Statistics and Its Interface Volume 17 (2024) 91–106

Abnormal sample detection based on robust Mahalanobis distance estimation in adversarial machine learning*

Wan Tian, Lingyue Zhang, and Hengjian Cui†

This paper addresses the problem of abnormal sample detection in deep learning-based computer vision, focusing on two types of abnormal samples: outlier samples and adversarial samples. The presence of these abnormal samples can significantly degrade the performance and robustness of deep learning models, posing security risks in critical areas. To address this, we propose a method that combines robust Mahalanobis distance (RMD) estimation with a pre-trained convolutional neural networks (CNNs) model. The RMD estimation involves using minimum covariance matrix determinant (MCD), T-type, and S estimators. Furthermore, we theoretically analyze the breakdown point and influence function of the T-type estimator. To evaluate the effectiveness and robustness of our method, we utilize public datasets, CNN models, and adversarial sample generation algorithms commonly employed in the field. The experimental results demonstrate the effectiveness of our algorithm in detecting abnormal samples.

AMS 2000 subject classifications: Primary 62H30, 62G35; secondary 62H35.
Keywords and phrases: Abnormal sample detection, MCD estimator, T-type estimator, Breakdown point, Influence function.

1. INTRODUCTION

Deep learning models in various fields, such as natural language processing [25], recommender systems [6], computer vision [11], fraud and malware detection [29], finance [13], and autonomous driving [33], have made significant advancements in recent years. The improvements in hardware computing power and the availability of large-scale data have contributed to these advancements. In computer vision, tasks such as object detection [44], instance segmentation [12], image classification [11], and semantic segmentation [4] have achieved state-of-the-art performance using CNNs. In some cases, CNNs have even surpassed human performance [22].

*This paper is for the special issue celebrating Professor Lincheng Zhao's 80th birthday.
†Corresponding author.

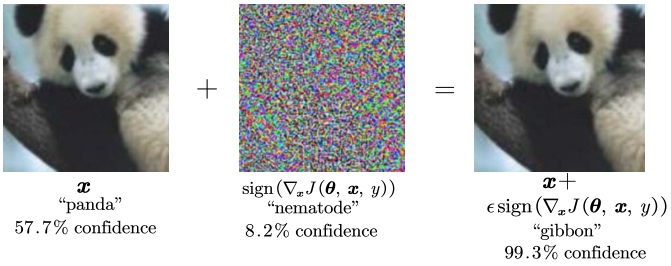


Figure 1. Example of adversarial sample generation. By adding subtle disturbances to the image, "panda" was mistakenly classified as "gibbon" with great confidence. (Image Credit: [9]).

In 2013, a "counter-intuitive" phenomenon was discovered by Szegedy et al. [39] in the field of computer vision. They observed that adding an imperceptible disturbance to an original image $\boldsymbol{x}$, specifically using $\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$, could generate a new sample $\boldsymbol{x}' = \boldsymbol{x} + \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ that can cause a deep learning model to confidently misclassify it. These new samples are known as adversarial samples. Figure 1 provides an illustration of an adversarial sample.

The assailant orchestrates a manipulation of the input sample by artfully constructing a subtle perturbation, thereby inducing the image recognition system, which relies on deep neural networks, to yield erroneous outcomes. These samples, triggering misclassification within the deep learning system, are commonly referred to as adversarial samples. The precise definition is as follows:

> Adversarial examples definition [43]: *Adversarial examples are inputs to machine learning models that an attacker intentionally designed to cause the model to make mistakes.*

Adversarial examples are not confined to images; they also exist in other domains, including text and speech. Modifying different aspects in these domains can similarly lead to misclassification. For instance, altering an edge in a graph CNN can cause a graph neural network to incorrectly match nodes [8], while modifying a segment of text can result in text classification errors [24]. An illustrative example is provided below.

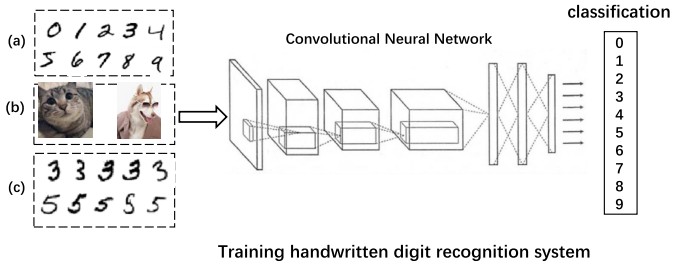


Figure 2. Use the dataset MNIST to train a handwritten digit recognition system. (a) in-distribution sample with the correct label, (b) OOD with label belonging to $\{0, 1, 2, \ldots, 9\}$*, (c) in-distribution sample with the wrong label.*

Original text: South Africa's historic Soweto township marks its 100th birthday on Tuesday in a mood of optimism.$\Longrightarrow$ *classification: 57%* ***World***

Adversarial text: South Africa's historic Soweto township marks its 100th birthday on Tuesday in a mooP of optimism.$\Longrightarrow$ *classification 95%* ***Sci/Tech***

Adversarial machine learning (AML) has emerged as a crucial area of research to tackle the security challenges posed to machine learning systems [19]. The significance of this field lies in the development of robust and effective defense mechanisms against adversarial attacks. In response to these challenges, the adoption of Bayesian perspectives is proposed as a promising approach to bolster the resilience of ML models, providing a foundation for enhanced security measures.

In addition to adversarial examples, the performance of deep learning models can also be influenced by Out-of-Distribution (OOD) samples [14]. Consider a scenario where a training set for a handwritten digit recognition system, utilizing the MNIST dataset and a CNN, includes non-digit images (such as animal images) or images with incorrect labels. This incorporation of OOD samples can have a substantial impact on the model's performance. An example of an OOD sample is depicted in Figure 2. The presence of adversarial examples and outliers highlights the fragility and instability of deep learning models [39]. In the current era, with the widespread utilization of deep learning technology, particularly in safety-critical domains like autonomous driving and intelligent medical diagnosis [28], accurately determining whether a sample is abnormal or in-distribution before feeding it into the deep learning system becomes paramount.

The defense against adversarial samples in computer vision can be categorized based on two factors: the defense objective and the defense approach [31]. These two classification methods can be further divided as follows. Defense objective: (a) Proactive defense: This type of defense primarily focuses on the training stage, aiming to minimize the impact of adversarial samples on the model's training process. The objective is to ensure that the model becomes more robust against adversarial attacks. (b) Reactive defense: In this case, the objective is to assess samples before they enter the model, preventing abnormal samples from being fed into the deep learning system. If a sample is classified as normal, it will proceed through the machine learning system; otherwise, further processing will be implemented. The defense approach can encompass a variety of strategies including gradient masking, auxiliary detection models, statistical methods, preprocessing techniques, classifier integration techniques, adversarial training, and defensive distillation. Although currently deployed defenses against abnormal samples have yielded certain positive results, numerous limitations and challenges persist. These include factors like protracted training times, the presence of a large number of adversarial samples, the challenge of concurrently detecting outliers and adversarial samples, and the limited capacity to defend against specific types of adversarial samples. These complexities emphasize the need for ongoing research to advance methods of defense against abnormal samples.

Currently, several effective adversarial detection algorithms have been proposed in the literature. Hendrycks and Gimpel [14] introduced the use of the maximum value of the posterior distribution of a classifier to detect outlier samples. Their approach also involved processing the input and output of the deep learning model to enhance its performance. Ma et al. [30] proposed the utilization of local intrinsic dimensionality (LID) for outlier sample detection. Meanwhile, Lee et al. [26] employed a softmax classifier based on the Mahalanobis distance and pre-trained CNN models to detect both adversarial and outlier samples. More details can be found in [31] and the references therein. The method proposed by Lee et al. [26] based on Mahalanobis distance has been supported by numerous experiments. However, Lee et al. [26] did not consider the presence of outliers in the dataset, which can greatly impact the estimation of Mahalanobis distance. In this paper, we propose to make Mahalanobis distance robust by using three robust estimators and demonstrate its advantages in various scenarios.

The organization of the paper is as follows. In Section 2, we introduce common adversarial example generation algorithms and the datasets used in the experiments. Section 3 presents two robust estimators along with their corresponding breakdown point theory and bounded influence function properties. Section 4 validates the effectiveness of the proposed method through extensive experiments on outlier detection and adversarial detection. Finally, Section 5 provides a comprehensive summary of the article.

2. PRELIMINARIES

2.1 Adversarial sample generation algorithms

The key information in a machine learning system includes the architecture of the machine learning model, training data, optimization algorithms and strategies, and the

loss function. Depending on the level of knowledge the attacker possesses about this information, attacks can be categorized into White-Box attacks, Black-Box attacks, and Gray-Box attacks.

White-box attacks In a white-box attack scenario, the attacker possesses complete knowledge about the target machine learning system, including its architecture, gradients, loss function, and other relevant information. This advantage enables the attacker to efficiently generate adversarial samples to subvert the machine learning system. However, for security reasons, white-box attacks are considered improbable in real-world scenarios. Nonetheless, developing effective defense mechanisms against white-box attacks remains an open and challenging research problem [42].

Black-box attacks In a black-box attack scenario, the attacker lacks any access or knowledge about the machine learning system. The attacker can only interact with the machine learning system by inputting data and observing its outputs. Based on these outputs, the attacker can infer the underlying architecture of the machine learning system and construct an auxiliary model to generate adversarial samples for attacking the system. Black-box attacks are prevalent in real-world applications since organizations typically do not expose their machine learning systems to external parties [43].

Gray-box attacks Gray box attacks are alternatively referred to as Semi-White Box attacks. In the context of Gray-Box attacks, the assailant possesses the capability to access machine learning information, yet remains uninformed about the defensive strategies employed by the machine learning system. Gray-Box attacks serve as a middle-ground approach to assess the security of machine learning systems, as they pose significantly greater risks compared to Black-Box attacks [43].

In adversarial sample detection, we mainly consider the following methods of adversarial sample generation: fast gradient sign method (FGSM) [9], basic iterative method (BIM) [23], DeepFool [34] and Carlini-Wagner (C&W) [3]. FGSM is a single-step, fast adversarial sample generation algorithm, formulated as

$$\begin{aligned} \boldsymbol{x}' &= \boldsymbol{x} + \epsilon \operatorname{sign}\big(\nabla_{\boldsymbol{x}} \mathcal{L}(\theta, \boldsymbol{x}, y)\big), \quad \text{non-target}, \\ \boldsymbol{x}' &= \boldsymbol{x} + \epsilon \operatorname{sign}\big(\nabla_{\boldsymbol{x}} \mathcal{L}(\theta, \boldsymbol{x}, y)\big), \quad \text{target on } y. \end{aligned} \tag{1}$$

Target attack (1) is equivalent to

$$\min \mathcal{L}\big(\theta, \boldsymbol{x}', y\big), \quad \text{s.t. } \big\|\boldsymbol{x}' - \boldsymbol{x}\big\|_\infty \le \epsilon \text{ and } \boldsymbol{x}' \in [0,1]^m,$$

where $\mathcal{L}$ and ϵ respectively represent the loss function and magnitude of disturbance. Because FGSM only needs one back propagation process to quickly generate adversarial samples, FGSM is widely used in scenarios that require a large number of adversarial examples such as adversarial training. Figure 1 shows the adversarial samples generated by FGSM based on ImageNet [38]. BIM [23] is an iterative version of FGSM [9]. The iterative process of generating adversarial samples $\boldsymbol{x}'$ is as follows:

$$\boldsymbol{x}_0 = \boldsymbol{x}, \quad \boldsymbol{x}_{t+1} = \operatorname{Clip}_{\boldsymbol{x},\epsilon}\big(\boldsymbol{x}_t + \alpha \operatorname{sign}\big(\nabla_{\boldsymbol{x}} \mathcal{L}(\theta, \boldsymbol{x}_t, y)\big)\big),$$

where Clip represents project sample $\boldsymbol{x}'$ into the $B_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}' : \|\boldsymbol{x}' - \boldsymbol{x}\|_\infty \le \epsilon\}$ sphere of $\boldsymbol{x}$, α denotes the step size, and t represents the number of iterations. When $\boldsymbol{x}$ is initialized randomly, BIM and projected gradient descent (PGM) are equivalent. C&W [3] is to counter the effective defensive strategy for FGSM [9] and L-BFGS [40], and its adversarial samples generation process is

$$\min \big\|\boldsymbol{x} - \boldsymbol{x}'\big\|_2^2 + a f\big(\boldsymbol{x}', y\big), \quad \text{s.t. } \boldsymbol{x}' \in [0,1]^m,$$

where $f(\boldsymbol{x}', y) = (\max_{i \ne y} Z(\boldsymbol{x}')_i - Z(\boldsymbol{x}')_y)^+$, minimizing $f(\boldsymbol{x}', y)$ will result in sample $\boldsymbol{x}'$ being classified into Class y with the highest score. Hyperparameter a is obtained by line search. DeepFool [34] studies the decision hyperplane around the sample $\boldsymbol{x}$ and finds the best path beyond the hyperplane, so that the sample is misclassified. Figure 3 shows an example of DeepFool.

