Effective and Robust Detection of Adversarial Examples via Benford-Fourier Coefficients

Cheng-Cheng Ma1,2*, Bao-Yuan Wu3,4*, Yan-Bo Fan5, Yong Zhang5, Zhi-Feng Li5

1 National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
2 School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
3 School of Data Science, The Chinese University of Hong Kong, Shenzhen 518172, China
4 Shenzhen Research Institute of Big Data, Shenzhen 518172, China
5 AI Lab, Tencent Inc., Shenzhen 518057, China

Abstract: Adversarial example has been well known as a serious threat to deep neural networks (DNNs). In this work, we study the detection of adversarial examples based on the assumption that the output and internal responses of one DNN model for both adversarial and benign examples follow the generalized Gaussian distribution (GGD) but with different parameters (i.e., shape factor, mean, and variance). GGD is a general distribution family that covers many popular distributions (e.g., Laplacian, Gaussian, or uniform). Therefore, it is more likely to approximate the intrinsic distributions of internal responses than any specific distribution. Besides, since the shape factor is more robust to different databases rather than the other two parameters, we propose to construct discriminative features via the shape factor for adversarial detection, employing the magnitude of Benford-Fourier (MBF) coefficients, which can be easily estimated using responses. Finally, a support vector machine is trained as an adversarial detector leveraging the MBF features. Extensive experiments in terms of image classification demonstrate that the proposed detector is much more effective and robust in detecting adversarial examples of different crafting methods and sources compared to state-of-the-art adversarial detection methods.

Keywords: Adversarial defense, adversarial detection, generalized Gaussian distribution, Benford-Fourier coefficients, image classification.

Citation: C. C. Ma, B. Y. Wu, Y. B. Fan, Y. Zhang, Z. F. Li. Effective and robust detection of adversarial examples via Benford-Fourier coefficients. Machine Intelligence Research, vol.20, no.5, pp.666-682, 2023. http://doi.org/10.1007/s11633-022-1328-1

1 Introduction

Deep neural networks (DNNs) have achieved remarkable success in many important applications, such as image classification1-4, facial recognition5-10, object detection11, 12, etc. However, DNNs have been shown to be very vulnerable to adversarial examples. Many real-world scenarios have very restrictive requirements about the robustness of DNNs, such as face verification for login or semantic segmentation in autonomous driving. Therefore, adversarial example is a serious threat to the applications of DNNs to these important tasks. Since many kinds of adversarial attack methods have been proposed to fool DNNs, it is more urgent to equip effective defensive strategies to ensure the safety of deep models in real-world applications. However, defense seems to be more challenging than attack, as it has to face adversarial examples from unknown crafting methods and unknown data sources. Typical defensive strategies include adversarial training, adversarial denoising, and adversarial detection. Compared to the former two strategies, adversarial detection is somewhat more cost-effective, as it often needs no re-training or modifying of the original model.

There are two main challenges for adversarial detection. 1) The adversarial examples are designed to camouflage themselves to be close to the corresponding benign examples in the input space. Then, the challenge is to identify where and how to extract the discriminative information to train the detector. 2) The data sources and the generating methods of adversarial examples are often inaccessible to the detector. In this case, it is essential to validate if the detector can be stably effective across different data sources and different attack methods. In other words, a good adversarial detector is required to be not only effective to distinguish adversarial and benign examples but also robust to different data sources and attack methods.

To satisfy the first requirement of effectiveness, we utilize the other principle of crafting adversarial ex-
amples that the outputs between benign and adversarial examples should be large to encourage the change of the final prediction. It means that the imperceptible difference between benign and adversarial examples in the input space is enlarged along with the DNN model, leading to a significant difference in the output space. Inspired by this fact, we assume that the output or responses of internal layers of the DNN model should include discriminative information for benign and adversarial examples. A few works have attempted to extract different types of discriminative features from the output or internal responses, such as kernel density (KD)\cite{13} and local intrinsic dimensionality (LID)\cite{14}, etc. In order to achieve robustness, the extracted discriminative features should model the intrinsic difference between adversarial and benign examples rather than the difference from the changes in data sources or attack methods. Many existing methods have shown effectiveness in detecting adversarial examples of specific data sources and attack methods. However, their robustness, especially across different data sources, has not been well studied or verified.

In this work, we propose a novel detection method based on the assumption that the internal responses of both adversarial and benign examples follow the generalized Gaussian distribution (GGD)\cite{15} but with different parameters, including the shape factor, mean, and variance. The rationale behind this assumption is that GGD covers many popular distributions with varied shape factors (such as the Laplacian, Gaussian, or uniform distribution), so that GGD is more likely to approximate the intrinsic response distributions rather than one specific distribution. Moreover, the mean and variance of GGD may vary significantly with respect to different classes and data sources, even for benign examples, while the shape factor is more robust than them. For example, the mean and variance of two Gaussian distributions could be totally different, but their numbers of shape factors are the same (both of them equal to 2). Thus, we propose to use the shape factor as effective and robust discrimination between adversarial and benign examples. However, it is difficult to estimate the shape factor exactly in practice. We resort to the magnitude of the Benford-Fourier (MBF) coefficients\cite{16}, which is a function of the shape factor. It can be easily estimated using internal responses, according to the definition of the Fourier transform. Then, the magnitudes estimated from internal responses of different convolutional layers are concatenated as a novel representation. Finally, a support vector machine (SVM)\cite{17} is trained using the new representations as to the adversarial detector. Extensive experiments carried out in several databases verify the effectiveness and robustness of the proposed detection method. To further verify the rationale of our assumption, we present the empirical analysis using the Kolmogorov-Smirnov test (KS test)\cite{18}. The KS test verifies that 1) the posterior vectors of both adversarial and benign examples predicted by the CNN model follow the distribution of GGD but with different parameters, and 2) the MBF features of adversarial and benign examples follow different distributions, and the MBF features of adversarial examples crafted from different attack methods follow the same distribution, as well as that the MBF features of adversarial/benign examples from different data sources follow the same distribution. Moreover, we visualize the statistics (i.e., mean ± standard deviation) of the extracted MBF features for adversarial and benign examples. The visualization reveals the distinct difference between adversarial and benign examples. This empirical analysis demonstrates the effectiveness and robustness of the proposed MBF method.

