the feature distance spaces which capture the relative positions of an example with respect to a given in-distribution example in the feature space. The embedding enables compact encoding of the evolution of the activations through the forward pass of the network. Hence, facilitating the search for differences between the trajectories of in-distribution and adversarial inputs. An LSTM based binary detector is trained to analyze the sequence of deep features embedded in a distance space and detect adversarial examples. The experimental results show that the detection scheme is able to detect a variety of adversarial examples targeting the ResNet-50 classifier pre-trained on the ImageNet dataset.

In [60], an expansion-based measure of intrinsic dimensionality is used as an alternative to density measure to detect adversarial example. The expansion model of dimensionality assesses the local dimensional structure of the data and characterizes the intrinsic dimensionality as a property of the datasets. The Local Intrinsic Dimensionality (LID) generalizes this concept to the local distance distribution from a reference point to its neighbors – the dimensionality of the local data submanifold in the vicinity of the reference point is revealed by the growth characteristics of the CDF. The authors use LID to characterize the intrinsic dimensionality of the regions where adversarial examples lie, and use estimates of LID to detect adversarial examples. Note that LID is a function of the nearest neighbor distances and is found to be significantly higher for the adversarial examples than the benign examples. A binary adversarial example detector is trained by using the training data to construct features for each sample, based on its LID across different layers, where the class label is assigned positive for adversarial examples and assigned negative for in-distribution examples. Experiments on several attack strategies show that LID based detector outperforms several state-of-the-art detection measures by large margins.

In [61], a three layer regression NN is used as a detector that takes logits of in-distribution and adversarial examples from a pre-trained DNN as the input and predicts the confidence value, i.e., whether the classification is normal or adversarial. The classifier used is a pre-trained CNN, trained using in-distribution datasets (MNIST and CIFAR), and the detector is trained on logits of both in-distribution and adversarial examples generated using different methods. This work shows that logits of a pre-trained network provide relevant information to detect adversarial examples.

### B. SEMI-SUPERVISED APPROACHES

Semi-supervised anomaly detection techniques utilize unlabeled contaminated data (or information) in addition to labeled instances of in-distribution class. Since, these techniques do not require the knowledge of an unlabeled instance being in-distribution or adversarial, they should be more widely applicable than supervised techniques. However, we could not find any existing semi-supervised adversarial example detection approach in the literature. Note that this may be a worthwhile direction to pursue in future research.

### C. UNSUPERVISED APPROACHES

We refer to the detection techniques as unsupervised if they only utilize in-distribution data for adversarial detection.

In [62], the probabilities of all the training images under the generative model (such as, PixelCNN) is computed. Then, for a test example, the probability density at the input is computed and its rank among the density values of all the training examples is evaluated. This rank can be used as a test statistic which gives a $p$-value for whether the example is normal or adversarial. The method improves resilience of the state-of-the-art methods against attacks and increases the detection accuracy by a significant margin. Further, the authors suggest purifying adversarial examples by searching for more probable images within a small distance of the original training ones. By utilizing $L^{\infty}$ distance, the true labels of the purified images remain unchanged. The resulting purified images have higher probability under in-distribution so that the classifier trained on normal images will have more reliable predictions on these purified images. This intuition is used to build a more effective defense against adversarial attacks.

The motivation for the method in [63] is that adversarial examples should be both (a) ''too atypical'' (i.e., have atypically low likelihood) under the density model for the DNN-predicted class, and (b) ''too typical'' (i.e., have too high a likelihood) under some class other than the DNN-predicted class. While it may seem that one requires to use two detection thresholds, they instead propose a single decision statistic that captures both requirements. Specifically, they define (a) a two-class posterior evaluated with respect to the (density-based) null model, and (b) corresponding two-class posterior evaluated via the DNN. Both deviations (''too atypical'' and ''too typical'') are captured by the Kullback-Leibler divergence decision statistic. A sample is declared adversarial if this statistic exceeds a pre-defined threshold value.

The approach in [64] performs a kNN similarity search among the deep features obtained from the training images to a given test image classified by the DNN. They then use the score assigned by a kNN classifier to the class predicted by the DNN as a measure of confidence of the classification. Note that this approach does not rely on the classification produced by the kNN classifier, but only use the score assigned to the DNN prediction as a measure of confidence. The intuition behind this approach is that while it is unlikely that a class correctly predicted by the DNN has the highest kNN score among the scores of all the classes, it is implausible that a correct classification has a very low score. Results on the ImageNet dataset show that hidden layers activations can be used to detect misclassifications caused by various attacks.

In [65], intrinsic properties of the pre-trained DNN, i.e., output distributions of the hidden neurons, are used to detect adversarial examples. Their motivation is that when the DNN incorrectly assigns an adversarial example to a specific class label, the distribution of its hidden states are very different as compared to those obtained by the normal data of the same class. They use Gaussian Mixture Model (GMM) to approximate the hidden state distribution of each class using benign training data. Likelihoods are then compared to the respective class thresholds to detect whether an example is adversarial or not. Experimental results on standard datasets (MNIST, F-MNIST, CIFAR-10) against several attack methods show that this approach can achieve state-of-the-art robustness in defending black-box and gray-box attacks.

The authors in [66] found that adversarial examples mainly exploit two attack channels: the provenance channel and the activation value distribution channel. The provenance channel implies instability of DNN output to small changes in activation values, which eventually leads to misclassification. On the other hand, the activation channel implies that while the provenance changes slightly, the activation values of a layer may be substantially different from those in the presence of benign inputs. Exploiting these observations, they propose a method that extracts two kinds of invariants (or probability distributions denoted by models), the value invariants to guard the value channel and the provenance invariants to guard the provenance channel. This is achieved by training a set of models for individual layers to describe the activation and provenance distributions only using in-distribution inputs. In other words, invariant models are trained as a One-Class Classification (OCC) problem where all training samples are positive (i.e., in-distribution inputs in this context). At test time, an input is passed through all the invariant models which provide independent predictions about whether the input induces states that violate the invariant distributions. The final result is a joint decision based on all these predictions. Extensive experiments on various attacks, datasets and models suggest that this method can achieve consistently high detection accuracy on all different types of attacks, while the performance of baseline detectors is not consistent.

In [67], the idea is that inherent distance of adversarial perturbation from the training data manifold will cause the overall network uncertainty to exceed that of the normal example. To this end, random sampling of hidden units of each layer of a pre-trained network is used to introduce randomness and the overall uncertainty of a test image is quantified in terms of the hidden layer components. A mutual information based thresholding test is used to detect adversarial examples. The performance is further improved by optimizing over the sampling probabilities to minimize uncertainty. Experiments on the CIFAR10 and the cats-and-dogs datasets on deep state-of-the-art CNNs demonstrated the importance sampling parameter optimization, which readily translate to improved attack detection.