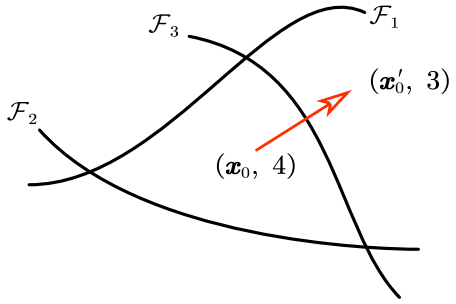


Figure 3. $\mathcal{F}_1$ ($\mathcal{F}_2$ or $\mathcal{F}_3$) is the decision hyperplane of Classes 4 and 1 (2 or 3). DeepFool finds the best path to cross $\mathcal{F}_1$ and misclassifies $\boldsymbol{x}$. (Image Credit: [34].)

The decision boundary of Classes 3 and 4 is $\mathcal{F}_3 = \{\boldsymbol{x} : F(\boldsymbol{x})_4 - F(\boldsymbol{x})_3 = 0\}$. Let $f(\boldsymbol{x}) = F(\boldsymbol{x})_4 - F(\boldsymbol{x})_3 = 0$, and perform Taylor expansion at $\boldsymbol{x}_0$,

$$\mathcal{F}_3' = \big\{\boldsymbol{x} : f(\boldsymbol{x}) \approx f(\boldsymbol{x}_0) + \nabla_{\boldsymbol{x}}^\top f(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0)\big\}.$$

DeepFool calculates the orthogonal vector w from $\boldsymbol{x}_0$ to $\mathcal{F}_3'$ and moves the sample $\boldsymbol{x}$ along the direction w to generate the adversarial sample $\boldsymbol{x}_0'$. DeepFool experiments show that most of the samples are located near the decision boundary. Using MNIST to train LeNet [41], almost 90% of the test samples will be attacked by DeepFool, which shows that the deep learning algorithm is not robust.

2.2 Datasets

Since this paper primarily addresses the attack and defense of adversarial examples in computer vision, this subsection briefly introduces the benchmark datasets employed

Table 1. Basic description of the datasets

Dataset	Training size	Test size	Classes	Figure size
CIFAR-10	50000	10000	10	32×32
CIFAR-100	50000	10000	10	32×32
SVHN	73257	26032	10	32×32
LSUN	\	10000	10	32×32
TinyImageNet	\	10000	200	32×32

for evaluating algorithms. The datasets considered for evaluation are CIFAR-10, CIFAR-100, TinyImageNet, SVHN, and LSUN. Table 1 provides a basic description of these datasets. The paper does not utilize the LSUN and TinyImageNet datasets for training the model. Therefore, there is no specific training set mentioned. As for the test set, it consists of only 10,000 images selected from the original dataset. Furthermore, all the images used in the paper undergo scaling to a size of 32×32.

3. PROPOSED METHODS

Let $\mathcal{X}$ be an input, $\mathcal{Y} = \{1, 2, 3, \ldots, C\}$ be its label set and $\mathcal{D} =: \{((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)) : \boldsymbol{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}, 1 \leq i \leq n\}$. For a pre-trained neural network model as a feature extractor, we use the following softmax classifier

$$P(y = c|\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_c^\top f(\boldsymbol{x}) + b_c)}{\sum_{c' \in \mathcal{Y}} \exp(\boldsymbol{w}_{c'}^\top f(\boldsymbol{x}) + b_{c'})}, \quad \boldsymbol{x} \in \mathcal{X}, y \in \mathcal{Y}$$

to validate the accuracy of the pre-trained neural network features, where $\boldsymbol{w}_c^\top$ and b_c represent the weight vector and bias of Class c, respectively, and $f(\boldsymbol{x})$ is the output of the penultimate layer before softmax.

It is worth noting that, under the Gaussian assumption, the conditional distribution of $f(\boldsymbol{x})$ given $y = c$ follows a Gaussian distribution with a mean $\boldsymbol{\mu}_c$ and a common covariance matrix $\boldsymbol{\Sigma}$, denoted as $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma})$. The prior probability of label y is defined as $P(y = c) = \frac{b_c}{\sum_{c' \in \mathcal{Y}} b_{c'}}$, where b_c represents the weight for label c and $\mathcal{Y}$ is the set of all possible labels. Based on these assumptions, the posterior probability of $y = c$ given $\boldsymbol{x}$ can be expressed as:

$$\begin{aligned} P(y = c|\boldsymbol{x}) &= \frac{P(y = c)\phi(f(\boldsymbol{x})|y = c)}{\sum_{c' \in \mathcal{Y}} P(y = c')\phi(f(\boldsymbol{x})|y = c')} \\ &= \frac{b_c \exp(-\frac{1}{2}(f(\boldsymbol{x}) - \boldsymbol{\mu}_c)^\top \boldsymbol{\Sigma}^{-1}(f(\boldsymbol{x}) - \boldsymbol{\mu}_c))}{\sum_{c' \in \mathcal{Y}} b_{c'} \exp(-\frac{1}{2}(f(\boldsymbol{x}) - \boldsymbol{\mu}_{c'})^\top \boldsymbol{\Sigma}^{-1}(f(\boldsymbol{x}) - \boldsymbol{\mu}_{c'}))} \\ &= \frac{\exp(\boldsymbol{\mu}_c^\top \boldsymbol{\Sigma}^{-1} f(\boldsymbol{x}) - \frac{1}{2}\boldsymbol{\mu}_c^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log b_c)}{\sum_{c' \in \mathcal{Y}} \exp(\boldsymbol{\mu}_{c'}^\top \boldsymbol{\Sigma}^{-1} f(\boldsymbol{x}) - \frac{1}{2}\boldsymbol{\mu}_{c'}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{c'} + \log b_{c'})}, \end{aligned}$$

where $\phi(f(\boldsymbol{x})|y = c')$ denotes the probability density function (PDF) of $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma})$. Then, Gaussian discriminant analysis (GDA) can be considered equivalent to a softmax classifier. Moreover, the classification procedure described above can be implemented using the Mahalanobis distance,

$$y(\boldsymbol{x}) = \arg\min_c \big(f(\boldsymbol{x}) - \boldsymbol{\mu}_c\big)^\top \boldsymbol{\Sigma}^{-1} \big(f(\boldsymbol{x}) - \boldsymbol{\mu}_c\big),$$

which emphasizes the rationality and generality of classification based on the Mahalanobis distance. Since the parameters $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}$ are unknown, they are typically estimated using the maximum likelihood estimation. To estimate these parameters, the model can be trained using pre-trained features, such as

$$\begin{aligned} \widehat{\boldsymbol{\mu}}_c &= \frac{1}{n_c} \sum_{i:y_i=c} f(\boldsymbol{x}_i), \\ \widehat{\boldsymbol{\Sigma}} &= \frac{1}{n} \sum_{c \in \mathcal{Y}} \sum_{i:y_i=c} \big(f(\boldsymbol{x}_i) - \widehat{\boldsymbol{\mu}}_c\big)\big(f(\boldsymbol{x}_i) - \widehat{\boldsymbol{\mu}}_c\big)^\top, \end{aligned}$$

where $n_c = \#\{i : y_i = c, 1 \leq i \leq n\}$ and $\#\mathcal{A}$ denotes the cardinality of the set $\mathcal{A}$. In [26], the confidence score based on estimated Mahalanobis distance is defined as

$$\begin{aligned} \widehat{K}(\boldsymbol{x}) &= \max_c -\big(f(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_c\big)^\top \widehat{\boldsymbol{\Sigma}}^{-1} \big(f(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_c\big) \\ &= \min_c \big(f(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_c\big)^\top \widehat{\boldsymbol{\Sigma}}^{-1} \big(f(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_c\big). \end{aligned} \tag{2}$$

In their work, Lee et al. [26] demonstrate the effectiveness of abnormal sample detection using the Mahalanobis distance through experiments. Denote Mahalanobis distance (MD) of the ith observation $(\boldsymbol{x}_i, y_i = c)$ by

$$\mathrm{MD}_i = \big(f(\boldsymbol{x}_i) - \widehat{\boldsymbol{\mu}}_c\big)^\top \widehat{\boldsymbol{\Sigma}}^{-1} \big(f(\boldsymbol{x}_i) - \widehat{\boldsymbol{\mu}}_c\big). \tag{3}$$

Since (3) is a quadratic form, the lower 0.975th quantile of Chi-square distribution with p degrees of freedom $\chi^2_{p,0.975}$ is usually selected as the threshold to judge an outlier, i.e.

$$\begin{cases} \boldsymbol{x}_i \text{ is an outlier,} & \mathrm{MD}_i \geq \chi^2_{p,0.975}, \\ \boldsymbol{x}_i \text{ is a in-distribution,} & \mathrm{MD}_i < \chi^2_{p,0.975}. \end{cases}$$

There are two problems for abnormal detection based on Mahalanobis distance:

1. *Masking problem*: when there are outliers involved in the estimation of the mean and the covariance, for other outliers, there may not be a Mahalanobis distance larger than the in-distribution sample;
2. *Swamping problem*: because both mean and covariance matrix are involved in Mahalanobis distance, the observations with large Mahalanobis distance may be not outliers.

Therefore, when utilizing the Mahalanobis distance for discrimination, it becomes crucial to consider the robust estimation of the location and scale parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. This paper recommends two robust estimators for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, which are introduced in the following two subsections.