### 2 Related work

The general idea of most existing detection methods is learning or constructing a new representation to discriminate between adversarial and benign examples, utilizing the outputs or immediate responses of an original classification network. Li et al.\cite{19} trained a cascading classifier based on the principal component analysis (PCA)\cite{20} to detect adversarial examples using the statistical test of maximum mean discrepancy (MMD), and Gao et al.\cite{27} proposed a detection network along with an original classification network, which takes the internal responses of the original network as inputs. It shows effectiveness in detecting adversarial examples generated by simple attacks (such as fast gradient signed method (FGSM)\cite{23} and Jacobian-based saliency map attack (JSMA)\cite{24}) while performing much worse when faced with more advanced attacks (such as the Carlini and Wagner attack (C&W)\cite{25}). Moreover, it tells us that this method is sensitive to attack methods. Grosse et al.\cite{20} attempted to detect adversarial examples using the statistical test of maximum mean discrepancy (MMD), and Gao et al.\cite{27} recently suggested some enhancements to the traditional MMD method, such as replacing the Gaussian kernel with a deep kernel. Although the above detection methods show effectiveness in some attack methods and some databases, a thorough evaluation presented in \cite{28} has shown that these methods are sensitive to attack methods or data bases and can be quite easily invaded by new attacks.

Some recent works proposed to utilize neighboring samples in the same database to construct a better representation of a current sample. Feinman et al.\cite{13} defined two metrics based on the responses of the final hidden layer of the classification neural network, including kernel density estimation (KDE) and Bayesian neural network uncertainty (BU). If the metric score of KDE/BU is lower/higher than a pre-defined threshold, then the example is predicted as adversarial. Ma et al.\cite{14} utilized the local intrinsic dimensionality (LID) to measure the char-
acterization of adversarial regions of DNNs. LID describes the distance between an example and its $k$-nearest neighbor sample in the feature space of immediate responses of the original classification network. The distances computed from different layers are concatenated as the example representation, which is then used to train a shallow classifier to discriminate between adversarial and benign examples. Zheng and Hong[29] defined the intrinsic hidden state distribution (IHSID) of the responses of the original classification network to model different classes. The Gaussian mixture model (GMM) was used to approximate the IHSID of each class. Then, the posterior probability of one sample assigned to GMM is computed as the metric. If the probability is lower than a predefined threshold, then it is recognized as adversarial. Lee et al.[30] computed the class-conditional Gaussian distribution of the responses of the original classification network based on the whole training set. Then, the Mahalanobis distance between one sample and its nearest class-conditional Gaussian distribution is used as the metric for detection. If the distance is larger than a pre-defined threshold, then it is detected as adversarial. Compared to some aforementioned single-representation-based detection methods, these joint-representation-based methods showed better performance in some databases. However, the detection cost for each example is much higher, as the responses of its neighboring samples should also be computed. Besides, since the representation is highly dependent on the neighbors or all training examples, the detection performance may be sensitive to data sources, which will be studied in later experiments.

Beyond that, there are some approaches that utilize the statistical test to detect adversarial examples. For example, Roth et al.[31] took variations in the differences between values in the penultimate layer inside the classifier as a discriminative feature. Raghuram et al.[32] obtained and combined p-value scores per layer, then applied a multinominal statistical test. However, it is non-intuitive of this kind for detection methods to transfer across different attacks and databases.

There are also some other approaches that do not construct representations from the responses of the original classification network. Hendrycks and Dimpel[33] adopted PCA statistics to discriminate between adversarial and benign images, independent of any DNN model. However, the study presented in [28] has demonstrated that this method works for Modified National Institute of Standards and Technology Database (MNIST) but not for CIFAR-10, and PCA statistics are not robust features to detect adversarial images. Pang et al.[34] proposed a novel loss called reverse cross-entropy (RCE) to train the classification network, so that the distance measured by kernel density[13] between adversarial and benign examples could be enlarged. Samangouei et al.[35] proposed Defense-GAN to model the distribution of benign examples using a generative adversarial network (GAN). If the norm of the difference between one example and its corresponding example generated by GAN is larger than a threshold, then it is detected as adversarial. Hu et al.[36] designed two paradoxical criteria for adversarial examples and rejected the input as adversarial if at least one of the criteria is not satisfied. However, the above four methods are much more costly than other methods.