Approaches such as [68] and [69] rely on projecting the test image to benign dataset manifold to detect adversarial examples. The underlying assumption in these approaches is that adversarial perturbations move the test image away from the benign image manifold and the effect of adversary can be nullified by projecting the images back onto the benign manifold before classifying them. As the true image manifold is unknown, various estimation techniques are used. For example, [68] use a sample approximation comprising a database of billions of natural images. On the other hand, [69] use a generative model trained on benign images to estimate the manifold. Given the estimated benign manifold, the projection is done by nearest neighbor search in [68] and gradient-based search in [69]. These methods are found to be robust against gray-box and black-box attacks where the adversary is unaware of the defense strategy.

Recently, the authors in [70] proposed an adversarial example detection technique named VisionGuard. The motivation behind this approach is the observation that adversarial images are sensitive to lossy compression based transformations. To determine if an image is adversarial, VisionGuard checks if the output of the classifier on a given test image changes significantly after feeding it a transformed version of the image under investigation.

### D. OTHER MISCELLANEOUS TECHNIQUES

Here, we discuss some other techniques that are used for adversarial example detection which do not fall in the

aforementioned categorizations of the post-hoc processing and are summarized in Table 4.

**TABLE 4. Miscellaneous papers on adversarial example detection.**

| Classification Type | Reference | Contributions |
|---|---|---|
| Uncertainty Quantification | [72] | Epistemic uncertainty is estimated using softmax variance that is used as a proxy to mutual information which is shown to perform better than other uncertainty measures to detect adversarial examples |
| Architecture design | [73] | $L_1$ norm of the DNN's prediction of the original test example and the prediction of the example after squeezing is used to detect adversarial examples |
| Architecture design | [74] | ReLU output from later layers of DNN are quantized to generate discrete code pattern for test examples that are used to detect adversarial examples |
| Architecture design | [75] | Softmax layer outputs for a test example from two pre-trained models are compared by SVM to detect whether the example is adversarial |
| Architecture design | [76] | Locality preserving hash transformations and Reconstruction error of the denoising autoencoder are used to detect adversarial examples |
| Training-based | [77] | Logit value which is the log joint distribution of test example and given label is used to detect adversarial examples |
| Training-based | [78] | Log likelihood values are computed from the class-conditional distribution of the features extracted from various layers of DNN to detect adversarial examples |
| Training-based | [79] | kNN based distance metric is combined with influence function for adversarial example detection |
| Training-based | [80] | Jaccard similarity measure that consists of robust extracted features and extracted features of the test example is used for adversarial example detection |
| Training-based | [81] | Random feature nullification introduces randomization that increases the robustness of DNN against adversarial examples |
| Training-based | [82] | Features extracted from last layers of DNN are used by three proposed methods to retrain or create histogram to detect adversarial examples |

In [72], various uncertainty measures, e.g., entropy, mutual information, softmax variance, for adversarial example detection are examined. Each of these measures capture distinct types of uncertainty and are analyzed from the perspective of adversarial example detection. The authors show that only the mutual information gets useful detection performance on adversarial examples. In fact, most of other measures of uncertainty seem to be worse than random guessing on MNIST and Kaggle dogs vs. cats classification datasets.

The approach in [73] is motivated by the observation that the DNN feature spaces are often unnecessarily large, and this provides extensive degrees of freedom for an attacker to construct adversarial examples. The authors propose to reduce the degrees of freedom for constructing adversarial examples by ''squeezing'' out unnecessary input features. Specifically, they compare the model's prediction of the original test example with its prediction of the test example after squeezing, i.e., reducing the color depth of images, and using smoothing to reduce the variation among pixels. If the original and the squeezed inputs produce substantially different predictions then the example is declared adversarial.

In [74] SafetyNet is proposed which consists of the original classifier, and an adversary detector which looks at the internal state of the later layers in the original classifier. Here, the output from the ReLU is quantized to generate a discrete code based on some set of thresholds. They claimed that different code patterns appear for natural examples and adversarial examples. An adversarial example detector (i.e., RBF-SVM) is used that compares a code produced at test time with a collection of examples, i.e., an attacker must make the network produce a code that is acceptable to the detector which is shown to be hard.

The motivation in [75] is the observation that different neural networks presented with the same adversarial example will make different mistakes. The authors propose to use such mistake patterns for adversarial example detection. Experiments on the MNIST and CIFAR10 datasets show that such detection approach generalizes well across different adversarial example generation methods.

In [76], a framework is presented for enhancing the robustness of DNN against adversarial examples. The idea is to use locality-preserving hash functions to transform examples to enhance the robustness. The hash representations of the examples are reconstructed by using a denoising auto-encoder (DAE) that enables the DNN classifier to attain the locality information in the latent space. Moreover, the DAE can detect the adversarial examples that are far from the support of the underlying training distribution.

In [77], the method improves the naive Bayes used in many generative classifiers by combining it with variational auto-encoder. They propose three adversarial example detection methods. The first two use the learned generative model as a proxy of the data manifold, and reject inputs that are far away from it. The third computes statistics for the classifier's output probability vector, and rejects inputs that lead to under-confident predictions. Experimental results suggest that deep Bayes classifiers are more robust than deep discriminative classifiers, and that the detection methods based on deep Bayes are effective against various attacks.

In [78], the authors propose to model the outputs of the various layers (deep features) with parametric probability distributions (Gaussian and Gaussian Mixture Models). At test time, the log-likelihood scores of the features of a test sample are calculated with respect to these distributions and used as anomaly score to discriminate in-distribution examples (which should have high likelihood) from adversarial examples (which should have low likelihood).

The main idea in [79] is to combine kNN based distance measure [83] with influence function which is a measure of how much a test sample classification is affected by each training sample. The motivation behind this approach is that for an in-distribution input, its kNN training samples (nearest neighbors in the embedding space) and the most helpful training samples (found using the influence function) should correlate. However, this correlation is much weaker for adversarial examples, and serves as an indication of the attack.

In [80], robust feature alignment is used to detect adversarial examples. By using an object detector, the authors first extract higher-level robust features contained in images. Next, the approach quantifies the similarity between the image's extracted features with the expected features of its

predicted class. A similarity threshold is finally used to classify a test sample as benign or adversarial.

In [81], anomaly detection is performed by introducing random feature nullification in both training and testing phases that ensures the non-deterministic nature of the DNN. Here, the randomization introduced at the test time ensures that the model's processing of the input decreases the effectiveness of the adversarial examples even if the attacker learns critical features.