3.1 The MCD estimator

There exist several methods for robust estimation of the Mahalanobis distance in the literature. One such method is the MCD estimator proposed by Rousseeuw [20]. The MCD approach aims to select a sub-sample with a sample size of h, where $n/2 \leq h < n$, that minimizes the determinant of the covariance matrix. However, due to its high computational complexity and limitations in computing power at the time, the MCD method was not widely utilized. It was not until the introduction of the fast-MCD algorithm by Rousseeuw and Driessen [37] that the computational challenges associated with the MCD method were addressed. The fast-MCD algorithm provides an efficient solution to the MCD estimation problem. One of the key steps in the fast-MCD algorithm is the C-step. To explain the MCD estimators for the location and scale parameters of the dataset $\mathbf{Z}_n = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\}^\top$ from a population $\mathbf{Z}$ of dimension p, we consider the penultimate layer $\mathbf{z}$ at iteration t as an example. Here, $H_t \subset \{1, 2, \ldots, n\}$ represents the selected subset, and $h_t = \#H_t$ denotes the number of elements in H_t. Using maximum likelihood estimation

$$
\begin{aligned}
\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}^{(t)} &=: \frac{1}{h_t} \sum_{i \in H_t} \boldsymbol{z}_i, \\
\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}^{(t)} &=: \frac{1}{h_t} \sum_{i \in H_t} \big(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}^{(t)}\big)\big(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}^{(t)}\big)^\top,
\end{aligned} \tag{4}
$$

if $|\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}^{(t)}| \neq 0$, for $i \in H_t$, define the relative distance as follows

$$
d_t(i) = \sqrt{\big(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}^{(t)}\big)^\top \widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}^{(t)-1} \big(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}^{(t)}\big)},
$$

where $|\boldsymbol{\Sigma}|$ denote the determinant of $\boldsymbol{\Sigma}$. Then, take the following set H_{t+1} to satisfy

$$
\big\{d_t(i) : i \in H_{t+1}\big\} =: \big\{(d_t)_{1:n}, \ldots, (d_t)_{h:n}\big\},
$$

where $(d_t)_{1:n} \leq (d_t)_{2:n} \leq \cdots \leq (d_t)_{n:n}$. Next calculate $\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}^{(t+1)}$ and $\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}^{(t+1)}$ based on H_{t+1}, satisfying

$$
\big|\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}^{(t)}\big| \geq \big|\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}^{(t+1)}\big|. \tag{5}
$$

Rousseeuw and Driessen [37] demonstrates that the equation holds in (5), if and only if $\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}^{(t)} = \widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}^{(t+1)}, \widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}^{(t)} = \widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}^{(t+1)}$.

The MCD estimator is known for its *affine equivariance* [18]. This property states that for any non-singular matrix $\boldsymbol{A}$ and constant vector $\boldsymbol{b}$, the following relationships hold true:

$$
\begin{aligned}
\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}(\boldsymbol{AZ} + \boldsymbol{b}) &= \boldsymbol{A}\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}(\boldsymbol{Z}) + \boldsymbol{b}, \\
\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}(\boldsymbol{AZ} + \boldsymbol{b}) &= \boldsymbol{A}\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}(\boldsymbol{Z})\boldsymbol{A}^\top.
\end{aligned}
$$

This property implies that rotations, linear transformations, and scaling of the data will not affect the detection of outliers when using the MCD estimator [18]. In other words, the MCD estimator is robust to variations in the data's location, orientation, and scale.

The breakdown point [17] and influence function [5] are important indicators to evaluate the robustness of an estimator. Croux and Haesbroeck [5] provides the influence functions for the MCD location and scatter matrix estimators at elliptically symmetric distributions F with stochastic representation

$$
\boldsymbol{Z} = \boldsymbol{\mu} + r\boldsymbol{\Sigma}^{1/2}\boldsymbol{U} \tag{6}
$$

which has the density of the form

$$
|\boldsymbol{\Sigma}|^{-1/2} g\big((\boldsymbol{z} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{z} - \boldsymbol{\mu})\big), \tag{7}
$$

where r denotes a nonnegative random variable independent of $\boldsymbol{U}$ that is a p-dimensional random vector uniformly distributed on the unit hypersphere, and g is called density generator.

Theorem 3.1 ([5]). *Assume that g in (7) has a strictly negative derivative g'. Let $0 < \alpha < 1$ be the mass of the data not determining the MCD, and $q_\alpha = G^{-1}(1-\alpha)$, where G is the CDF of $\boldsymbol{z}^\top\boldsymbol{z}$ and $\boldsymbol{z}$ follows distribution F with $\boldsymbol{\mu} = \boldsymbol{0}$ and $\boldsymbol{\Sigma} = \boldsymbol{I}_p$. Then, the influence function of $\widehat{\boldsymbol{\mu}}_{MCD}$ is*

$$
I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\mu}}_{MCD}, F) = -\frac{1}{2} c_1 \boldsymbol{x} I\big(\|\boldsymbol{x}\|^2 \leqslant q_\alpha\big),
$$

where $c_1 = \big(\frac{\pi^{p/2}}{\Gamma(p/2+1)} \int_0^{\sqrt{q_\alpha}} r^{p+1} g'(r^2) dr\big)^{-1}$.

The influence function of the MCD estimator of the scatter matrix is separated into expressions for diagonal and off-diagonal elements.

$$
\begin{aligned}
&I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\Sigma}}_{ii,MCD}, F) \\
&= \frac{1}{b_1} \left\{ c_2 x_i^2 I\big(\|\boldsymbol{x}\|^2 \leqslant q_\alpha\big) + \frac{b_2 c_2}{b_1 - p b_2} \|\boldsymbol{x}\|^2 I\big(\|\boldsymbol{x}\|^2 \leqslant q_\alpha\big) \right. \\
&\quad \left. + \frac{b_1}{b_1 - p b_2} \left(c_2 \frac{q_\alpha}{p} \big(1 - \alpha - I\big(\|\boldsymbol{x}\|^2 \leqslant q_\alpha\big)\big) - 1 \right) \right\},
\end{aligned}
$$

$$
I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\Sigma}}_{ij,MCD}, F) = \frac{x_i x_j}{-2c_3} I\big(\|\boldsymbol{x}\|^2 \leqslant q_\alpha\big), \quad \text{if } i \neq j,
$$

where $c_2 = \big(\frac{\pi^{p/2}}{\Gamma(p/2+1)} \int_0^{\sqrt{q_\alpha}} r^{p+1} g(r^2) dr\big)^{-1}$, and the constants b_1, b_2 and c_3 are determined by the relations

$$
\begin{aligned}
b_1 &= -2c_2c_3, \quad b_2 = \frac{1}{2} + c_2 \left(c_3 - \frac{q_\alpha}{p} \left(\frac{1}{c_1} + \frac{1-\alpha}{2} \right) \right), \\
c_3 &= \frac{\pi^{p/2}}{(p+2)\Gamma(p/2+1)} \int_0^{\sqrt{q_\alpha}} r^{p+3} g'\big(r^2\big) \mathrm{d}r.
\end{aligned}
$$

Moreover, MCD is a recognized high breakdown point estimator [18]. The breakdown point is the smallest proportion of observations replaced with arbitrary values when the estimator is invalid. Let $\boldsymbol{Z}_m$ denote the data set after replacing any m observations in $\boldsymbol{Z}_n$. Then, the breakdown point

of $\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}$ and $\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}$ are

$$\epsilon_n^*(\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}};\boldsymbol{Z}_n)=\frac{1}{n}\min\Big\{m\in\{1,2,\ldots,n\}:\sup_m\big\|\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}(\boldsymbol{Z}_n)-\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}}(\boldsymbol{Z}_m)\big\|=+\infty\Big\},$$

$$\epsilon_n^*(\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}};\boldsymbol{Z}_n)=\tfrac{1}{n}\min\Big\{m\in\{1,2,\ldots,n\}:\sup_m\max_i\big\{\big|\log\big(\lambda_i(\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}(\boldsymbol{Z}_n)\big)\big)-\log\big(\lambda_i(\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}(\boldsymbol{Z}_m)\big)\big)\big|\big\}\Big\},$$

respectively, where $\lambda_1(\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}})\geq\cdots\geq\lambda_p(\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}})>0$ are all eigenvalues of $\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}}$. This means that when any eigenvalue of the scatter estimator tends to 0 or infinity, the MCD estimation will collapse. Let $k(\boldsymbol{Z}_n)$ denote the maximum number of observations in $\boldsymbol{Z}_n$ lying on the hyperplane. Assume that $k(\boldsymbol{Z}_n)<h$, then for the MCD estimator of location and scatter, Rousseeuw and Driessen [37] shows that

$$\epsilon_n^*(\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}};\boldsymbol{Z}_n)=\epsilon_n^*(\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}};\boldsymbol{Z}_n)=\frac{\min\{n-h+1,h-k\}}{n}.$$

If $\boldsymbol{Z}$ is a continuous random vector, then

$$P\big(k(\boldsymbol{Z}_n)=p\big)=1,\quad a.s.,$$

and

$$\epsilon_n^*(\widehat{\boldsymbol{\mu}}_{\mathrm{MCD}};\boldsymbol{Z}_n)=\epsilon_n^*(\widehat{\boldsymbol{\Sigma}}_{\mathrm{MCD}};\boldsymbol{Z}_n)=\frac{\min\{n-h+1,h-p\}}{n}.$$

It implies that for any $[(n+p)/2]\leq h\leq[(n+p+1)/2]$, we can have the highest breakdown point $[n-p+2]/(2n)$, where $[n]$ denotes the integer part of n. For the existence, consistency, and weak convergence of MCD estimates, one can refer to [2].

3.2 The T-type estimator

In many cases, when data is obtained, it is commonly assumed that the data or the measurement errors associated with the data follow a normal distribution. This assumption forms the basis for modeling and parameter estimation. However, this assumption of normal distribution may not hold in many scenarios, especially when dealing with heavy-tailed noise or outliers in the data. Therefore, it is crucial to make reasonable assumptions about the distribution of the data, taking into account the presence of noise and outliers. In the context of image data, noise and outliers are often inevitable due to various factors such as sensor limitations, environmental conditions, or image acquisition processes. These factors can introduce additional variability and deviations from the idealized assumptions of normality. Therefore, it becomes essential to consider alternative distributions or robust estimation techniques that can better capture the characteristics of the data and handle the presence of noise and outliers effectively.

In this subsection, we consider the case where $\boldsymbol{z}\sim t_p(\boldsymbol{\mu},\boldsymbol{\Sigma},\nu)$, where p, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, and ν represent the dimension, location, scatter, and degree of freedom parameters, respectively. The probability density function of the t-distribution, for a given $\nu>0$, is given by:

$$\varphi_\nu(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{|\boldsymbol{\Sigma}|^{-1/2}\Gamma((\nu+p)/2)}{\Gamma^p(1/2)\Gamma(\nu/2)\nu^{p/2}}\left(1+\frac{(\boldsymbol{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})}{\nu}\right)^{-\frac{\nu+p}{2}}.\tag{8}$$