3 Preliminaries

3.1 Generalized Gaussian distribution

Assume that a random variable $X \in \mathbb{R}^d$ follows the generalized Gaussian distribution[15]. Then, its probability density function (PDF) is formulated with two positive parameters, including the shape factor $c$ and the standard deviation $\sigma$:

$$P_X(x) = A \times e^{-|\beta x|^c}$$  \hspace{1cm} (1)

where $\beta = \frac{1}{\sigma} \left( \Gamma(3/c) / \Gamma(1/c) \right)^{1/2}$ and $A = \frac{\beta c}{2 \Gamma(1/c)}$, with $\Gamma(\cdot)$ being the Gamma function. Note that the mean parameter $\mu$ is omitted above, as $\mu$ has no relation to the shape of the distribution, and we set it as 0 without loss of generality. A nice characteristic of GGD is that it covers many popular distributions with varied shape factors. For example, when $c = 1$, then it becomes the Laplacian distribution; when $c = 2$, then it is the Gaussian distribution with a variance of $\sigma^2$, when $c \to +\infty$, then it is specified as a uniform distribution on $(-\sqrt{2}\sigma, \sqrt{2}\sigma)$.

3.2 Benford-Fourier coefficients

Although the generalized Gaussian distribution is able to cover a lot of distributions, it is hard to accurately depict the exact forms of GGD. To this end, we further define a random variable $Z = \log|X| \mod 1$ for detecting and distinguishing different forms of GGD, of which the PDF is formulated by means of Fourier Series as [37], with the fundamental period being fixed as $2\pi$.

$$P_Z(z) = 1 + 2 \sum_{n=1}^{+\infty} \left| A_n \cos (2\pi n z) + B_n \sin (2\pi n z) \right| = 1 + 2 \sum_{n=1}^{+\infty} |a_n| \cos(2\pi n z + \phi_n)$$  \hspace{1cm} (2)

where $z \in [0, 1)$ corresponds to the domain of random variable $Z$, the phase of the Fourier series is explained as $\phi_n = \arctan \left( -\frac{B_n}{A_n} \right)$, and the magnitude denotes $|a_n| = \sqrt{A_n^2 + B_n^2}$, $a_n = |a_n| \times e^{i\phi_n}$ denotes the $n$-th Fourier coefficient of $P_Z(z)$ evaluated at $2\pi n$, and its definition is
\[ a_n = \int_{-\infty}^{+\infty} P_Z(z) \times e^{-2\pi n \log z} dz = \frac{2Ae^{2\pi n \log c}}{\beta c} \times \Gamma \left( \frac{2\pi n + \log 10}{c \log 10} \right). \] (3)

\( a_n \) is also called the Benford-Fourier coefficient\(^1\). Note that \( a_n \) is a complex number. According to Equation \((8.326)\) in \([38]\), its magnitude is calculated as

\[ |a_n| = \left( \prod_{k=0}^{\infty} \left[ 1 + \left( \frac{2\pi n}{\log(ck+1)} \right)^2 \right]^{-1} \right)^{\frac{1}{2}}. \] (4)

Note that \(|a_n|\) gets smaller as \(n \in \mathbb{N}\) increases. And, an interesting property of \(|a_n|\) is that it only depends on the shape factor \(c\), while it is independent of the parameter \(\sigma\). Thus, one set of the absolute values of Benford-Fourier coefficients \(\{|a_n|\}_{n \in \mathbb{N}}\) correspond to one identical \(c\), i.e., one identical special distribution of GGD. In other words, we could use \(\{|a_n|\}_{n \in \mathbb{N}}\) as features or representations to discriminate different special distributions of GGD.

However, if it is often difficult to know or even estimate the shape factor \(c\), we cannot compute the value of \(|a_n|\). But fortunately, recalling that \(a_n\) is the \(n\)-th Fourier coefficient of \(P_Z(z)\) evaluated at \(2\pi n\), we can derive an easy estimation. Specifically, assume that \(X = \{x_1, \cdots, x_M\}\) is a set of \(M\) i.i.d. points sampled from GGD with the same shape factor \(c\). Then, the corresponding Benford-Fourier coefficients can be estimated as follows\([66]\):

\[ \tilde{a}_n = \frac{1}{M} \sum_{m=1}^{M} e^{-2\pi n \log|X_m|} = \frac{1}{M} \sum_{m=1}^{M} \left[ \cos \left( 2\pi n \log|X_m| \right) - j \sin \left( 2\pi n \log|X_m| \right) \right]. \] (5)

The gap between \(\tilde{a}_n\) and \(a_n\) is analyzed in Theorem 1. It tells that \(\tilde{a}_n\) gets closer to \(a_n\) as \(M\) increases. For clarity, we firstly introduce a few notations: \(T = e^{-2\pi n \log|X|}\) is a random variable with \(X\) obeying the generalized Gaussian distribution, and \(\hat{a}_n\) is an observation of the random variable \(Y = \frac{1}{M} \sum_{m=1}^{M} T_m\). Due to the space limit, the proof will be presented in Appendix A.1.

**Theorem 1.** Assume that the estimation error \(\varepsilon_n = \tilde{a}_n - a_n\) is an observation of the random variable \(\mathcal{E} = Y - a_n\). \(|\mathcal{E}|\) follows the Rayleigh distribution\([86]\), of which the probability density function is formulated as

\[ P_{|\mathcal{E}|}(r) = 2Mr e^{-Mr^2}. \] (6)

And the expectation and variance are

\[ E(|\mathcal{E}|) = \frac{1}{2} \sqrt{\frac{\pi}{M}}, \quad D(|\mathcal{E}|) = \frac{4 - \pi}{4M}. \]

which implies that the estimation error \(\varepsilon_n\) gets closer to 0 as the number of samples \(M\) increases.