In [82], three strategies are presented. First, regularized feature vectors are used to retrain the last layer of the CNN. This can be used to detect whether the input is adversarial. Second, histograms are created from the absolute values of the hidden layer outputs and are combined to form a vector which is used by the SVM to classify. Third, the input is perturbed to reinforce the parts of the input example that are ignored by the DNN which can then be used for adversarial example detection. Finally, the authors combine the best aspects of these methods to develop a more robust approach.

## VI. RELATIVE STRENGTHS AND WEAKNESS

We note that the underlying principles and detection methodologies of anomaly detection for the OOD examples and the adversarial examples are similar. Since, the relative strengths and weaknesses of these methodologies for OOD and adversarial example detection are also similar. We provide the strengths and weaknesses of the methodologies in general in this section without differentiating between OOD and adversarial example detection.

The supervised techniques usually have higher performance compared to other methods as they use the labeled examples from both normal and anomaly classes. They are able to learn the boundary from the labeled training examples and then more easily classify the unseen test examples into normal or anomaly classes. However, when training data for anomalies (the known unknowns) may not represent the full spectrum of anomalies, supervised approaches may overfit and perform poorly on unseen anomalous data (the unknown unknowns). Furthermore, due to the lack of availability of labeled anomalous examples, supervised techniques are not as popular as the semi-supervised or unsupervised techniques.

Unsupervised techniques are quite flexible and broadly applicable as they do not rely on the availability of the anomalous data and corresponding labels. The techniques learn inherent characteristics or unique features solely from in-distribution data that are useful in separating normal from anomalous examples. Unfortunately, this flexibility comes at the cost of robustness – the unsupervised techniques are very sensitive to noise, and data corruptions and are often less accurate than supervised or semi-supervised techniques.

Semi-supervised techniques exploit unlabeled data in addition to labeled in-distribution data to improve the performance of unsupervised techniques. Though, whether unlabeled data is in-distribution or anomaly is not known, it is observed that unlabeled data is helpful in improving the performance of anomaly detection. Note that unlabeled data can be obtained easily in real-world applications making semi-supervised techniques amenable in practice. These methods also suffer from the overfitting problem on unseen anomalies.

Distance-based methods, e.g., kNN approaches, require appropriate distance measure to be defined a priori. Most distance measures are not effective in high-dimension. Further, such methods are typically heuristic and require manual selection of parameters. Projection-based methods, e.g., GAN approaches, are very flexible and address the high-dimensionality challenge. However, their performance is heavily dependent on the quality of the image manifold estimate. In certain applications, it may not be easy to estimate the image manifold with sample approximation or generative modeling. Probabilistic methods, e.g., density estimation approaches, make use of the distribution of the training data or features to determine the location of the anomaly boundary. The performance of such methods is very poor in the small data regime as reliable estimates cannot be obtained. Uncertainty-based methods, e.g., entropy approaches, require a metric that is sensitive enough to detect the effects of anomalies in the dataset. Although these methods are easy to implement in practice, the performance of such methods is highly dependent on the the quality of uncertainties. Uncertainty quantification in DL is an ongoing research topic and high quality uncertainty estimates will surely improve the performance of uncertainty-based methods.

The computational complexity of these methods is another important aspect to consider. In general, probabilistic and uncertainty-based methods have computationally expensive training phases, however efficient testing. On the other hand, distance-based and projection-based methods, in general, are computationally expensive in the testing phase. Depending on the application requirements, a user should choose the most appropriate anomaly detection method.

## VII. APPLICATION DOMAINS

In this section, we briefly discuss several applications of OOD and adversarial example detection. We also suggest future research that is needed for these application domains.

### A. INTRUSION DETECTION

An Intrusion Detection System is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered. A key challenge for intrusion detection is the huge volume of data and sophisticated malicious patterns. Therefore, DL techniques are quite promising in the intrusion detection application.

In [84], a neural network based intrusion detector is trained to identify intruders. In [85], a deep hierarchical model is proposed for intrusion detection. The model is a combination of a restricted Boltzmann machine (RBM) for unsupervised feature learning and a supervised learning network called as Backpropagation network. In [86], a network intrusion model is proposed where feature learning is performed by stacking

dilated convolutional autoencoders. These features are then used to train a softmax classifier to perform supervised intrusion detection. In [87], an autoencoder based model in combination with a stochastic anomaly threshold determination method is proposed for intrusion detection. The algorithm computes the threshold using the empirical mean and standard deviation which are found from the training set via the trained autoencoder.

As mentioned earlier, these DL based systems are susceptible to both OOD and adversarial examples [88]–[90]. In [88], the authors analyze the performances of the state-of-the-art attack algorithms against DL-based intrusion detection. The susceptibility of DNNs used in the intrusion detection system is validated by experiments and the role of individual features is also explored. The authors in [89] demonstrate that an adversary can generate effective adversarial examples against DL based intrusion detection systems even when the internal information of the target model is not available to the adversary. Note that in intrusion detection applications, a large amount of labeled data corresponding to normal behavior is usually available, while labels for intrusions are not. Therefore, semi-supervised and unsupervised OOD and adversarial example detection techniques discussed in the previous sections are worthwhile directions to pursue.

### B. FRAUD DETECTION

Fraud detection refers to detection of fraudulent activities occurring in many e-commerce domains, such as, banking, insurance, law enforcement, etc. A good fraud detection system should be able to identify the fraudulent transactions accurately and should make the detection possible in real-time. There is an increase in interest in applying DL techniques in fraud detection systems. In [91], fraud detection is modeled as a sequence classification task. An LSTM is used to generate transaction sequences and incorporate aggregation functions like mean, absolute value to aggregate the learned features for fraud detection. Furthermore, in [92], feature sequencing is performed using CNNs for detecting transaction fraud. Recently, the authors in [93] analyzed the vulnerability of deep fraud detector to adversarial examples, i.e., slight perturbations in input transactions designed to fool the fraud detector. They show that the deployed deep fraud detector is highly vulnerable to attacks as the average precision is decreased from 90% to as low as 20%.

This motivates the study of the effect of unintentional and intentional anomalies in deep fraud detection systems. Techniques discussed in the previous sections will be applicable for such a problem and are potential viable solutions for designing robust deep fraud detection systems.

### C. ANOMALY DETECTION IN HEALTHCARE AND INDUSTRIAL DOMAINS

Anomaly detection in the healthcare domain tries to detect abnormal patient conditions or instrumentation errors. Anomaly detection is a very critical problem in this domain and requires high degree of accuracy. Similarly, in industrial systems like wind turbines, power plants, and storage devices which are exposed to large amounts of stress on a daily basis, it is critical to detect any damages as quickly as possible. The medical abnormalities and industrial damage are rare events and detecting them can be modeled as an anomaly detection problem. Therefore, there is a surge of interest in applying DL in both medical [94] and industrial application domains [95].