Let $\rho_\nu(\boldsymbol{x})=(\nu+p)\log(1+\frac{\boldsymbol{x}^\top\boldsymbol{x}}{\nu})\propto-2\log(\varphi_\nu(\boldsymbol{x}|\boldsymbol{0},\mathbf{I}_p))$. If $\boldsymbol{Z}_n$ is obtained, we get the maximum likelihood estimate (MLE) of $(\boldsymbol{\mu},\boldsymbol{\Sigma})$ as

$$\begin{aligned}(\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{\Sigma}})&=\arg\max_{(\boldsymbol{\mu},\boldsymbol{\Sigma}>0)}\sum_{i=1}^n\log\varphi_\nu(\boldsymbol{z}_i|\boldsymbol{\mu},\boldsymbol{\Sigma})\\&=\arg\min_{(\boldsymbol{\mu},\boldsymbol{\Sigma}>0)}\sum_{i=1}^n\big(\rho_\nu\big(\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{z}_i-\boldsymbol{\mu})\big)+\log|\boldsymbol{\Sigma}|\big)\\&=:\arg\min_{(\boldsymbol{\mu},\boldsymbol{\Sigma}>0)}\sum_{i=1}^n l(\boldsymbol{\mu},\boldsymbol{\Sigma};\boldsymbol{z}_i),\end{aligned}\tag{9}$$

where $l(\boldsymbol{\mu},\boldsymbol{\Sigma};\boldsymbol{z})$ is viewed as a loss function. It is obvious to see

$$t_p(\boldsymbol{\mu},\boldsymbol{\Sigma})\overset{\mathrm{d}}{=}\boldsymbol{\mu}+\frac{\boldsymbol{\Sigma}^{1/2}\boldsymbol{Y}}{\sqrt{\kappa}},$$

where $\boldsymbol{Y}\sim\mathcal{N}_p(\boldsymbol{0},\boldsymbol{I}_p),\nu\kappa\sim\chi^2_\nu$. It yields that

$$\boldsymbol{Z}|(\boldsymbol{\mu},\boldsymbol{\Sigma},\kappa)\sim\mathcal{N}_p(\boldsymbol{\mu},\boldsymbol{\Sigma}/\kappa),\tag{10}$$

and we treat κ as weight of $\boldsymbol{Z}$. When $\nu>1$, $\boldsymbol{\mu}$ is the mean of $\boldsymbol{Z}$, and if $\nu>2$, $\nu/(\nu-2)\boldsymbol{\Sigma}$ is the covariance matrix. As $\nu\to\infty$, $\boldsymbol{Z}\sim\mathcal{N}_p(\boldsymbol{\mu},\boldsymbol{\Sigma})$. Therefore, the t distribution family provides a heavy-tailed alternative for the normal family. Let $\mathrm{MD}_{\boldsymbol{z}}(\boldsymbol{\mu},\boldsymbol{\Sigma})=(\boldsymbol{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})$ be the Mahalanobis distance between $\boldsymbol{z}$ and the center $\boldsymbol{\mu}$ with respect to $\boldsymbol{\Sigma}$. Because the Gamma distribution is a conjugate prior distribution, according to (10), the conditional posterior distribution of κ given $(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\Sigma})$ follows

$$\kappa|(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\Sigma})\propto\mathrm{Gamma}\left(\frac{\nu+p}{2},\frac{\nu+\mathrm{MD}_{\boldsymbol{z}}(\boldsymbol{\mu},\boldsymbol{\Sigma})}{2}\right),$$

from which we have

$$E(\kappa|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{\nu+p}{\nu+\mathrm{MD}_{\boldsymbol{z}}(\boldsymbol{\mu},\boldsymbol{\Sigma})}.\tag{11}$$

By (10), we can get the following likelihood function of $\boldsymbol{Z}_n$ and $\boldsymbol{\kappa}_n=\{\kappa_1,\kappa_2,\ldots,\kappa_n\}$

$$\begin{aligned}&\log L_N(\boldsymbol{\mu},\boldsymbol{\Sigma}|\boldsymbol{Z}_n,\boldsymbol{\kappa}_n)\\&\quad=-\frac{n}{2}\log|\boldsymbol{\Sigma}|-\frac{1}{2}\sum_{i=1}^n\kappa_i(\boldsymbol{z}_i-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_i-\boldsymbol{\mu}).\end{aligned}\tag{12}$$

In statistical practice, the EM algorithm is used for the MLE of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with missing $\boldsymbol{\kappa}_n$. Let $\Theta^{(t)} = \{\boldsymbol{Z}_n, \widehat{\boldsymbol{\mu}}_T^{(t)}, \widehat{\boldsymbol{\Sigma}}_T^{(t)}\}$. Initialize $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\Sigma}^{(0)}$ to zero vector and identity matrix, respectively. From (11) and (12), at iteration $t+1$ with input $\Theta^{(t)}$,

E-step: Calculate

(13)
$$w_i^{(t+1)} = E\big(\kappa_i|\Theta^{(t)}\big) = \frac{\nu+p}{\nu + \mathrm{MD}_{\boldsymbol{z}_i}(\widehat{\boldsymbol{\mu}}_T^{(t)}, \widehat{\boldsymbol{\Sigma}}_T^{(t)})}, \ i = 1, \ldots, n;$$

M-step: Calculate

(14)
$$\begin{cases} \widehat{\boldsymbol{\mu}}_T^{(t+1)} = \dfrac{\sum_{i=1}^n w_i^{(t+1)} \boldsymbol{z}_i}{\sum_{i=1}^n w_i^{(t+1)}}, \\ \widehat{\boldsymbol{\Sigma}}_T^{(t+1)} = \dfrac{1}{n}\displaystyle\sum_{i=1}^n w_i^{(t+1)} (\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_T^{(t+1)})(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_T^{(t+1)})^\top. \end{cases}$$

Next, if $\boldsymbol{Z}$ follows a general distribution $F(\cdot)$ with center $\boldsymbol{\mu}$ and scatter matrix $\boldsymbol{\Sigma}$, we still minimize the specific loss function on the right side of (8), and define the T-type estimate of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as

$$(\widehat{\boldsymbol{\mu}}_T, \widehat{\boldsymbol{\Sigma}}_T) = \underset{(\boldsymbol{\mu}, \boldsymbol{\Sigma}>0)}{\arg\min} \sum_{i=1}^n \big(\rho_\nu\big(\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{z}_i - \boldsymbol{\mu})\big) + \log|\boldsymbol{\Sigma}|\big).$$

It is worthy to note that the EM algorithm (12) and (13) are also available for the calculation of T-type estimator $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Usually, we recommend the degree of freedom ν taken as $3 \sim 5$ due to a tradeoff between robustness and efficiency of estimate $(\widehat{\boldsymbol{\mu}}_T, \widehat{\boldsymbol{\Sigma}}_T)$.

Analogous to MCD estimator, the T-type estimator is also affine equivariant [10].

Finally, we provide two robustness indicators: the influence function and the breakdown bound, for a multivariate T-type estimator. The proof of the following theorem is provided in the appendix.

Theorem 3.2. *Let $\boldsymbol{Z}$ be a random vector with location and scale parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Given $\nu > 0$, the influence function of $\widehat{\boldsymbol{\mu}}_T$ of $\boldsymbol{Z}$ at F is*

$$\begin{aligned} &I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\mu}}_T, F) \\ &= \left\{ E_F \frac{[\nu + (\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})]\boldsymbol{\Sigma}^{-1} - 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})(\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}}{[\nu + (\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})]^2} \right\}^{-1} \\ &\quad \cdot \frac{\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{\nu + (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \end{aligned}$$

and the influence function of $\widehat{\boldsymbol{\Sigma}}_T$ denoted by $I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\Sigma}}_T, F)$ satisfies the following equation

(15)
$$\begin{aligned} &I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\Sigma}}_T, F) + (\nu+p) E_F \\ &\quad \times \left(\frac{(\boldsymbol{Z}-\boldsymbol{\mu})(\boldsymbol{Z}-\boldsymbol{\mu})^\top \mathrm{tr}\{\boldsymbol{\Sigma}(F)^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})(\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(F)^{-1} I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\Sigma}}_T, F)\}}{[\nu + (\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(F)^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})]^2} \right) \\ &= \frac{(\nu+p)(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top}{\nu + (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(F)^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \\ &\quad - (\nu+p) E_F \left(\frac{(\boldsymbol{Z}-\boldsymbol{\mu})(\boldsymbol{Z}-\boldsymbol{\mu})^\top}{\nu + (\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(F)^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})} \right). \end{aligned}$$

Denote $\boldsymbol{\Sigma}(F)^{-1/2} I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\Sigma}}_T, F) \boldsymbol{\Sigma}(F)^{-1/2} = (h_{i,j}(\boldsymbol{x}))_{p\times p}$. If F is further assumed to be an elliptically symmetric distribution with stochastic representation (6), for $i \neq j$, the diagonal and off-diagonal entries of $\boldsymbol{\Sigma}(F)^{-1/2} I\!F(\boldsymbol{x}, \widehat{\boldsymbol{\Sigma}}_T, F) \boldsymbol{\Sigma}(F)^{-1/2}$ are

$$\begin{aligned} h_{i,i}(\boldsymbol{x}) &= \left(1 + \frac{3(\nu+p)}{p(p+2)} E \frac{r^2}{(\nu+r^2)^2}\right)^{-1} \\ &\quad \cdot \left(M_{i,i}(\boldsymbol{x}) - \frac{\nu+p}{p(p+2)} E \frac{r^2}{(\nu+r^2)^2} \sum_{j\neq i} h_{j,j}(\boldsymbol{x}) \right), \\ h_{i,j}(\boldsymbol{x}) &= \left(1 + \frac{2(\nu+p)}{p(p+2)} E \frac{r^2}{(\nu+r^2)^2}\right)^{-1} M_{i,j}(\boldsymbol{x}), \end{aligned}$$

where $(M_{i,j}(\boldsymbol{x}))_{p\times p} = \frac{(\nu+p)\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1/2}}{\nu + (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(F)^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} - \boldsymbol{I}_p$.

Figures 4 and 5 plot the influence functions of T-type location and scale estimators at bivariate $t(1)$ distribution with mean zero and covariance matrix $(0.5^{|i-j|})_{2\times 2}$. It shows the functions are bounded and smooth.

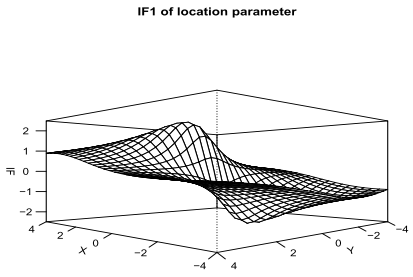


Figure 4. Influence function of the first element of T-type location estimator at bivariate $t(1)$ distribution with mean zero and covariance matrix $(0.5^{|i-j|})_{2\times 2}$.

Denote $F_\epsilon = (1-\epsilon)F + \epsilon\Delta_{\boldsymbol{x}}$, where $\Delta_{\boldsymbol{x}}$ is a Dirac measure putting all its mass on $\boldsymbol{x}$. Operating with distributions of the form F_ϵ, to calculate the breakdown bound ϵ^* of $(\widehat{\boldsymbol{\mu}}_T, \widehat{\boldsymbol{\Sigma}}_T)$, letting $\boldsymbol{x} \to \infty$, Maronna [32] obtains

$$\epsilon^*\big((\widehat{\boldsymbol{\mu}}_T, \widehat{\boldsymbol{\Sigma}}_T); \boldsymbol{Z}_n\big) \leq \min\left\{ \frac{1}{\nu+p}, \frac{\nu}{\nu+p} \right\}.$$

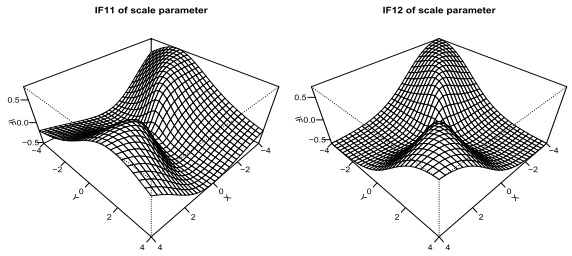


Figure 5. Influence function of T-type scale estimator at bivariate $t(1)$ distribution with mean zero and covariance matrix $(0.5^{|i-j|})_{2\times 2}$, IF11 the first diagonal entry of the scatter matrix, IF12 for the off-diagonal entry of the scatter matrix.