## 4 Adversarial detection via Benford-Fourier coefficients

### 4.1 Training procedure of adversarial detector

There are three stages to train the proposed adversarial detector: 1) building a training set based on benign images; 2) extracting novel representations of the training set via Benford-Fourier coefficients; 3) training an SVM classifier as the adversarial detector. They will be explained in detail sequentially. The overall training procedure is briefly summarized in Algorithm 1, and the symbols are listed in Table 1.

| Symbol | Meaning |
|--------|---------|
| \(x_i\) | The \(i\)-th clean image |
| \(\tilde{x}_i\) | The \(i\)-th adversarial image |
| \(\hat{x}_i\) | The \(i\)-th noisy image |
| \(D_{tr}\) | The training image set |
| \(f_{\theta}\) | The classification model |
| \(r_{l}^l\) | The \(l\)-th intermediate feature in the \(l\)-th layer of the \(i\)-th image |
| \((\tilde{a}_n)_l^l\) | The \(n\)-th Benford-Fourier coefficient in the \(l\)-th layer of the \(i\)-th image |

**Build a training set.** Firstly, we collect \(N\) clean images \(\{x_1, \cdots, x_N\}\), which can be correctly predicted by \(f_{\theta}\). Then, we adopt one adversarial attack method (e.g., C&W\([25]\) or BIM (basic iterative method)\([41]\)) to generate one adversarial image corresponding to each clean image. The crafted \(N\) adversarial examples are denoted as \(\{\tilde{x}_1, \cdots, \tilde{x}_N\}\). Besides, to avoid that the noisy image (polluted by some kinds of non-malicious noises but still can be correctly predicted by \(f_{\theta}\)) is incorrectly detected as adversarial, we also create one noisy image by adding small random Gaussian noises to each clean image. These \(N\) noisy examples are denoted as \(\{x_1, \cdots, x_N\}\). Note that, hereafter, benign examples include both clean and Gaussian noisy examples. Consequently, we obtain a training set with \(3N\) examples, denoted as \(D_{tr} = \{(x_i, -1), (\tilde{x}_i, -1), (\hat{x}_i, +1)\}_{i=1, \cdots, N}\).

**Extract discriminative representations.** We firstly feed the \(i\)-th training image from \(D_{tr}\) into \(f_{\theta}\). We concatenate all response entries of the \(l\)-th layer in \(f_{\theta}\) to

\[ E(|\mathcal{E}|) = \frac{1}{2} \sqrt{\frac{\pi}{M}}, \quad D(|\mathcal{E}|) = \frac{4 - \pi}{4M}. \]
obtain one vector \( \mathbf{r}_i^l \). Then, we estimate the corresponding Benford-Fourier coefficients according to (5), as follows:

\[
(\hat{a}_n)_i^l = \frac{1}{M_l^l} \sum_{m=1}^{M_l^l} e^{-2\pi i n \log |(r_m)_i^l|}.
\] (7)

**Algorithm 1.** Training the adversarial detector via the magnitude of Benford-Fourier coefficients.

**Require:** The trained CNN model \( f_\theta \) with \( L \) layers, and the training set \( \mathcal{D}_{tr} \).

**Ensure:** The trained binary SVM classifier

1) While \( i \leq |\mathcal{D}_{tr}| \) do
2) While \( l \leq L \) do
3) Compute \( |(\hat{a}_n)_i^l| \) as (8), with \( n = 1, \ldots, T \)
4) Concatenate \( \{ |(\hat{a}_n)_i^l| \}_{n=1}^{T} \) to obtain a vector \( \hat{a}_i^l \)
5) \( l = l + 1 \)
6) Concatenate \( \{ \hat{a}_i^l \}_{l=1}^{L} \) to obtain a long vector \( \hat{a}_i \)
7) end while
8) \( i = i + 1 \)
9) end while
10) Build a novel representation of the training set, denoted as \( \hat{\mathcal{A}}_i = \{ (\hat{a}_i, \pm 1) \}_{i=1}^{|\mathcal{D}_{tr}|} \)
11) Train a binary SVM classifier based on \( \hat{\mathcal{A}}_i \)

\( M_l^l \) indicates the length of \( \mathbf{r}_i^l \). The magnitude of \( |(\hat{a}_n)_i^l| \) is computed as follows:

\[
|(\hat{a}_n)_i^l| = \frac{1}{M_l^l} \left( \sum_{m=1}^{M_l^l} \cos (2\pi n \log |(r_m)_i^l|) \right)^2 + \frac{1}{M_l^l} \left( \sum_{m=1}^{M_l^l} \sin (2\pi n \log |(r_m)_i^l|) \right)^2 \right)^{1/2}.
\] (8)

Then, we extract one \( T \)-dimensional feature vector \( \mathbf{a}_i^l = [(\hat{a}_1)_i^l], \ldots, [(\hat{a}_T)_i^l] \in \mathbb{R}_+^T \) for the \( i \)-th training image from the \( l \)-th layer. We set \( T = 16 \) in experiments, as \( |(\hat{a}_n)_i^l| \) of larger \( n \) is too small for discrimination. Finally, we concatenate the feature vectors of all layers to form a long vector \( \hat{a}_i = [\hat{a}_1^1; \cdots; \hat{a}_T^L] \in \mathbb{R}_+^{TL} \), with \( L \) being the number of layers in \( f_\theta \). Consequently, we obtain a novel representation of training images, denoted as \( \hat{\mathcal{A}}_i = \{ (\hat{a}_i, \pm 1) \}_{i=1}^{N} \), where the label +1 or −1 is directly obtained from \( \mathcal{D}_{tr} \).