Unfortunately, similar to other DL applications, these systems are equally susceptible to OOD and adversarial examples. For example, the authors in [96] demonstrated that adversarial examples are capable of manipulating DL systems across three clinical domains: diabetic retinopathy from retinal fundoscopy, pneumothorax from chest-Xray, and melanoma from dermoscopic photographs.

This motivates the study of the effect of anomalies in DL based healthcare and industrial systems. Techniques discussed in the previous sections can be used for designing robust healthcare and damage detection systems.

### D. MALWARE DETECTION

Malware detection focuses on detecting malware software by monitoring the activity of the computer systems and classifying it as normal or anomalous. The velocity, volume, and the complexity of malware are posing new challenges to the anti-malware community. Current state-of-the-art research shows that recently, researchers started applying ML and DL methods for malware analysis and detection, [97]. In [81], malware detection is performed by introducing random feature nullification in both training and testing phases that ensures the non-deterministic nature of the DNNs. Intuitively, the non-deterministic nature ensures that the model's processing of the input decreases the effectiveness of the adversarial examples even if the attacker learns critical features. Furthermore, in [98], a stacked autoencoders model is used for malware detection. The model employs a greedy layerwise training operation for unsupervised feature learning and supervised parameter tuning. Furthermore, in [99], fake malware is generated and is learned to distinguish from the real data using a novel GAN architecture.

Authors, in [100], [101], expanded on existing adversarial example crafting algorithms to construct a highly-effective attack against malware detection models. Using the augmented adversarial crafting algorithm, authors managed to mislead the malware detection classifier for 63% of all malware samples. In [76], the authors analyzed the effect of several attacks on the Android malware classification task.

Given the susceptibility of the state-of-the-art malware detection classifiers to adversarial examples, it will be useful to utilize OOD and adversarial example detection techniques in deep malware detection systems.

### E. TIME SERIES AND VIDEO SURVEILLANCE ANOMALY DETECTION

The task of detecting anomalies in multivariate time series data is quite challenging. Hence, efficient detection of multivariate time series anomalies is critical for fault diagnostics.

RNN and LSTM based methods perform well in detecting anomalies in multivariate time series data. In [102], a generic framework based on DL for detecting anomalies in multivariate time series data is presented. Deep attention based models are used in [103] for anomaly detection for effective detection of anomalies. Many works have applied the deep learning models for video surveillance anomaly detection in [104]–[106].

Unfortunately, some recent papers [107], [108] have shown that one can design adversarial examples on time-series classifiers as well. Thus, in our opinion, future researchers should incorporate OOD and adversarial example detectors in their time series classification systems to improve the resilience and consider model robustness as an evaluative metric.

### F. ANOMALY DETECTION IN INTERNET OF THINGS (IoT)

The IoT network consists of a wireless medium to transmit data which makes it more vulnerable to intentional and unintentional anomalies. Note that an anomaly in an IoT network can cover a larger area and can have devastating effects on the network. For example, a cybercrime attack on an IoT network owned by the government or a private agency may raise serious privacy concerns or security threats. Therefore, anomaly detection is crucial in maintaining the safety and security of the IoT network. Due to the large numbers of sensors, utilizing deep learning approaches to process the large amounts of data is a rising trend. Performing post hoc anomaly detection with deep learning in the network is critical for its security. Note that there are works that have applied deep learning in IoT networks [109]–[112]. However, these papers have not considered intentional and unintentional anomalies in deep learning for IoT applications. Hence, in our opinion, future work should also incorporate both OOD and adversarial example detectors in IoT networks to improve their overall robustness.

## VIII. CONCLUSION AND OPEN QUESTIONS

In this survey, we discussed various techniques for detecting OOD and adversarial examples given a pre-trained DNN. For each category of anomaly detection techniques, we discussed the strengths and weaknesses of these techniques. Finally, we discussed various application domains where the post-hoc processing, as well as, training based anomaly detection techniques are applicable.

There are several open issues and worthwhile future directions for further research. Several of these are identified by analyzing and comparing existing literature and the research considered in this survey.

### A. METHODS

We classified anomaly detection algorithms based on the availability of the labels of anomalous examples and the type of metrics used. Based on the availability of the labels, the techniques are classified as supervised, semi-supervised, and unsupervised. Based on the type of metric, the techniques are classified as probability-based, distance-based, projection-based, and uncertainty-based. Each category of methods have their own strengths and weaknesses, and faces different challenges as discussed in Section VI. We conjecture that exploration of ensemble detection approaches can be a worthwhile future direction. The ensemble approach combines outputs of multiple detectors offering complementary strengths into a single one, thus yielding better performance compared to using individual detectors.

### B. DEFINING ANOMALIES

Majority of the research on detecting OOD and adversarial examples in DL focuses on detecting independent anomalies (e.g., adversarial examples generated independently from one another). However, anomalous behaviors can be much more complex requiring more sophisticated detection approaches than currently available. An example of this is discussed in [113] where a simple correlated anomaly generation approach is discussed. It is shown that current defenses are not capable of defending against this simple scheme. Further, defining collective and contextual anomalies [114] in the context of OOD and adversarial examples in DL can be very interesting and detecting them will certainly require the development of a new class of detectors. Also, we want to emphasize that it is important for future research on anomaly detection to be cognizant of the fact that anomalies may not adhere to our definitions and assumptions and can have extremely complex unknown behavior. This is similar to the concept of unknown-unknowns [115]. We believe that the research on domain generalization [116] and meta learning [117] can be used to solve some of these issues.

### C. GOING BEYOND IMAGE CLASSIFICATION

Most of the papers discussed in this survey (and in the literature) focus on the detection of anomalous examples in DNN based image classification problems. However, in recent years there has been a surge of interest in applying DL on other data types, e.g. text, graphs, trees, manifolds etc. These data types are ubiquitous in several high-impact applications including bioinformatics, neuroscience, social sciences, and molecular chemistry. Unfortunately, DL approaches in these data types also suffer from the existence of OOD and adversarial examples [118], [119]. Post-hoc detection of such anomalies has not received much attention. Furthermore, going beyond classification problems and exploring the design and the detection of anomalies in DL based object detection, control, and planning problems can be a high-impact future research direction. Anomaly segmentation [120] also deserves more attention.

### D. PERFORMANCE EVALUATION

Reliably evaluating the performance of OOD and adversarial example detection methods has proven to be extremely difficult. Previous evaluation methods are found to be ineffective and performing incorrect or incomplete evaluations [121], [122]. Absence of a standard definition for anomalies makes this problem very challenging. Furthermore, as anomalies

become more sophisticated, it may become even harder to reliably evaluate the detection performance. Majority of current approaches evaluate the performance of anomaly detectors on OOD and adversarial examples. Assuming that training data may not represent the full spectrum of anomalies, this evaluation approach raises the risk of overfitting. Ideally, one should adopt an evaluation method that can assess the detection performance on adaptive and unseen anomalies (the unknown unknowns) over methods that only can assess the detection performance on previously seen anomalies (the known unknowns). Due to these reasons, there is an immediate need for designing principled benchmarks to reliably evaluate the anomaly detection performance [122], [123].