3.3 A weight remark

Combining (4) and (14), we provide the following generic expression of the robust estimators of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$

$$\begin{aligned}\widehat{\boldsymbol{\mu}} &= \sum_{i=1}^{n} \varpi_{1i}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n)\boldsymbol{z}_i,\\ \widehat{\boldsymbol{\Sigma}} &= \sum_{i=1}^{n} \varpi_{2i}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n)(\boldsymbol{z}_i-\widehat{\boldsymbol{\mu}})(\boldsymbol{z}_i-\widehat{\boldsymbol{\mu}})^\top,\end{aligned} \tag{16}$$

where $\varpi_i(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n)$ is a bounded weight function based on the distance between $\boldsymbol{z}_i$ and $\boldsymbol{\mu}$ and makes the corresponding estimator affine equivariant. In particular, when $\varpi_{1i}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n) = \varpi_{2i}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n) = I_{H_{t+1}}(i)/\sum_{j=1}^{n} I_{H_{t+1}}(j)$, (16) is the MCD estimators at $(t+1)$ iteration, where $I_{H_{t+1}}(\cdot)$ is the indicator function of a subset H_{t+1} of $\{1,\ldots,n\}$; When $\varpi_{1i}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n) = \omega_i^{(t+1)}/\sum_{i=1}^{n}\omega_i^{(t+1)}$ and $\varpi_{2i}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n) = \omega_i^{(t+1)}/n$, (16) represents the $(t+1)$-step of T-type estimators, where $\omega_i^{(t+1)}$ is defined as (13); When $\varpi_{1i}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n) = \varpi_{2i}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n) = 1/n$, (16) is the sample mean and sample covariance matrix.

Recall (2) and replace $(\widehat{\boldsymbol{\mu}}_c, \widehat{\boldsymbol{\Sigma}})$ with the robust estimators of the location parameters in each class and the common scatter matrix. Then,

$$\widehat{K}_{\mathrm{R}}(\boldsymbol{x}) = \min_c \big(f(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_{c,\mathrm{R}}\big)^\top \widehat{\boldsymbol{\Sigma}}_{\mathrm{R}}^{-1}\big(f(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_{c,\mathrm{R}}\big), \tag{17}$$

where the subscript R represents MCD or T-type estimators, or other robust estimation methods.

In order to enhance the model's performance and utilize the low-level features of the deep neural network, the features from each level of the network are integrated. For a given sample $\mathbf{x}$, the features of the ℓth hidden layer are denoted as $f_\ell(\boldsymbol{x})$. The mean $\{\widehat{\boldsymbol{\mu}}_{\ell,1},\ldots,\widehat{\boldsymbol{\mu}}_{\ell,C}\}$ and covariance $\boldsymbol{\Sigma}_\ell$ of this layer are calculated. To measure the confidence score of $\boldsymbol{x}$ in layer ℓ, we employ (17). Consequently, the weighted confidence score of $\boldsymbol{x}$ across each layer of the neural network is defined as:

$$A_{\mathrm{R}}(\boldsymbol{x}) = \sum_{\ell} \alpha_\ell K_{\ell,\mathrm{R}}(\boldsymbol{x}), \tag{18}$$

where $A_{\mathrm{R}}(\boldsymbol{x})$ serves as the measure for anomaly detection, and the threshold is determined through cross-validation. Here, α_ℓ denotes the weight assigned to the ℓth layer, which is obtained by training a logistic model.

3.4 Class incremental learning

The confidence score based on Mahalanobis distance can be naturally extended to class incremental learning [35]. This approach allows a pre-training model to gradually incorporate new class samples without the need to retrain the neural network model. This capability is crucial in real-world applications, as deep neural networks are commonly used for processing large volumes of data. Retraining the model every time a new class of data is obtained can be costly, and there is no guarantee that the new class data does not contain abnormal samples. Therefore, ensuring robust estimation of the Mahalanobis distance is particularly important when using the confidence score based on it for discrimination. In Algorithm 1, we present a general framework for class incremental learning based on the Mahalanobis distance confidence score. This framework is applicable to the two robust estimation methods discussed in the paper: MCD and T-type estimators. It can also be extended to other robust estimation methods.

Algorithm 1 Extension of confidence score based on RMD to class incremental learning

Require: Dataset from new class $\boldsymbol{x}_i$, $i = 1,2,\ldots,n_{C+1}$; mean and covariance of observed classes $(\widehat{\boldsymbol{\mu}}_c, \widehat{\boldsymbol{\Sigma}})$, $c = 1,\ldots,C$

Ensure: Mean and covariance of all classes $(\widehat{\boldsymbol{\mu}}_c, \widehat{\boldsymbol{\Sigma}})$, $c = 1,\ldots,C,C+1$

1: Based on MCD or T-type estimators and using $\boldsymbol{x}_i$, $i = 1,2,\ldots,n_{C+1}$, calculate $(\widehat{\boldsymbol{\mu}}_{C+1}, \widehat{\boldsymbol{\Sigma}}_{C+1})$

2: Update the covariance of all classes $\widehat{\boldsymbol{\Sigma}} \longleftarrow \frac{C}{C+1}\widehat{\boldsymbol{\Sigma}} + \frac{1}{C}\widehat{\boldsymbol{\Sigma}}_{C+1}$

4. REAL DATA ANALYSIS

In this section, we evaluate the performance of the proposed method by conducting experiments. We utilize the DenseNet and ResNet architectures as the CNN models. Furthermore, we select the following image datasets for our experiments: SVHN, TinyImageNet, LSUN, CIFAR-10, and CIFAR-100.

4.1 Outlier detection

Setup In our outlier detection experiments, we trained DenseNet with 100 layers and ResNet with 34 layers, following the architectures proposed in Huang, Liu and Weinberger [16] and He et al. [11] respectively. The training data

Table 2. Outlier detection (all numbers are percentages; the best results are bolded)

In-dist (model)	OOD	Validation on OOD samples TNR at TPR 95% AUROC Detection acc. Baseline method/ODIN/Mahalanobis/S/MCD/T-type estimator		
CIFAR-10 (DenseNet)	SVHN	40.2/86.2/90.8/88.5/86.5/**99.1**	89.9/95.5/**98.1**/96.7/95.9/97.7	83.2/91.4/93.9/92.3/91.7/**96.3**
	TinyImageNet	58.9/92.4/95.0/91.1/91.2/**98.3**	94.1/98.5/**98.8**/97.3/97.1/98.1	88.5/93.9/95.0/93.2/93.2/**97.7**
	LSUN	66.6/96.2/97.2/96.7/95.4/**97.5**	95.4/99.2/**99.3**/98.8/98.2/99.2	90.3/95.7/96.3/96.0/95.4/**99.5**
CIFAR-100 (DenseNet)	SVHN	26.7/70.6/82.5/79.9/80.5/**99.7**	82.7/93.8/**97.2**/93.8/94.0/93.7	75.6/86.6/91.5/88.5/88.9/**98.8**
	TinyImageNet	17.6/42.6/86.6/81.6/76.1/**90.7**	71.7/85.2/**97.4**/96.1/90.6/91.8	65.7/77.0/92.2/90.6/86.7/**97.5**
	LSUN	16.7/41.2/91.4/84.0/78.9/**93.2**	70.8/85.5/**98.0**/95.5/91.9/94.3	64.9/77.1/93.9/90.6/87.6/**99.6**
SVHN (DenseNet)	CIFAR-10	69.3/71.7/**96.8**/95.4/79.3/96.6	91.9/91.4/**98.9**/98.6/91.4/98.7	86.6/85.8/**95.9**/95.4/88.1/**95.9**
	TinyImageNet	79.8/84.1/**99.9**/99.7/99.2/**99.9**	94.8/95.1/**99.9**/99.7/99.3/99.8	90.2/90.4/**98.9**/98.4/97.8/98.6
	LSUN	77.1/81.1/**100**/99.9/99.5/99.9	94.1/94.5/**99.9**/99.7/99.4/99.8	89.1/89.2/**99.3**/98.6/98.1/99.1
CIFAR-10 (ResNet)	SVHN	32.5/86.6/**96.4**/95.3/95.4/96.2	89.9/96.7/**99.1**/99.0/**99.1**/**99.1**	85.1/91.1/**95.8**/95.3/95.3/95.6
	TinyImageNet	44.7/72.5/**97.1**/96.3/95.2/**97.1**	91.0/94.0/**99.5**/99.2/99.0/99.4	85.1/86.5/96.3/96.2/95.2/**96.4**
	LSUN	45.4/73.8/**98.9**/98.6/97.5/**98.9**	91.0/94.1/**99.7**/99.5/99.1/**99.7**	85.3/86.7/**97.7**/97.5/96.8/**97.7**
CIFAR-100 (ResNet)	SVHN	20.3/62.7/**91.9**/91.4/**91.9**/90.2	79.5/93.9/**98.4**/98.0/98.3/98.2	73.2/88.0/93.7/93.5/**93.6**/93.3
	TinyImageNet	20.4/49.2/**90.9**/85.7/88.3/88.5	77.2/87.6/**98.2**/97.0/97.6/97.7	70.8/80.1/**93.3**/91.3/92.2/92.3
	LSUN	18.8/45.6/**90.9**/84.6/88.7/89.5	75.8/85.6/**98.2**/97.0/97.8/97.9	69.9/78.3/**93.5**/91.3/92.8/93.0
SVHN (ResNet)	CIFAR-10	78.3/79.8/98.4/98.2/98.1/**98.5**	92.9/92.1/**99.3**/**99.3**/99.0/99.2	90.0/89.4/96.9/96.8/96.6/**97.0**
	TinyImageNet	79.0/82.1/**99.9**/**99.9**/**99.9**/**99.9**	93.5/92.0/**99.9**/**99.9**/99.8/99.8	90.4/89.4/**99.1**/98.9/98.9/98.8
	LSUN	74.3/77.3/99.9/99.8/**100.0**/**100.0**	91.6/89.4/**99.9**/99.8/**99.9**/**99.9**	89.0/87.2/**99.5**/99.1/99.3/99.3

for the CNN consisted of CIFAR-10, CIFAR-100, and SVHN datasets, which were considered as in-distribution samples (positive). Conversely, the outlier data (negative) were not included in the training process. The SVHN, TinyImageNet, and LSUN datasets were used as abnormal samples in our experiments. Since the objective is to distinguish between in-distribution and abnormal observations, the outlier detection task can be framed as a binary classification problem. Utilizing the weighted confidence formulation described in Equation (18), we determined an appropriate threshold for outlier identification through cross-validation. To assess the performance of our model, we employed three evaluation metrics: the true negative rate (TNR) at a fixed true positive rate (TPR) of 95%, the area under the receiver operating characteristic curve (AUROC), and the detection accuracy. To benchmark the effectiveness of our proposed approaches, we compared them with the baseline method introduced by Hendrycks and Gimpel [14], ODIN by Liang, Li and Srikant [27], and the method based on Mahalanobis distance presented in [26].

In our experiment, we extracted the confidence scores from ResNet's residual block and DenseNet's dense block. These scores were then combined to obtain the final weighted confidence score. To tune the hyperparameters, we selected 1000 pairs of in-distribution samples and outliers. The logistic parameters were obtained through nested cross-validation using the validation dataset. The experimental results are presented in Table 2.

From Table 2, it is evident that almost all of the best results are achieved by methods based on Mahalanobis distance and T-type estimator. When the feature extraction network is DenseNet, the T-type estimator consistently yields the optimal results. On the other hand, when the feature extraction network is ResNet, the method based on Mahalanobis distance obtains the best results. In scenarios where the T-type estimator performs optimally, the other methods generally produce less satisfactory results. For instance, when the in-distribution samples are from CIFAR-10 and the outlier samples are from SVHN, with a DenseNet network structure, the T-type estimator achieves a remarkable TNR of 99.1%, whereas the other methods do not surpass 91% TNR. Similarly, when the method based on Mahalanobis distance achieves the best results, the differences between the other methods and the Mahalanobis distance results are relatively minor. For example, when the in-distribution samples are from CIFAR-100 and the outlier samples are from SVHN, with a ResNet network structure, the TNR achieved by the Mahalanobis distance method is 91.9%, and the TNR of the other methods is around 91%. These findings suggest that the T-type estimator significantly improves upon the limitations of existing methods and is capable of maintaining detection performance similar to the best methods in many scenarios. This underscores the robustness of the T-type estimator and its potential to address shortcomings in outlier detection tasks effectively.