**Train an adversarial detector.** Finally, we train a binary SVM classifier based on \( \hat{\mathcal{A}}_i \). The trained SVM classifier will serve as the adversarial detector for the CNN model \( f_\theta \).

**Testing.** One novel testing example is firstly predicted as adversarial or not by the trained adversarial detector. If adversarial, then it is rejected; otherwise, it is fed into \( f_\theta \) to predict its category label.

**Remark 1.** Note that in the derivation of \( a_n \) (see (3)), the mean parameter of GGD is set to 0. In the experiments, we calculate the mean values of internal-layer responses of all networks for every image, and find that the mean parameters at most layers are close to zero, while the mean parameters at a few layers could be large. However, the mean value is subtracted from each response entry when we extract the MBF features in our experiments. Thus, the derived \( a_n \) is applicable to our task. Besides, according to Theorem 1, the estimation error of \( (\hat{a}_n)_i^l \) is inversely proportional to \( M_l^l \). It tells that the coefficient estimated from the larger-sized layer is more accurate than smaller-sized one. In many neural networks (e.g., visual geometry group (VGG)[11]), the response sizes of high layers get smaller, which means a less accurate estimation. However, we believe that the discrimination between adversarial and benign examples in higher layers is more evident than that in lower layers. There is a trade-off between estimation accuracy and discrimination. This is why we concatenate the estimated magnitudes of all layers to construct the novel representation.

## 5 Experiments

As a discriminative features-based method, we compare the proposed MBF with three methods within the same category and report the experimental results in Section 5.1. Besides, we noted that other branches of adversarial detection methods have recently merged, including generation-based, criterion-based, and statistical test-based methods. To ensure fair comparisons, we report the comparison results respectively in Sections 5.2–5.4 due to the inconsistency of experimental settings in their official implementations.

### 5.1 Comparison with discriminative feature-based detection

**Datasets and network architectures.** We conducted experiments on three databases for classification, including CIFAR-10[41], SVHN (Street View House Number Dataset)[42], and a subset of ImageNet[43]. In terms of CIFAR-10 and SVHN, we adopt the same settings with the compared method LID[44]. Specifically, a 17-layer ConvNet with max-pooling and dropout pre-trained on the training set of CIFAR-10 achieves the 93.57% accuracy on the testing set with 10000 benign images (denoted as model A in Tables 2 and 3); a 9-layer ConvNet with max-pooling and dropout pre-trained on the training set of SVHN achieves the 91.66% accuracy on the testing set with 26032 benign images denoted as model B in Tables 2 and 3. Then, we add a small noise drawn from the Gaussian distribution \( \mathcal{N}(0, \sigma^2) \) on each testing image, with \( \sigma \) being the similar level of the \( \ell_2 \) norm of each type of adversarial perturbation. If both the benign image and its noisy version can be correctly classified,
then it is picked out for the experiment of adversarial detection. Finally, we collected a total of 9,357 and 23,862 benign images from CIFAR-10 and SVHN, respectively. These images are randomly partitioned into the 80% training set and the 20% testing set, used for the detector training and testing. We also collect a subset from ImageNet, including 800 benign images of 8 classes (snowbird, spoonbill, bobtail, leonberg, hamster, proboscis mon-
key, cypripedium calceolus, and earthstar). The 100 images within each category contain 50 images from the ImageNet validation set and 50 images randomly selected from the ImageNet training set. We fine-tune the pre-trained checkpoints of both AlexNet and VGG to these 800 images to achieve 100% accuracy. Then, a total of 785 benign images are kept for detection, as their noisy version (by Gaussian noise) can be correctly classified by both the fine-tuned AlexNet and VGG models. These 785 images are then randomly partitioned into 480 training images and 305 testing images used for adversarial detection. For each database, as described in Section 4.1, one noisy and one adversarial image are generated for each benign image; then, all of these benign, noisy, and adversarial images are used for detection.

**Attack methods.** We adopt four popular adversarial attack methods to craft adversarial examples, including the basic iterative method (BIM)\(^2\), C&\&W-\&\&z (CW-L2\(^2\)), DeepFool\(^4\), and random projected gradient descent (R-PGD\(^4\)). We emphasize the detailed parameters of these attack strategies for reproducibility, which are BIM (\(\text{eps} = 0.016, \text{stepsize} = 0.002, 5, \text{iterations} = 10\)), CW-L2 (binary_search_steps = 5, confidence = 0.0, learning_rate = 0.005, max_iterations = 1000), DeepFool (max_steps = 100), R-PGD (\(\text{eps} = 0.016, \text{stepsize} = 0.0005, \text{iterations} = 40\)). In fact, \(\text{eps} = 0.016 \approx 4/255\) on the [0, 1] pixel scale, which is commonly adopted in the domain of adversarial attack and defense. All of the attack strategies are implemented by Foolbox\(^3\). We present some of the generated adversarial examples in Appendix A.2.