### E. THEORETICAL ANALYSIS AND FUNDAMENTAL LIMITS

Finally, we need to make efforts on the theoretical front to understand the nature of the anomaly detection problem in DL-based systems. In the recent past, a pattern has emerged in which the majority of heuristics based defenses (both posthoc detection and training based) are easily broken by new attacks [121], [124]. Therefore, the development of a coherent theory and methodology that guides practical design with guarantees for anomaly detection in DL-based systems [125]–[128], and fundamental characterizations of the existence of adversarial examples [129] is of utmost importance. How to leverage special learning properties such as the spatial and temporal consistencies to identify anomalous examples [130], [131] are also worth further exploration.

To summarize, OOD and adversarial example detection in DL-based systems is an open problem. We highlighted several aspects of the problem to be understood on both theoretical and algorithmic front to improve the effectiveness and feasibility of anomaly detection. We hope that this survey will provide a comprehensive understanding of the different approaches, show the bigger picture of the problem, and suggest few promising directions for researchers to pursue in further investigations on the anomaly detection in DL-based systems.

## REFERENCES

[1] M. Zamini and S. M. H. Hasheminejad, "A comprehensive survey of anomaly detection in banking, wireless sensor networks, social networks, and healthcare," *Intell. Decis. Technol.*, vol. 13, no. 2, pp. 229–270, May 2019.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[3] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "Review: A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, Jun. 2014.

[4] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: https://arxiv.org/abs/1901.03407

[5] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Comput. Sci.*, vol. 60, pp. 708–713, 2015.

[6] M. Salehi and L. Rashidi, "A survey on anomaly detection in evolving data: [With application to forest fire risk prediction]," *ACM SIGKDD Explor. Newslett.*, vol. 20, no. 1, pp. 13–23, May 2018.

[7] L. Kalinichenko, I. Shanin, and I. Taraban, "Methods for anomaly detection: A survey," in *Proc. All-Russian Conf. Digit. Libraries, Adv. Methods Technol., Digit. Collections (RCDL)*, 2014, pp. 20–25.

[8] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007.

[9] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.

[10] G. Muruti, F. A. Rahim, and Z.-A. B. Ibrahim, "A survey on anomalies detection techniques and measurement methods," in *Proc. IEEE Conf. Appl., Inf. Netw. Secur. (AINS)*, Nov. 2018, pp. 81–86.

[11] T. Dunning and E. Friedman, *Practical Machine Learning: A New Look at Anomaly Detection*. Newton, MA, USA: O'Reilly Media, 2014.

[12] K. Mehrotra, C. Mohan, and H. Huang, *Anomaly Detection Principles and Algorithms*. Cham, Switzerland: Springer, 2017.

[13] C. Aggarwal, *Outlier Analysis*. New York, NY, USA: Springer, 2016.

[14] M. Bhuyan, D. Bhattacharyya, and J. Kalita, *Network Traffic Anomaly Detection and Prevention: Concepts, Techniques, and Tools*. Berlin, Germany: Springer, 2017.

[15] L. Zhang, "Transfer adaptation learning: A decade survey," 2019, *arXiv:1903.04687*. [Online]. Available: http://arxiv.org/abs/1903.04687

[16] R. Domingues, P. Michiardi, J. Barlet, and M. Filippone, "A comparative evaluation of novelty detection algorithms for discrete sequences," 2019, *arXiv:1902.10940*. [Online]. Available: http://arxiv.org/abs/1902.10940

[17] S. Marsland, "Novelty detection in learning systems," *Neural Comput. Surv.*, vol. 3, no. 2, pp. 157–195, 2003.

[18] M.-R. Bouguelia, S. Nowaczyk, and A. H. Payberah, "An adaptive algorithm for anomaly and novelty detection in evolving data streams," *Data Mining Knowl. Discovery*, vol. 32, no. 6, pp. 1597–1633, Nov. 2018.

[19] P. Oberdiek, M. Rottmann, and H. Gottschalk, "Classification uncertainty of deep neural networks based on gradient information," *CoRR*, vol. abs/1805.08440, 2018.

[20] 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.*, 2018, pp. 7167–7177.

[21] Y. Bahat and G. Shakhnarovich, "Confidence from invariance to image transformations," 2018, *arXiv:1804.00657*. [Online]. Available: http://arxiv.org/abs/1804.00657

[22] H. Jiang, B. Kim, M. Guan, and M. Gupta, "To trust or not to trust a classifier," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5541–5552.

[23] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," 2019, *arXiv:1906.02845*. [Online]. Available: http://arxiv.org/abs/1906.02845

[24] Q. Yu and K. Aizawa, "Unsupervised out-of-distribution detection by maximum classifier discrepancy," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 9518–9526.

[25] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2016, *arXiv:1610.02136*. [Online]. Available: https://arxiv.org/abs/1610.02136

[26] 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.*, 2018.

[27] W. Lawson, E. Bekele, and K. Sullivan, "Finding anomalies with generative adversarial networks for a patrolbot," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 484–485.

[28] X. W. Wenhu Chen, Y. Shen, and W. Wang, "Enhancing the robustness of prior network in out-of-distribution detection," 2018, *arXiv:1811.07308*. [Online]. Available: https://arxiv.org/abs/1811.07308

[29] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9758–9769.

[30] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, "Improving reconstruction autoencoder Out-of-distribution detection with mahalanobis distance," 2018, *arXiv:1812.02765*. [Online]. Available: http://arxiv.org/abs/1812.02765

[31] P. Schulam and S. Saria, "Can you trust this prediction? Auditing pointwise reliability after learning," in *Proc. Mach. Learn. Res.*, vol. 89, Apr. 2019, pp. 1022–1031.