4.2 Adversarial detection

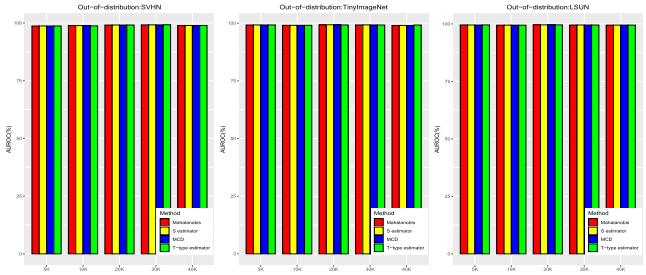
Setup. In adversarial sample detection, DenseNet and ResNet are employed for training on CIFAR-10, CIFAR-100, and SVHN datasets to construct pre-training models. These training datasets are considered as in-distribution samples (positive), while adversarial samples (negative) are generated based on the training dataset using four methods: FGSM, BIM, DeepFool, and C&W. For comparison,

Table 3. Adversarial detection (all numbers are AUROC(%) and the best results are bolded)

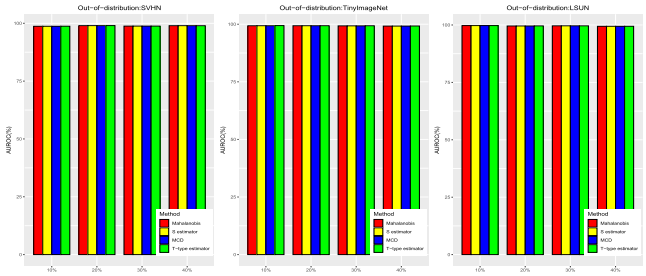
Model	Dataset	Score	Detection of known attack				Detection of unknown attack			
			FGSM	BIM	DeepFool	C&W	FGSM(seen)	BIM	DeepFool	C&W
DenseNet	CIFAR-10	KD+PU	85.96	96.80	68.05	58.72	85.96	3.10	68.34	53.21
		LID	98.20	99.74	85.14	80.05	98.20	94.55	70.86	71.50
		Mahalanobis	**99.94**	**99.78**	**83.41**	**87.31**	**99.94**	**99.51**	**83.42**	**87.95**
		S estimator	98.41	92.96	63.16	77.86	98.41	96.79	48.69	44.34
		MCD	99.50	86.93	69.85	76.67	99.50	94.94	58.29	55.16
		T-type estimator	99.77	97.12	61.08	77.71	99.77	98.57	59.89	57.83
	CIFAR-100	KD+PU	90.13	89.69	68.29	57.51	90.13	66.86	65.30	58.08
		LID	99.35	98.17	70.17	73.37	99.35	68.62	69.68	72.36
		Mahalanobis	**99.86**	**99.17**	**77.57**	**87.05**	**99.86**	**98.27**	**75.63**	**86.20**
		S estimator	96.93	86.39	67.17	72.61	96.93	93.24	61.81	66.27
		MCD	96.85	94.65	66.68	77.72	96.85	90.51	60.33	62.81
		T-type estimator	99.45	98.40	66.75	74.41	99.45	90.36	66.02	71.02
	SVHN	KD+PU	86.95	82.06	89.51	85.68	86.95	83.28	84.38	82.94
		LID	99.35	94.87	91.79	94.70	99.35	92.21	80.14	85.09
		Mahalanobis	**99.85**	**99.28**	**95.10**	**97.03**	**99.85**	**99.12**	**93.47**	**96.95**
		S estimator	99.80	99.06	88.53	96.47	99.80	98.31	74.12	95.79
		MCD	99.72	98.74	86.13	95.67	99.72	98.51	85.81	95.47
		T-type estimator	99.71	99.16	85.97	96.73	99.71	96.87	88.91	93.47
ResNet	CIFAR-10	KD+PU	81.21	82.28	81.07	55.93	83.51	16.16	76.80	56.30
		LID	99.69	96.28	88.51	82.23	99.69	95.38	71.86	77.53
		Mahalanobis	**99.94**	**99.57**	**91.57**	**95.84**	**99.94**	**98.91**	78.06	**93.90**
		S estimator	99.80	98.42	83.87	93.80	99.80	97.79	80.08	91.96
		MCD	99.78	98.01	84.79	90.71	99.78	97.99	**80.70**	90.49
		T-type estimator	99.92	99.14	84.47	95.04	99.92	98.71	75.60	93.39
	CIFAR-100	KD+PU	89.90	83.67	80.22	77.37	89.90	68.85	57.78	73.72
		LID	98.73	96.89	71.95	78.67	98.73	55.82	63.15	75.03
		Mahalanobis	**99.77**	96.90	**85.26**	**91.77**	**99.77**	96.38	**81.95**	**90.96**
		S estimator	99.68	**96.99**	75.87	90.59	99.68	**97.04**	71.38	90.72
		MCD	99.69	96.45	77.45	90.61	99.69	96.42	75.06	90.86
		T-type estimator	99.70	96.78	77.43	91.37	99.70	96.55	73.13	90.80
	SVHN	KD+PU	82.67	66.19	89.71	76.57	82.67	43.21	84.30	67.85
		LID	97.86	90.74	92.40	88.24	97.86	84.88	67.28	76.58
		Mahalanobis	**99.62**	**97.15**	**95.73**	**92.15**	**99.62**	**95.39**	72.20	86.73
		S estimator	98.84	96.01	86.87	91.79	98.84	93.94	74.38	88.70
		MCD	98.86	95.75	86.99	92.06	98.86	93.98	**77.99**	88.95
		T-type estimator	99.04	96.48	87.28	91.86	99.04	95.21	66.52	**89.00**

methods based on robust Mahalanobis distance estimation (MCD, S, and T-type estimators) are compared with the original Mahalanobis distance-based method (Mahalanobis) [26], kernel density (KD) [7], predictive uncertainty (PU) [7], and local intrinsic dimensionality (LID) scores [30]. To train logistic parameters, 10% of the test data is randomly selected, and the remaining test set is used for model evaluation. Nested cross-validation and training data are utilized for hyperparameter tuning. Additionally, considering various scenarios where the specific attack type is unknown, we conducted further experiments for comparison. Specifically, we used in-distribution samples and their adversarial samples generated using FGSM to train a logistic classifier, which was then used to detect the other three types of adversarial samples. The results are presented in the rightmost column of Table 3.

From the table above, it is evident that methods based on Mahalanobis distance and three robust estimation techniques outperform the two baseline methods, KD+PU and LID. Overall, their performance ranks as follows: Mahalanobis $\succ$ T-type estimator $\succ$ MCD $\succ$ S estimator, where $\succ$ denotes 'outperforms'. Additionally, the Mahalanobis distance-based approach achieves superior detection performance in most scenarios. However, we should also note that there is no significant difference in numerical results between the Mahalanobis distance method and the three robust estimation methods. For instance, when in-distribution samples are from CIFAR-10, the network is ResNet, and the abnormal samples are generated using FGSM, the Mahalanobis distance yields an AUROC of 99.94%, while the other three robust estimation techniques achieve AUROC of 99.80%, 99.78%, and 99.92%, respectively. This indicates that the

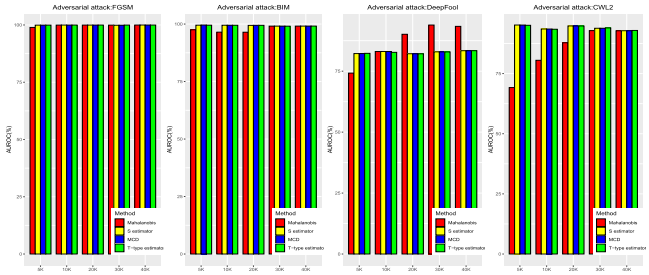


(a) Small training data: the x-axis represents the number of training data

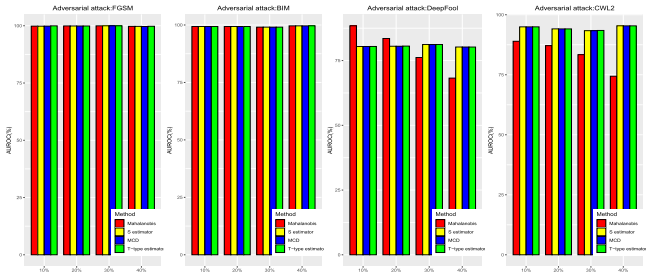


(b) Noisy training data: the x-axis represents the percentage of training data with random label

Figure 6. Comparison of AUROC (%) under extreme scenarios: (a) small number of training data. (b) Random label is assigned to training data on CIFAR-10 dataset.



(a) Small training data: the x-axis represents the number of training data



(b) Noisy training data: the x-axis represents the percentage of training data with random label

Figure 7. Comparison of AUROC (%) under different training data. To evaluate the robustness of the methods based on Mahalanobis distance, we train ResNet (a) by varying the number of training and (b) by assigning random labels to training data on the CIFAR-10 dataset.

three robust estimation methods are also highly competitive and may achieve optimal performance in more complex scenarios.

4.3 Comparison of robustness

In this section, we designed more complex scenarios to compare the method based on Mahalanobis distance with the three robust estimation methods. The entire detection process includes training the neural network, extracting features at each layer, estimating the mean vectors and covariance matrices, and training a logistic model used for the detection process. Since our experiments were conducted based on a pre-trained model, we intentionally made the second and third steps more complex to conduct a robustness comparison.

In the context of abnormal detection, we used CIFAR-10 as the in-distribution sample and employed SVHN, TinyImageNet, and LSUN as the abnormal samples. For adversarial detection, we utilized CIFAR-10 as the in-distribution sample and generated abnormal samples based on the four adversarial sample generation algorithms discussed in Section 2. Two scenarios were considered for mean vector and covariance matrix estimation: data proportion and random labels. The experimental results are presented in Figures 6 and 7. Throughout all the experiments, ResNet was used as the network architecture.

From Figures 6 and 7, it can be observed that the abnormal detection performance of the Mahalanobis distance-based method and the three robust estimation methods remain unchanged with variations in the sample ratio and random label ratio. Furthermore, all four methods exhibit excellent detection performance. In adversarial sample detection, regardless of changes in sample size or random label ratios, the three robust estimation methods can maintain similar levels of detection performance. However, the detection performance of the method based on Mahalanobis distance varies greatly, indicating the importance of robust Mahalanobis distance estimation. When the adversarial attacks are generated by FGSM and BIM methods, the three robust estimation methods exhibit almost identical detection performance to the Mahalanobis distance-based method, with a slight advantage in some scenarios, such as when the sample ratio is 5k and the adversarial attack is BIM. In scenarios where the adversarial attacks are DeepFool and CML2, the majority of the three robust estimation methods outperform the Mahalanobis distance-based method. It is worth noting that DeepFool and CML2 are relatively the strongest among the four attack methods, which further emphasizes the necessity and importance of robust Mahalanobis distance estimation.