**Compared detection methods.** We compare three open-source effective adversarial detection methods, including KD-BU\(^4\), Mahalanobis distance\(^5\) (M-D)\(^3\), and LID\(^6\). To ensure a fair comparison, the SVM classifier is trained with all the compared methods, implemented by the `fitcsvm` function in Matlab. There are two important hyper-parameters in LID, i.e., the size of the mini-batch and the number of neighbors. CIFAR-10 and SVHN are respectively set as 100 and 20 respectively, as suggested in [14]; on ImageNet, since there are only 400 benign training images, they are respectively set as 50 and 20 in experiments. Moreover, we found that there are some unfair settings in the implementations of the compared methods. For example, KD-BU utilizes the extra 50,000 images of CIFAR-10 to compute the kernel density of each training and testing image; M-D also uses these extra images to compute the mean and co-variance of GMM. Since extra images of a similar distribution to the training images are often unavailable, extra images are not used for KD-BU and M-D to ensure fair comparisons in our experiments. Besides, LID utilizes other benign testing images as neighborhoods to extract features for each testing image. However, we argue that the testing images may be detected one by one in practical application. In our experiments, we instead used benign training images for neighboring each testing image.

**Three comparison cases and evaluation metrics.** We conduct experiments on three cases, including: 1) non-transfer, both training and testing adversarial examples are crafted by the same attack method; 2) attack-transfer, both training and testing adversarial examples are crafted by different attack methods; 3) data-transfer, both training and testing adversarial examples are crafted by the same attack method, but the data sources of training and testing benign examples are different. Two widely used metrics are used to evaluate the detection performance, including area under the receiver operating characteristics (AUROC) and detection accuracy, which is the diagonal summation of the confusion matrix, with the threshold on the posterior probability being 0.5. Higher values of both metrics indicate better performance.

**Experimental results.** Detection results in the non-transfer case are shown in Table 2. The proposed MBF method shows the best performance in all cases and is much superior to all the methods compared. LID performs second best in most cases. KD-BU gives good detection performance on CIFAR-10 and SVHN but performs very poorly on ImageNet. It tells us that KD-BU is very sensitive to different databases and networks. Note that we re-implement the M-D method to support for computing Mahalanobis scores on both convolution and fully-connected layers and scale the scores into [0, 1] for their better detection performances.

Detection results in the attack-transfer case evaluated by AUROC are presented in Table 3. MBF still shows much better performance among various transfer cases than all the compared methods, and the changes in detecting different attacks are very small. It verifies the robustness of MBF to different attacks.

We also conducted a data-transfer experiment on ImageNet. Specifically, we collect extra 365 images of 8 classes (the same as the classes used for the detection training) through searching the category names on Baidu\(^8\) and Facebook\(^9\) websites. These 365 benign images and their noisy images can be correctly predicted by both the fine-tuned AlexNet and VGG models. The results of these images are shown in Table 4. MBF still shows the best performance. Compared to the corresponding results in Table 2, the AUROC/accuracy scores of MBF on detecting different attacks change very slowly, verifying its robustness to different data sources.

---

\(^2\) https://pytorch.org/docs/robust/torchvision/models.html
\(^3\) https://foolbox.readthedocs.io/en/v1.8.0
\(^4\) https://github.com/rf reimann/detecting-adversarial-samples
\(^5\) https://github.com/pokapokap/deep_Mahalanobis_detector
\(^6\) https://github.com/xingjunm/lid_adversarial_subspace_detection
\(^8\) https://image.baidu.com
\(^9\) https://www.facebook.com
### 5.2 Comparison with generation based detection

Defense-GAN\cite{35} has been recently proposed to detect and restore adversarial examples. Differently from the aforementioned compared methods, which directly extract the discriminative features from the intermediate activation, Samangouei et al.\cite{35} trained a database-specific generative adversarial network (GAN)\cite{46} and took the norm of the pixel difference between the detected image and its restored version by GAN as the discriminative feature. The image is considered as adversarial if the norm of the difference is larger than a given threshold. In our experiments, we adopt the official code\cite{36} of Defense-GAN and train a generative model on the training set of MNIST\cite{47} by 133,500 iterations (note that their official implementation only supports three databases, including MNIST, Fashion-MNIST, and CelebA). In terms of conducting an adversarial attack, we train a classification model achieving the accuracy of 98.7% on the testing set of MNIST. We tested with four aforementioned attacks, namely BIM, CW-L2, DeepFool, and R-PGD. After removing images that are misclassified or fail to attack, we pick 6,000 images to train SVMs based on extracted MBF features, and leave nearly 4,000 images for testing. There are two key hyper-parameters in Defense-GAN, including the iteration number in each gradient descent (GD) run, and the total number of GD runs. We have tried multiple settings of these two hyper-parameters and report the best results. The AUROC scores on nearly 4,000 test images are shown in Table 5. Our MBF method exceeds Defense-GAN by a large margin in quantitative performance.