[32] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: An extension of MNIST to handwritten letters," 2017, *arXiv:1702.05373*. [Online]. Available: http://arxiv.org/abs/1702.05373

[33] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[34] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.
[35] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011.
[36] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*. [Online]. Available: http://arxiv.org/abs/1605.07146
[37] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[38] 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.*, 2019, pp. 15637–15648.
[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[40] J. Elson, J. J. Douceur, J. Howell, and J. Saul, "Asirra: A CAPTCHA that exploits interest-aligned manual image categorization," in *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 7, 2007, pp. 366–374.
[41] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.
[42] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1861–1869.
[43] H. Choi, E. Jang, and A. A. Alemi, "WAIC, but why? Generative ensembles for robust anomaly detection," 2018, *arXiv:1810.01392*. [Online]. Available: http://arxiv.org/abs/1810.01392
[44] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, and Z. Nado, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13969–13980.
[45] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 797–806.
[46] E. Techapanurak and T. Okatani, "Hyperparameter-free out-of-distribution detection using softmax of scaled cosine similarity," 2019, *arXiv:1905.10628*. [Online]. Available: https://arxiv.org/abs/1905.10628
[47] W. Wang, M. Zhao, and J. Wang, "Effective Android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 8, pp. 3035–3043, Aug. 2019.
[48] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke, "Out-of-distribution detection using an ensemble of self supervised leave-out classifiers," 2018, *arXiv:1809.03576*. [Online]. Available: https://arxiv.org/abs/1809.03576
[49] P. Filonov, F. Kitashov, and A. Lavrentyev, "RNN-based early cyber-attack detection for the tennessee eastman process," 2017, *arXiv:1709.02232*. [Online]. Available: http://arxiv.org/abs/1709.02232
[50] 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.*, 2018.
[51] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," 2018, *arXiv:1812.04606*. [Online]. Available: http://arxiv.org/abs/1812.04606
[52] J. Song, Y. Song, and S. Ermon, "Unsupervised Out-of-Distribution detection with batch normalization," 2019, *arXiv:1910.09115*. [Online]. Available: http://arxiv.org/abs/1910.09115
[53] R. Roady, T. L. Hayes, R. Kemker, A. Gonzales, and C. Kanan, "Are Out-of-Distribution detection methods effective on large-scale datasets?" 2019, *arXiv:1910.14034*. [Online]. Available: http://arxiv.org/abs/1910.14034
[54] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha, "Robust out-of-distribution detection for neural networks," 2020, *arXiv:2003.09711*. [Online]. Available: http://arxiv.org/abs/2003.09711
[55] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*. [Online]. Available: http://arxiv.org/abs/1312.6199
[56] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*. [Online]. Available: http://arxiv.org/abs/1412.6572
[57] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," 2017, *arXiv:1702.04267*. [Online]. Available: http://arxiv.org/abs/1702.04267
[58] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017, *arXiv:1703.00410*. [Online]. Available: http://arxiv.org/abs/1703.00410
[59] F. Carrara, R. Becarelli, R. Caldelli, F. Falchi, and G. Amato, "Adversarial examples detection in features distance spaces," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018.
[60] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," 2018, *arXiv:1801.02613*. [Online]. Available: http://arxiv.org/abs/1801.02613
[61] J. Aigrain and M. Detyniecki, "Detecting adversarial examples and other misclassifications in neural networks by introspection," 2019, *arXiv:1905.09186*. [Online]. Available: http://arxiv.org/abs/1905.09186
[62] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *CoRR*, vol. abs/1710.10766, 2017.
[63] D. Miller, Y. Wang, and G. Kesidis, "When not to classify: Anomaly detection of attacks (ADA) on DNN classifiers at test time," *Neural Comput.*, vol. 31, no. 8, pp. 1624–1670, 2017.
[64] F. Carrara, F. Falchi, R. Caldelli, G. Amato, R. Fumarola, and R. Becarelli, "Detecting adversarial example attacks to deep neural networks," in *Proc. 15th Int. Workshop Content-Based Multimedia Indexing*, 2017, p. 38.
[65] Z. Zheng and P. Hong, "Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7913–7922.
[66] S. Ma, Y. Liu, G. Tao, W.-C. Lee, and X. Zhang, "NIC: Detecting adversarial samples with neural network invariant checking," in *Proc. NDSS*, 2019.
[67] F. Sheikholeslami, S. Jain, and G. B. Giannakis, "Minimum uncertainty based detection of adversaries in deep neural networks," 2019, *arXiv:1904.02841*. [Online]. Available: http://arxiv.org/abs/1904.02841
[68] A. Dubey, L. van der Maaten, Z. Yalniz, Y. Li, and D. Mahajan, "Defense against adversarial images using Web-scale nearest-neighbor search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8767–8776.
[69] R. Anirudh, J. J. Thiagarajan, B. Kailkhura, and T. Bremer, "MimicGAN: Robust projection onto image manifolds with corruption mimicking," 2019, *arXiv:1912.07748*. [Online]. Available: http://arxiv.org/abs/1912.07748
[70] Y. Kantaros, T. Carpenter, S. Park, R. Ivanov, S. Jang, I. Lee, and J. Weimer, "VisionGuard: Runtime detection of adversarial inputs to perception systems," 2020, *arXiv:2002.09792*. [Online]. Available: http://arxiv.org/abs/2002.09792
[71] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," 2017, *arXiv:1702.06280*. [Online]. Available: http://arxiv.org/abs/1702.06280
[72] L. Smith and Y. Gal, "Understanding measures of uncertainty for adversarial example detection," 2018, *arXiv:1803.08533*. [Online]. Available: http://arxiv.org/abs/1803.08533
[73] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," 2017, *arXiv:1704.01155*. [Online]. Available: http://arxiv.org/abs/1704.01155
[74] J. Lu, T. Issaranon, and D. Forsyth, "SafetyNet: Detecting and rejecting adversarial examples robustly," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 446–454.
[75] J. Monteiro, I. Albuquerque, Z. Akhtar, and T. H. Falk, "Generalizable adversarial examples detection based on bi-model decision mismatch," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 2839–2844.
[76] D. Li, R. Baral, T. Li, H. Wang, Q. Li, and S. Xu, "HashTran-DNN: A framework for enhancing robustness of deep neural networks against adversarial malware samples," 2018, *arXiv:1809.06498*. [Online]. Available: http://arxiv.org/abs/1809.06498
[77] Y. Li, J. Bradshaw, and Y. Sharma, "Are generative classifiers more robust to adversarial attacks?" 2018, *arXiv:1802.06552*. [Online]. Available: http://arxiv.org/abs/1802.06552
[78] N. A. Ahuja, I. Ndiour, T. Kalyanpur, and O. Tickoo, "Probabilistic modeling of deep features for Out-of-Distribution and adversarial detection," 2019, *arXiv:1909.11786*. [Online]. Available: http://arxiv.org/abs/1909.11786
[79] G. Cohen, G. Sapiro, and R. Giryes, "Detecting adversarial samples using influence functions and nearest neighbors," 2019, *arXiv:1909.06872*. [Online]. Available: http://arxiv.org/abs/1909.06872
[80] S. Freitas, S.-T. Chen, Z. Wang, and D. Horng Chau, "UnMask: Adversarial detection and defense through robust feature alignment," 2020, *arXiv:2002.09576*. [Online]. Available: http://arxiv.org/abs/2002.09576
[81] Q. Wang, W. Guo, K. Zhang, X. Xing, C. L. Giles, and X. Liu, "Random feature nullification for adversary resistant deep architecture," 2016, *arXiv:1610.01239*. [Online]. Available: http://arxiv.org/abs/1610.01239