Our second robustness comparison experiment involved

Table 4. Detection results at different proportions (all numbers are AUROC(%) and the best results are bolded)

Model	Estimator	Proportions				
		(0.1, 0.9, 0.0)	(0.1, 0.8, 0.1)	(0.1, 0.6, 0.3)	(0.1, 0.2, 0.7)	(0.1, 0.0, 0.9)
ResNet	Mahalanobis	**99.48**	96.44	96.44	96.44	**96.34**
	S	90.12	79.54	79.54	79.54	95.18
	MCD	99.16	**99.29**	**99.29**	**99.29**	95.81
	T	97.94	86.98	86.98	86.98	90.23
DenseNet	Mahalanobis	92.41	85.00	85.00	85.00	89.67
	S	92.64	97.31	97.31	97.31	94.29
	MCD	95.43	95.43	95.43	95.43	96.40
	T	**95.51**	**97.43**	**97.43**	**97.43**	**97.23**

Three proportions represent, in order, the proportions of in-distribution samples, outlier samples, and adversarial samples.

training a logistic classifier with a combination of in-distribution, outlier, and adversarial samples. In this setup, the outliers and adversarial samples were considered as the negative class, while CIFAR-10 served as the positive class. Specifically, CIFAR-10 was used as the in-distribution sample, SVHN as the outlier sample, and FGSM-generated samples based on CIFAR-10 as the adversarial sample. The test set consisted of a proportion of 0.33 for in-distribution samples, 0.33 for outliers, and 0.34 for adversarial samples. The experimental results are presented in Table 4.

From Table 4, it is evident that for the ResNet network architecture, the methods based on Mahalanobis distance and MCD estimation achieve the best performance, while the S estimator performs the worst. For the DenseNet network architecture, the T-type estimator achieves the best performance, generally followed by the S estimator, while the method based on Mahalanobis distance exhibits the poorest performance. This implies that the method based on Mahalanobis distance is not universally effective in all scenarios. In more complex scenarios, robust estimation-based methods are superior.

4.4 Training time

Given a pre-trained model, the total computation time for abnormal detection includes the time for feature extraction, estimation of mean vectors and covariance matrices at each layer, calculation of confidence scores, and the discriminative process based on logistic regression. Therefore, the difference among the methods lies in the estimation time of mean vectors and covariance matrices, which are provided in the table below. The results indicate that the computation time for sample mean and sample covariance is clearly the shortest, followed by the T-estimator, while the MCD estimator has the longest computation time among all the robust estimators. Furthermore, the training time for the ResNet model is significantly shorter compared to the DenseNet model.

Table 5. Comparison of training time (in seconds)

In-dist (model)	Mahalanobis	S	MCD	T-type estimator
CIFAR-10 (Densenet)	19.0	410.2	1546.4	386.1
CIFAR-100 (Densenet)	24.7	498.2	1792.5	163.2
SVHN (Densenet)	23.0	985.1	1695.9	163.2
CIFAR-10 (Resnet)	0.4	212.4	644.8	178.5
CIFAR-100 (Resnet)	0.6	251.6	916.5	175.2
SVHN (Resnet)	0.7	464.2	986.8	42.3

5. CONCLUSION

In this article, the robust estimation of Mahalanobis distance is proposed to process abnormal sample detection. In terms of the powerful feature extraction capabilities of deep neural networks, besides image data, one can apply modified inception V3 and ResNest-50 models to audio data [15], and apply ConvNet with embedding FastText and LSTM models to text data [21], to obtain the pre-trained features for classification and detecting abnormal samples, so that the proposed method using various deep convolutional neural networks can be applied to not only vision datasets but also text, audio datasets. Deep learning models generally require a large number of training samples to achieve a better performance; but in actual application scenarios, they may not have so many training samples. At this time, the method of the article can be extended to high-dimensional scenarios, in which the training samples are limited, the neural network model is determined and the number of hidden layers or the feature dimension is higher. In this way, the Mahalanobis distance in high-dimensional scenarios can be robustly estimated, such as MRCD [1] and MDP [36] for abnormal sample detection.

Abnormal sample detection based on Mahalanobis distance can achieve better performance on the pre-training model. When the strength of the adversarial sample increases, or the outlier sample has a high degree of similarity to the training sample, the abnormal sample detection based on Mahalanobis distance is equivalent to having anomalies

in the features extracted by the deep neural network value. The methods based on MCD and T-type estimators have similar performance to the method based on Mahalanobis distance in most scenes, but the performance in some scenes is greatly improved;

It is often used in preprocessing to remove anomalous data, which is done for several significance. After the abnormal samples are detected and removed from the dataset, it allows the learning algorithm to learn a more accurate model and improve its predictive utility; the statistics of data such as the mean and standard deviation are closer to the corresponding true values of the population, which results in a statistically significant increase in accuracy, and the visualization of data can also be improved. Anomalies are also often the most important observations in the data to be found such as in intrusion detection or detecting abnormalities in medical images.

ACKNOWLEDGEMENTS

This work is supported by Key Projects of the National Natural Science Foundation of China (No: 12031016), the National Natural Science Foundation of China (Nos: 11971324, 11471223, 12201435), Beijing Postdoctoral Research Foundation (No: 2022-ZZ-084), the Interdisciplinary Construction of Bioinformatics and Statistics, and the Academy for Multidisciplinary Studies, Capital Normal University. The authors also thank Dr. Zhu Xiaoning of Beijing University of Posts and Telecommunications for his help and guidance in the experimental simulation process.

APPENDIX A. TECHNICAL DETAILS ON THE INFLUENCE FUNCTION OF T-TYPE ESTIMATION

A.1 Proof of Theorem 3.2

For a given $\nu > 0$, recall the loss function $l(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{z})$ on the right side of (9)

$$l(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{z}) = \log|\boldsymbol{\Sigma}| + (\nu + p)\log\left(1 + \frac{(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})}{\nu}\right)$$

which is opposite to the logarithm of (8). Since

$$\mathrm{d}\log|\boldsymbol{\Sigma}| = |\boldsymbol{\Sigma}|^{-1}\mathrm{d}|\boldsymbol{\Sigma}| = \mathrm{tr}\big(\boldsymbol{\Sigma}^{-1}\mathrm{d}\boldsymbol{\Sigma}\big)$$

and

$$\begin{aligned}
&\mathrm{d}\log\left(1 + \frac{(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})}{\nu}\right) \\
=&\left(1 + \frac{(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})}{\nu}\right)^{-1} \frac{1}{\nu}(\boldsymbol{z}-\boldsymbol{\mu})^\top \mathrm{d}\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) \\
=&-\frac{(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}\mathrm{d}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})}{\nu + (\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})} \\
=&-\frac{\mathrm{tr}(\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}\mathrm{d}\boldsymbol{\Sigma})}{\nu + (\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})},
\end{aligned}$$

we have $\mathrm{d}\log|\boldsymbol{\Sigma}|/\mathrm{d}\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{-1}$, and

$$\frac{\mathrm{d}\log(1 + \frac{(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})}{\nu})}{\mathrm{d}\boldsymbol{\Sigma}} = -\frac{\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}}{\nu + (\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})}.$$

Then,

(19)
$$\begin{aligned}
0 &= E_F\left(\frac{\partial l(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{Z})}{\partial \boldsymbol{\Sigma}}\right) \\
&= \boldsymbol{\Sigma}(F)^{-1} - (\nu+p)E_F\left(\frac{\boldsymbol{\Sigma}(F)^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})(\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(F)^{-1}}{\nu + (\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(F)^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})}\right),
\end{aligned}$$

(20)
$$\begin{aligned}
0 &= E_F\left(\frac{\partial l(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{Z}, \nu)}{\partial \boldsymbol{\mu}}\right) \\
&= -E_F\frac{(\nu+p)\boldsymbol{\Sigma}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu}(F))}{\nu + (\boldsymbol{Z}-\boldsymbol{\mu}(F))^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu}(F))}.
\end{aligned}$$

(19) is equivalent to

(21)
$$\boldsymbol{\Sigma}(F) - (\nu+p)E_F\left(\frac{(\boldsymbol{Z}-\boldsymbol{\mu})(\boldsymbol{Z}-\boldsymbol{\mu})^\top}{\nu + (\boldsymbol{Z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(F)^{-1}(\boldsymbol{Z}-\boldsymbol{\mu})}\right) = 0.$$

Then, substituting $F_\epsilon = (1-\epsilon)F + \epsilon\Delta_{\boldsymbol{x}}$ for F in (20) and (21) and taking the derivative w.r.t. ϵ at $\epsilon = 0$, we have

(22)
$$\begin{aligned}
&E_F\frac{(\nu+p)\boldsymbol{\Sigma}^{-1}IF(\boldsymbol{x}, \widehat{\boldsymbol{\mu}}_T, F)}{\nu + (\boldsymbol{Z}-\boldsymbol{\mu}(F))^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu}(F))} \\
&- E_F\frac{2(\nu+p)\boldsymbol{\Sigma}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu}(F))(\boldsymbol{Z}-\boldsymbol{\mu}(F))^\top \boldsymbol{\Sigma}^{-1}IF(\boldsymbol{x}, \widehat{\boldsymbol{\mu}}_T, F)}{[\nu + (\boldsymbol{Z}-\boldsymbol{\mu}(F))^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{Z}-\boldsymbol{\mu}(F))]^2} \\
&= \frac{(\nu+p)\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}(F))}{\nu + (\boldsymbol{x}-\boldsymbol{\mu}(F))^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}(F))}.
\end{aligned}$$

From (22), the influence functions of $\widehat{\boldsymbol{\mu}}_T$ and (15) hold.

When $\boldsymbol{Z}$ satisfies (6). using (21), (15) can be rewritten as

$$\begin{aligned}
\big(M_{i,j}(\boldsymbol{x})\big)_{p\times p} &= \big(h_{i,j}(\boldsymbol{x})\big)_{p\times p} \\
&\quad + (\nu+p)E_F\frac{r^2\boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{U}^\top(h_{i,j}(\boldsymbol{x}))_{p\times p}\boldsymbol{U}}{(\nu+r^2)^2}.
\end{aligned}$$

By calculation, we complete the proof.

A.2 Additional numerical results

A.2.1 Influence function

Under the bivariate $t(1)$ distribution with $\boldsymbol{\mu} = (\mu_1, \mu_2)^\top = \mathbf{0}$ and $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_{i,j})_{2\times 2} = (0.5^{|i-j|})_{2\times 2}$, Figures A.1 and A.2 further show the graphs of influence functions of T-type estimators for μ_2, $\boldsymbol{\Sigma}_{2,1}$ and $\boldsymbol{\Sigma}_{2,2}$, respectively. It can be seen that the influence functions of T-type

scale estimator for $\mathbf{\Sigma}_{1,2}$ and $\mathbf{\Sigma}_{2,1}$ are the same. By swapping the X and Y axis, Figures 4 and A.1 would have the same pattern, and so do the influence functions of $\widehat{\mathbf{\Sigma}}_T$ for the diagonal entries.

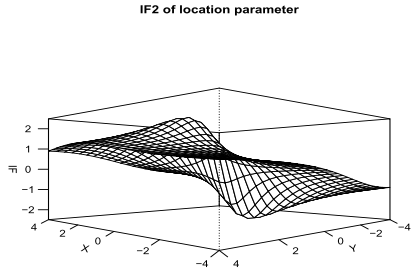


Figure A.1. Influence function of $\widehat{\boldsymbol{\mu}}_T$ at $t(1)$ model for the second element of T-type location estimator.