### 5.3 Comparison with criterion based detection

A recent paper named C1&C2t/u\cite{36} discusses the property of adversarial perturbations around benign examples and comes up with two paradoxical criteria for detection, that are low density of adversarial perturbations (C1) and close proximity to the decision boundary (C2t/u). It is assumed that the adversarial example cannot satisfy these two criteria simultaneously, while the benign example can do instead. To prove their correctness, the authors designed a novel loss function containing four terms to create the best-effort white-box adversarial examples, intending to evade the two detection criteria. In their implementation\cite{37}, they apply either the PGD-type or CW-type optimization to craft adversarial examples. During the detection process, an input example will be rejected as adversarial if at least one criterion is not satisfied. For a fair comparison, we employ our MBF method to detecting these adaptively crafted examples. We follow the official code of C1&C2t/u. The pre-trained classification model (VGG-19) achieves the 93.4% accuracy on 1,000 images picked from CIFAR-10. We randomly chose 600 from 934 correctly classified images as training images for adversarial detection and left the remaining 334 images for testing. As reported in [36], the learning rate (LR) in the creation of white-box adversarial examples significantly affects the detection performance of C1&C2t/u. As a result, we tested with different values of LR and report the corresponding results in Table 6. Our MBF method shows much better detection performance than C1&C2t/u in all cases. See their origin-

#### Table 4 Detection results evaluated by AUROC (%) in the data-transfer case. All detectors are tested on the out-of-sample set (including 965 images) beyond the ImageNet database. Best results are highlighted in bold.

| Detector | BIM | CW-L2 | DeepFool | R-PGD | BIM | CW-L2 | DeepFool | R-PGD |
|----------|-----|-------|----------|-------|-----|-------|----------|-------|
| KD-BU    | 65.3| 70.5  | 64.2     | 68.3  | 90.4| 86.7  | 85.1     | 86.6  |
| M-D\cite{30} | 82.7| 90.6  | 92.3     | 89.9  | 90.3| 99.0  | 99.2     | 93.5  |
| LID      | 70.0| 72.8  | 70.1     | 70.4  | 82.4| 85.2  | 87.6     | 84.6  |
| MBF      | **99.2**| **99.0**| **99.6**| **99.2**| **99.0**| **99.5**| **99.3**| **99.2**|

#### Table 5 Comparison with Defense-GAN on the detection results evaluated by AUROC (%) among nearly 4,000 images from the testing set of MNIST. The best results are highlighted in bold.

| Iteration number in each GD run | Number of GD runs | BIM   | CW-L2  | DeepFool | R-PGD |
|----------------------------------|-------------------|-------|--------|----------|-------|
| 200                              | 10                | 0.744 | 0.980  | 0.702    | 0.746 |
| 100                              | 10                | 0.689 | 0.973  | 0.650    | 0.690 |
| 100                              | 5                 | 0.664 | 0.969  | 0.628    | 0.666 |
| 100                              | 2                 | 0.612 | 0.959  | 0.586    | 0.614 |
| MBF                              | 1.000             | 0.991 | 1.000  | 1.000    | 1.000 |

\[10\] https://github.com/kabkabm/defensegan

\[11\] https://github.com/s-huu/TurningWeaknessIntoStrength
5.4 Comparison with statistical test-based detection

Recently, two methods established on statistical tests have been proposed for adversarial detection, which are “The odds are odd” (ODD)\cite{31} and “Joint statistical testing across DNN layers for anomalies” (JTLA)\cite{30}. As for the ODD method, the main assumption is that adversarial examples are less robust to non-malicious noise (e.g., uniform noise and Gaussian noise) than benign examples. After adding small noises, the variations in the differences between the logit value of the predicted category and all other categories are recorded, and the input image is rejected as adversarial if the expectation of variations is greater than a given threshold. The JTLA method obtains \(p\)-value scores per layer and combines them through the Fisher method. The final statistical test inside JTLA is multinomial. In our experiments, we adopt the official code of ODD\cite{36} and JTLA\cite{36} to conduct adversarial detection on the ResNet-34 classifier over the CIFAR-10 database against four attack methods implemented by Foolbox, including BIM, CW-L2, DeepFool, and R-PGD. We randomly choose 7,488 images from the testing set of CIFAR-10 as training samples for adversarial detection, and 1,257 images from the remaining part as testing samples. The AUROC score is utilized as the evaluation metric. As shown in Table 7, our proposed MBF method obtains consistent improvements compared to ODD and JTLA, especially for the BIM and DeepFool attacks.

5.5 Adaptive attack against adversarial detection

Similarly to [28] and [14], we also present an adaptive white-box attack, the goal of which is not only to fool the classifier, but also to evade the adversarial detector. It measures how easy it is to bypass the detector. Specifically, by combining the CW-L2 attack\cite{32} and the adaptive detector, the adaptive attack is formulated as follows:

\[
\arg\min_{\mathbf{x}_{\text{adv}} \in [0, 1]} \|\mathbf{x} - \mathbf{x}_{\text{adv}}\|_2^2 + \alpha \left(-l(f_{\theta}(\mathbf{x}_{\text{adv}}); y) + \|D(f_{\theta}(\mathbf{x}_{\text{adv}})) - D(f_{\theta}(\mathbf{x}))\|_2\right)
\]

where \(\mathbf{x}\) denotes the clean input, and \(y\) indicates the ground-truth label. \(l(\mathbf{x}_{\text{adv}}, y)\) is the hinge loss. \(D(f_{\theta}(\mathbf{x}_{\text{adv}}))\) represents the extracted features by the detector \(D\) from \(f_{\theta}(\mathbf{x}_{\text{adv}})\). The minimization of the last term \(\|D(f_{\theta}(\mathbf{x}_{\text{adv}})) - D(f_{\theta}(\mathbf{x}))\|_2\) encourages the discriminative features of adversarial and benign examples to be close, such that the detector could be evaded. With the same setting in [14], the trade-off parameter \(\alpha\) is determined by a binary search within the range \([10^{-3}, 10^6]\). To make the comparison, we specify the detector \(D\) by LID and the proposed MBF in our experiment since LID performs much better than KD+BU and M-D in the above experiments. For clarity, the discriminative features are only extracted from the pre-softmax layer for both LID and MBF. We note that LID assumes that the extracted feature of benign examples \(D(f_{\theta}(\mathbf{x}))\) is very close to zero, so we adopt the same setting in [14] and rewrite the last term in (9) as \(\|D(f_{\theta}(\mathbf{x}_{\text{adv}}))\|_2\) when conducting the adaptive attack against LID. We name the two adaptive attacks CW-LID and CW-MBF, respectively.