[82] S. Pertigkiozoglou and P. Maragos, "Detecting adversarial examples in convolutional neural networks," 2018, *arXiv:1812.03303*. [Online]. Available: http://arxiv.org/abs/1812.03303
[83] C. Sitawarin and D. Wagner, "Defending against adversarial examples with K-Nearest neighbor," 2019, *arXiv:1906.09525*. [Online]. Available: http://arxiv.org/abs/1906.09525
[84] J. Ryan, M.-J. Lin, and R. Miikkulainen, "Intrusion detection with neural networks," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 1998, pp. 943–949.
[85] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Proc. 2nd Int. Conf. Adv. Cloud Big Data*, Nov. 2014, pp. 247–252.
[86] Y. Yu, J. Long, and Z. Cai, "Network intrusion detection through stacking dilated convolutional autoencoders," *Secur. Commun. Netw.*, vol. 2017, Nov. 2017, Art. no. 4184196.
[87] R. C. Aygun and A. G. Yavuz, "Network anomaly detection with stochastically improved autoencoder based models," in *Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Jun. 2017, pp. 193–198.
[88] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018.
[89] K. Yang, J. Liu, C. Zhang, and Y. Fang, "Adversarial examples against the deep learning based network intrusion detection systems," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2018, pp. 559–564.
[90] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks," 2019, *arXiv:1905.05137*. [Online]. Available: http://arxiv.org/abs/1905.05137
[91] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, "Sequence classification for credit-card fraud detection," *Expert Syst. Appl.*, vol. 100, pp. 234–245, Jun. 2018.
[92] Z. Zhang, X. Zhou, X. Zhang, L. Wang, and P. Wang, "A model based on convolutional neural network for online transaction fraud detection," *Secur. Commun. Netw.*, vol. 2018, pp. 5680264:1–5680264:9, Aug. 2018.
[93] Q. Guo, Z. Li, B. An, P. Hui, J. Huang, L. Zhang, and M. Zhao, "Securing the deep fraud detector in large-scale E-commerce platform via adversarial machine learning approach," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 616–626.
[94] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Med.*, vol. 25, no. 1, pp. 24–29, Jan. 2019.
[95] W. Nash, T. Drummond, and N. Birbilis, "A review of deep learning in the study of materials degradation," *NPJ Mater. Degradation*, vol. 2, no. 1, pp. 1–12, Dec. 2018.
[96] S. G. Finlayson, H. Won Chung, I. S. Kohane, and A. L. Beam, "Adversarial attacks against medical deep learning systems," 2018, *arXiv:1804.05296*. [Online]. Available: http://arxiv.org/abs/1804.05296
[97] H. Rathore, S. Agarwal, S. K. Sahay, and M. Sewak, "Malware detection using machine learning and deep learning," in *Proc. Int. Conf. Big Data Anal.* Cham, Switzerland: Springer, 2018, pp. 402–411.
[98] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "DL4MD: A deep learning framework for intelligent malware detection," in *Proc. Int. Conf. Data Mining (DMIN)*, 2016, p. 61.
[99] J.-Y. Kim, S.-J. Bu, and S.-B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," *Inf. Sci.*, vols. 460–461, pp. 83–102, Sep. 2018.
[100] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *Proc. Eur. Symp. Res. Comput. Secur.* Springer, 2017, pp. 62–79.
[101] O. Suciu, S. E. Coull, and J. Johns, "Exploring adversarial examples in malware detection," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2019, pp. 8–14.
[102] T. S. Buda, B. Caglayan, and H. Assem, "DeepAD: A generic framework based on deep learning for time series anomaly detection," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2018, pp. 577–588.
[103] Y. Yuan, G. Xun, F. Ma, Y. Wang, N. Du, K. Jia, L. Su, and A. Zhang, "MuVAN: A multi-view attention network for multivariate temporal data," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 717–726.
[104] M. Gutoski, N. M. R. Aquino, M. Ribeiro, A. Lazzaretti, and H. S. Lopes, "Detection of video anomalies using convolutional autoencoders and one-class support vector machines," in *Proc. 13th Brazilian Congr. Comput. Intell.*, 2017.
[105] I. Ben-Ari and R. Shwartz-Ziv, "Attentioned convolutional LSTM inpaintingnetwork for anomaly detection in videos," 2018, *arXiv:1811.10228*. [Online]. Available: http://arxiv.org/abs/1811.10228
[106] G. Tripathi, K. Singh, and D. K. Vishwakarma, "Convolutional neural networks for crowd behaviour analysis: A survey," *Vis. Comput.*, vol. 35, no. 5, pp. 753–776, May 2019.
[107] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Adversarial attacks on deep neural networks for time series classification," 2019, *arXiv:1903.07054*. [Online]. Available: http://arxiv.org/abs/1903.07054
[108] F. Karim, S. Majumdar, and H. Darabi, "Adversarial attacks on time series," 2019, *arXiv:1902.10755*. [Online]. Available: http://arxiv.org/abs/1902.10755
[109] T. Yu, X. Wang, and A. Shami, "UAV-enabled spatial data sampling in large-scale IoT systems using denoising autoencoder neural network," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1856–1865, Apr. 2019.
[110] P. Zhang, X. Kang, D. Wu, and R. Wang, "High-accuracy entity state prediction method based on deep belief network toward IoT search," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 492–495, Apr. 2019.
[111] J. Liang, X. Yu, and H. Li, "Collaborative energy-efficient moving in Internet of Things: Genetic fuzzy tree versus neural networks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6070–6078, Aug. 2019.
[112] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, and X. Shen, "Deep reinforcement learning for autonomous Internet of Things: Model, applications and challenges," *IEEE Commun. Surveys Tuts.*, early access, Apr. 16, 2020, doi: 10.1109/COMST.2020.2988367.
[113] I. Goodfellow, "A research agenda: Dynamic models to defend against correlated attacks," 2019, *arXiv:1903.06293*. [Online]. Available: http://arxiv.org/abs/1903.06293
[114] H. Zhang, K. Nian, T. F. Coleman, and Y. Li, "Spectral ranking and unsupervised feature selection for point, collective, and contextual anomaly detection," *Int. J. Data Sci. Anal.*, vol. 9, pp. 57–75, Dec. 2018.
[115] H. Lakkaraju, E. Kamar, R. Caruana, and E. Horvitz, "Identifying unknown unknowns in the open world: Representations and policies for guided exploration," in *Proc. 31st AAAI Conf. Artif. Intell.*, vol. 1, 2017, p. 2.
[116] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 10–18.
[117] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artif. Intell. Rev.*, vol. 18, no. 2, pp. 77–95, 2002.
[118] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," 2019, *arXiv:1909.08072*. [Online]. Available: http://arxiv.org/abs/1909.08072
[119] D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song, "Pretrained transformers improve Out-of-Distribution robustness," 2020, *arXiv:2004.06100*. [Online]. Available: http://arxiv.org/abs/2004.06100
[120] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. Song, "A benchmark for anomaly segmentation," 2019, *arXiv:1911.11132*. [Online]. Available: http://arxiv.org/abs/1911.11132
[121] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 3–14.
[122] A. Shafaei, M. Schmidt, and J. J. Little, "A less biased evaluation of Out-of-distribution sample detectors," 2018, *arXiv:1809.04729*. [Online]. Available: http://arxiv.org/abs/1809.04729
[123] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," 2019, *arXiv:1902.06705*. [Online]. Available: http://arxiv.org/abs/1902.06705
[124] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," 2018, *arXiv:1802.00420*. [Online]. Available: http://arxiv.org/abs/1802.00420
[125] I. Shumailov, Y. Zhao, R. Mullins, and R. Anderson, "Towards certifiable adversarial sample detection," 2020, *arXiv:2002.08740*. [Online]. Available: http://arxiv.org/abs/2002.08740
[126] S. Vernekar, A. Gaurav, T. Denouden, B. Phan, V. Abdelzad, R. Salay, and K. Czarnecki, "Analysis of confident-classifiers for Out-of-distribution detection," 2019, *arXiv:1904.12220*. [Online]. Available: http://arxiv.org/abs/1904.12220
[127] Y. Zhang, W. Liu, Z. Chen, J. Wang, Z. Liu, K. Li, and H. Wei, "Out-of-Distribution detection with distance guarantee in deep generative models," 2020, *arXiv:2002.03328*. [Online]. Available: http://arxiv.org/abs/2002.03328
[128] A. Meinke and M. Hein, "Towards neural networks that provably know when they don't know," Tech. Rep., 2019.
[129] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein, "Are adversarial examples inevitable?" 2018, *arXiv:1809.02104*. [Online]. Available: http://arxiv.org/abs/1809.02104