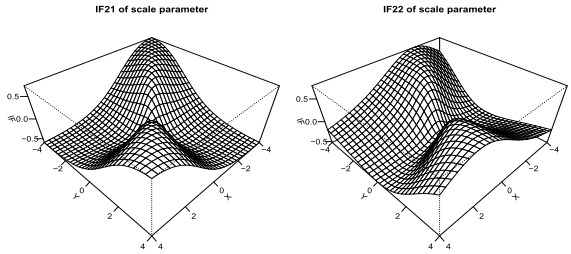


Figure A.2. Influence function of $\widehat{\mathbf{\Sigma}}_T$ at $t(1)$ model, IF21 for the off-diagonal entry $\mathbf{\Sigma}_{2,1}$ and IF22 for the second diagonal entry $\mathbf{\Sigma}_{2,2}$.

APPENDIX B. ADDITIONAL SIMULATION STUDIES FOR ABNORMAL DETECTION

In this subsection, we further explore the detection performance and robustness of four Mahalanobis distance-based methods in the presence of both OOD and adversarial samples through simulation experiments.

Given that the CNNs used in this paper extract features with different dimensions for each hidden layer, we set two layers in the simulation study. For the first layer, the feature of the in-distribution sample follows a 5-dimensional Gaussian mixture distribution $0.5\mathcal{N}_5(\boldsymbol{\mu}_1, \mathbf{\Sigma})+0.5\mathcal{N}_5(\boldsymbol{\mu}_2, \mathbf{\Sigma})$, and the OOD samples are generated from $\mathcal{N}_5(\boldsymbol{\mu}_3, \boldsymbol{I}_5)$, where $\boldsymbol{\mu}_1 = (2, \ldots, 2)^\top$, $\boldsymbol{\mu}_2 = (-2, \ldots, -2)^\top$, $\boldsymbol{\mu}_3 = (5, \ldots, 5)^\top$, $\mathbf{\Sigma} = (0.5^{|i-j|})_{p\times p}$. The dimension of the feature extracted by the second layer is ten, where the first five dimensions for both in-distribution and OOD samples are the same as the first layer, while the remaining five dimensions are drawn from a standard normal distribution. In other words, the last five dimensions of the extracted features in the second layer are noise following a standard normal distribution, while the first five dimensions maintain the classification characteristics from the first layer. During the training and testing phases, the sample size for in-distribution data is 1000, and the sample size for OOD data is 500. We consider the following two ways for generating adversarial samples:

- AD_1: Randomly select 500 observations from the in-distribution samples and add $t(1)$ random variables to each dimension of these 500 in-distribution observations;
- AD_2: Randomly select 500 observations from the in-distribution samples and add a constant value of 3 to each of these 500 in-distribution observations.

In order to assess the robustness of the four methods, during the training phase, we considered two types of perturbations. Firstly, we introduced a scenario where 30% of the labels were randomly confused and evenly distributed among the in-distribution, OOD, and adversarial samples. Secondly, we added a constant perturbation of 100 to one observation in the in-distribution sample. These two perturbation settings were referred to as "False label" and "1st obs contam." respectively, while the training set without any perturbations was referred to as "clean."

Table B.1. Abnormal detection (all numbers are TNR at TPR 95(%) and the best results are bolded)

Estimator	Clean		False Label		1st obs contam.	
	AD_1	AD_2	AD_1	AD_2	AD_1	AD_2
Mahalanobis	99.1	99.7	92.2	29.4	93.7	36.5
S	99.0	99.7	98.0	33.6	98.7	**99.9**
MCD	**99.2**	99.7	**98.8**	**99.9**	**98.9**	**99.9**
T	99.1	99.7	95.7	37.5	**98.9**	**99.9**

Table B.1 presents the detection accuracy of the four Mahalanobis distance-based abnormal detection methods under three perturbation settings for common OOD samples and two types of adversarial samples, measured by the TNR at TPR of 95(%). In the scenario of "1st obs contam.," the contaminated observation leads to a significantly large confidence score. Due to logistic regression being based on a linear model, it is sensitive to outliers, which greatly affects the fitting results when using such a large confidence score as an explanatory variable. To mitigate this effect, in the "1st obs contam." scenario, we removed confidence scores of in-distribution samples greater than the 0.997 quantile of the chi-square distribution with degrees of freedom equal to the feature dimension of each layer, before training the logistic regression detector. The results in Table 6 showed that under the "clean" scenario, the four methods performed similarly in abnormal detection. However, in the cases of "False label" and "1st obs contam", the robust Mahalanobis distance demonstrated significantly better performance than Mahalanobis, with MCD showing the best performance. In particular, when there exist false labels and AD_2 is used

as adversarial samples, MCD that selects a subsample with minimum covariance determinant for estimating the center and scatter matrix outperforms S and T-type estimators, which mitigates the influence of outliers by assigning weights to individual data points. However, the weights assigned by S and T-type estimators in such a scenario do not exhibit differences in orders of magnitude. Hence, the estimation of the center and scatter matrix by S and T-type estimators is not as effective as MCD, leading to inferior classification performance.

Considering the sensitivity of the logistic model to outliers and its reliance on Mahalanobis confidence scores for abnormal detection, we further discussed another approach to abnormal detection that directly utilizes these Mahalanobis confidence scores. The confidence scores of each observation at each layer are transformed with the cumulative distribution function of the chi-square distribution with degrees of freedom equal to the feature dimension of that layer. The transformed confidence scores are then used to determine the threshold for distinguishing abnormal samples during the training phase based on maximizing TNR at TPR 95(%). Table B.2 presents the detection accuracy of this method. The results show that, under the clean condition, the performance of all four methods remains similar. However, under False label and 1st obs contam., the robust Mahalanobis distance classifier outperforms the classical one, with the best performance observed in the T-type estimator. Nevertheless, compared to the results obtained using the logistic model in Table B.2, this distance-based discrimination method sacrifices some detection accuracy. Integrating the confidence scores from each layer in a robust manner will also be a future consideration.

Table B.2. Abnormal detection by distance discriminant (all numbers are TNR at TPR 95(%) and the best results are bolded)

Estimator	Clean		False Label		1st obs contam.	
	AD_1	AD_2	AD_1	AD_2	AD_1	AD_2
Mahalanobis	93.2	76.9	72.1	29.1	56.1	15.3
S	93.2	75.9	91.2	45.4	93.9	78.2
MCD	93.2	76.8	92.5	**77.8**	93.9	78.3
T	**93.3**	**77.6**	**99.9**	33.9	**94.9**	**79.8**

Received 3 November 2022

REFERENCES

[1] Boudt, K., Rousseeuw, P. J., Vanduffel, S. and Verdonck, T. (2020). The minimum regularized covariance determinant estimator. *Statistics and Computing* **30** 113–128. MR4057474

[2] Butler, R. W., Davies, P. L. and Jhun, M. (1993). Asymptotics for the minimum covariance determinant estimator. *The Annals of Statistics* **21** 1385–1400. MR1241271.

[3] Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)* 39–57.

[4] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40** 834–848.

[5] Croux and Haesbroeck (1999). Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. *Journal of Multivariate Analysis* **71** 161–190. MR1735108.

[6] Deldjoo, Y., Noia, T. D. and Merra, F. A. (2020). Adversarial machine learning in recommender systems (aml-recsys). In *WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining* 869–872. Association for Computing Machinery, New York, NY, United States.

[7] Feinman, R., Curtin, R. R., Shintre, S. and Gardner, A. B. (2017). Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

[8] Feng, F., He, X., Tang, J. and Chua, T.-S. (2019). Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering* **33** 2493–2504.

[9] Goodfellow, I. J., Shlens, J. and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

[10] He, X., Simpson, D. and Wang, G. (2000). Breakdown points of t-type regression estimators. *Biometrika* **87** 675–687. MR1789817.

[11] He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 770–778.

[12] He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* 2961–2969. IEEE, Venice, Italy.

[13] Heaton, J., Polson, N. and Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry* **33** 3–12. MR3615257.

[14] Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

[15] Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Aren Jansen, R. C. M., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J. and Wilson, K. (2017). CNN architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 131–135. IEEE.

[16] Huang, G., Liu, Z. and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 4700–4708.

[17] Hubert, R. and Aelst, V. (2008). High breakdown robust multivariate methods. *Statistical Science* **23** 92–119. MR2431867.

[18] Hubert, M., Debruyne, M. and Rousseeuw, P. J. (2018). Minimum covariance determinant and extensions. *Wiley Interdisciplinary Reviews: Computational Statistics* **10** e1421. MR3799918.

[19] Insua, D. R., Naveiro, R., Gallego, V. and Poulos, J. (2023). Adversarial machine learning: Bayesian perspectives. *Journal of the American Statistical Association* 1–12.

[20] Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association* **79** 871–880. MR0770281.

[21] Köksal, O. and Akgül, O. (2022). A comparative text classification study with deep learning-based algorithms. In *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)* 387–391. IEEE.

[22] Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM* **60** 84–90.

[23] Kurakin, A., Goodfellow, I. and Bengio, S. (2018). Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*. Chapman and Hall/CRC 99–112.

[24] Kwon, H. and Lee, S. (2022). Ensemble transfer attack targeting text classification systems. *Computers & Security* **117** 102695.
[25] Lauriola, I., Lavelli, A. and Aiolli, F. (2022). An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing* **470** 443–456.
[26] Lee, K., Lee, K., Lee, H. and Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems* **31**.
[27] Liang, S., Li, Y. and Srikant, R. (2017). Principled detection of out-of-distribution examples in neural networks. *arXiv preprint arXiv:1706.02690* 655–662.
[28] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Laak, J. A. V. D., Ginneken, B. V. and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis* **42** 60–88.
[29] M., G. and Sethuraman, S. C. (2023). A comprehensive survey on deep learning based malware detection techniques. *Computer Science Review* **47** 100529.
[30] Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M. E. and Bailey, J. (2018). Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*.
[31] Machado, G. R., Silva, E. and Goldschmidt, R. R. (2021). Adversarial machine learning in image classification: A survey towards the defender's perspective. *ACM Computing Surveys (CSUR)* **55** 1–38.
[32] Maronna, R. A. (1976). Robust M-estimators of multivariate location and scatter. *The Annals of Statistics* **4** 51–67. MR0388656.
[33] Mero, L. L., Yi, D., Dianati, M. and Mouzakitis, A. (2022). A Survey on Imitation Learning Techniques for End-to-End Autonomous Vehicles. In *IEEE Transactions on Intelligent Transportation Systems 9* **23** 14128–14147.
[34] Moosavi-Dezfooli, S.-M., Fawzi, A. and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2574–2582. IEEE.
[35] Rebuffi, S., Kolesnikov, A., Sperl, G. and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2001–2010.
[36] Ro, K., Zou, C., Wang, Z. and Yin, G. (2015). Outlier detection for high-dimensional data. *Biometrika* **102** 589–599. MR3394277.
[37] Rousseeuw, P. and Driessen, K. V. (1999). A fast algorithm for the Minimum Covariance Determinant estimator. *Technometrics* **41** 212–223.
[38] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A. and Bernstein, M. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* **115** 211–252.
[39] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R. (2013a). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
[40] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R. (2013b). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
[41] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1–9.
[42] Tramèr, F., Kurakin, A., Papernot, N., Ian Goodfellow, D. B. and McDaniel, P. (2017). Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.
[43] Xu, H., Ma, Y., Liu, H., Deb, D., Liu, H., Tang, J. and Jain, A. K. (2020). Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing* **17** 151–178.
[44] Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M. and Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing* **126** 103514.

Wan Tian
Capital Normal University
Beijing
China
E-mail address: wantian61@foxmail.com

Lingyue Zhang
Capital Normal University
Beijing
China
E-mail address: lingyue_zhang@126.com

Hengjian Cui
Capital Normal University
Beijing
China
E-mail address: hjcui@bnu.edu.cn