[130] C. Xiao, R. Deng, B. Li, F. Yu, M. Liu, and D. Song, "Characterizing adversarial examples based on spatial consistency information for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 217–234.

[131] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," 2018, *arXiv:1809.10875*. [Online]. Available: http://arxiv.org/abs/1809.10875

**SAIKIRAN BULUSU** (Graduate Student Member, IEEE) received the B.Tech. degree from the Mahatma Gandhi Institute of Technology, Hyderabad, in 2009, and the M.Tech. degree in communication engineering from the IIT Madras, in 2012. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY, USA. His research interests include the design and analysis of robust machine learning and distributed optimization algorithms.

**BHAVYA KAILKHURA** (Member, IEEE) is currently a Research Staff with the Lawrence Livermore National Labs, Livermore, CA, USA. His research interests include adversarial machine learning, robust statistics and control, and high-dimensional data analytics. He was the runner-up for Best Student Paper Award at the IEEE Asilomar Conference on Signals, Systems and Computers, in 2014. He received the All University Doctoral Prize 2017 by Syracuse University for superior achievement in completed dissertations. He was a recipient of the SPS Travel Grant Award and the Deputy Director for S&T Excellence in Publication Award, LLNL, in 2019. He serves as an Associate Editor for a guest issue of Frontiers in Big Data and Artificial Intelligence on Safe and Trustworthy Machine Learning. He has Co-Chaired workshops on human in the loop machine learning at GlobalSIP 2019 and ICASSP 2018. He also served on Technical Program Committees for the IEEE Workshop on Machine Learning and Artificial Intelligence for Multimedia Creation (ICME 2018), the ACM Workshop on Distributed Information Processing in Wireless Networks (MobiHoc 2018), the Workshop on Adversarial Learning Methods for Machine Learning and Data Mining (KDD 2019).

**BO LI** (Member, IEEE) is currently an Assistant Professor with the Department of Computer Science, University of Illinois at Urbana–Champaign. She has designed several robust learning algorithms against adversarial behaviors, a scalable framework for achieving robustness for a range of learning methods, and a privacy preserving data publishing systems. Her research interests include theoretical and practical aspects of security, machine learning, privacy, game theory, adversarial deep learning, generative models, and designing scalable robust machine learning models against unrestricted adversarial attacks. Her work has been featured by major publications and media outlets, such as *Nature*, *Wired*, *Fortune*, and the *IEEE Spectrum*. She was a recipient of the Symantec Research Labs Fellowship and the MIT Technology Review TR-35 Award.

**PRAMOD K. VARSHNEY** (Life Fellow, IEEE) received the B.S. degree (Hons.) in electrical engineering and computer science and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana–Champaign, USA, in 1972, 1974, and 1976, respectively, and the Ph.D. degree (honoris causa) from Drexel University, in 2014. Since 1976, he has been with Syracuse University, Syracuse, NY, USA, where he is currently a Distinguished Professor of electrical engineering and computer science and the Director of the Center for Advanced Systems and Engineering (CASE). He is a member of Tau Beta Pi. He was elected to the grade of Fellow of the IEEE, in 1997, for his contributions in the area of distributed detection and data fusion. He received the 1981 ASEE Dow Outstanding Young Faculty Award. In 2000, he received the Third Millennium Medal from the IEEE and Chancellor's Citation for exceptional academic achievement at Syracuse University, the IEEE 2012 Judith A. Resnik Award, the ECE Distinguished Alumni Award from the University of Illinois, in 2015 and the ISIF's Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion, in 2018. He is on the Editorial Board of the *Journal on Advances in Information Fusion*. He has served on the editorial boards of IEEE TRANSACTIONS OF SIGNAL PROCESSING, the *IEEE Signal Processing Magazine*, and so on. He was the President of International Society of Information Fusion, in 2001.

**DAWN SONG** (Fellow, IEEE) received the Ph.D. degree from UC Berkeley. Prior to joining UC Berkeley as a Faculty Member, she was a Faculty Member with Carnegie Mellon University, from 2002 to 2007. She is also a serial Entrepreneur and has been named on the Female Founder 100 List by Inc., and Wired25 List of Innovators. She is currently a Professor with the Department of Electrical Engineering and Computer Science, UC Berkeley. Her research interests include AI and deep learning, security, and privacy. She is an ACM Fellow. She is ranked the most cited scholar in computer security (AMiner Award). She was a recipient of various awards including the MacArthur Fellowship, the Guggenheim Fellowship, the NSF CAREER Award, the Alfred P. Sloan Research Fellowship, the MIT Technology Review TR-35 Award, and the Best Paper Awards from top conferences in Computer Security and Deep Learning.