We conducted these adaptive attacks on both CIFAR-
versarial example as $p_{\text{adv}}$ and that of a benign example as $p_{\text{ben}}$. The distribution of GGD is denoted as $P_{\text{GGD}}$. Then, we conduct two one-sample KS tests, including:

1) **H1.1.** The test of adversarial examples: $H_0$: $p_{\text{adv}} \sim P_{\text{GGD-adv}}$; $H_1$: $p_{\text{adv}} \neq P_{\text{GGD-adv}}$.

2) **H1.2.** The test of benign examples: $H_0$: $p_{\text{ben}} \sim P_{\text{GGD-ben}}$; $H_1$: $p_{\text{ben}} \neq P_{\text{GGD-ben}}$.

In H1.1, the reference distribution $P_{\text{GGD-adv}}$ is first estimated from $p_{\text{adv}}$, using the estimated method proposed in [48]. To alleviate the uncertainty of the estimation, we draw 1000 samples from the estimated $P_{\text{GGD-adv}}$. Then, we conduct the two-sample KS test between $p_{\text{adv}}$ and these 1000 samples, respectively. The average $p$-value over these 1000 tests is recorded. Finally, the mean of the average $p$-values over the whole database is reported. H1.2 is conducted similarly. The results tested on ImageNet-VGG-16 are shown in Table 9. Due to the space limit, the results on other databases and models will be presented in Appendix A.3. In all cases, the $p$-values are larger than the significance level of 0.05. Hence, we can conclude that the posterior vectors of both adversarial and benign examples follow the GGD. However, note that the parameters of their corresponding GGD are different, which will be further verified as below.

**Hypothesis test 2.** Here, we verify whether the extracted MBF features of adversarial and benign examples follow the same empirical distribution. We denote the MBF feature vector of an adversarial example as $m_{\text{adv}}$, and the corresponding empirical distribution is denoted as $P_{\text{adv-MBF}}$. Similarly, we define $m_{\text{ben}}$ and $P_{\text{ben-MBF}}$ for benign examples. Then, we conducted the following four two-sample KS tests:

1) **H2.1.** The test between adversarial and benign examples: $H_0$: $P_{\text{adv-MBF}} = P_{\text{ben-MBF}}$; $H_1$: $P_{\text{adv-MBF}} \neq P_{\text{ben-MBF}}$.

2) **H2.2.** The test between adversarial examples crafted from different attack methods: $H_0$: $\hat{P}_{\text{adv-MBF}}^{\text{attack-1}} = P_{\text{adv-MBF}}$; $H_1$: $\hat{P}_{\text{adv-MBF}}^{\text{attack-1}} \neq P_{\text{adv-MBF}}$.

3) **H2.3.** The test between adversarial examples from different data sources (i.e., train, test and out-of-sample set): $H_0$: $P_{\text{adv-MBF}} = P_{\text{adv-MBF}}^{\text{source-1}} = P_{\text{adv-MBF}}^{\text{source-2}} = P_{\text{adv-MBF}}^{\text{source-2}}$; $H_1$: $P_{\text{adv-MBF}} \neq P_{\text{adv-MBF}}^{\text{source-1}} \neq P_{\text{adv-MBF}}^{\text{source-2}}$.

4) **H2.4.** The test between benign examples from different data sources (i.e., train, test and out-of-sample set): $H_0$: $P_{\text{ben-MBF}} = P_{\text{ben-MBF}}^{\text{source-1}} = P_{\text{ben-MBF}}^{\text{source-2}} = P_{\text{ben-MBF}}^{\text{source-2}}$; $H_1$: $P_{\text{ben-MBF}} \neq P_{\text{ben-MBF}}^{\text{source-1}} \neq P_{\text{ben-MBF}}^{\text{source-2}}$.

In the above four two-sample KS tests, we tested the 16-dimensional MBF features extracted from the softmax layer. Since the implementation of scipy.stats.kstest cannot compare two vectors, we compare the feature of each dimension separately and then report the average $p$-value over all the dimensions. Specifically, when comparing two sets of samples, we first concatenate the feature of each dimension across all samples in the same set, leading to 16 long vectors for each set. Then, each pair of two long vectors corresponding to the same dimension from two sets is compared by the KS test. The average $p$-value for all 16 dimensions is reported. The $p$-values of

---

14 https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.kstest.html
Table 8  Comparison results of adaptive attacks with the LID detector. “Failure of generation” denotes the failure rate (%) of generating an adversarial example; “Failure of dodging” denotes the rate of being detected by the adversarial detector for those successfully generated examples, and “Failure of attack” denotes the attack failure rate, which indicates the union of the above two situations, and a higher rate indicates that it is harder to break the detector. See context for more details.

| Database  | Attack | Failure of generation | Failure of dodging | Failure of attack |
|-----------|--------|-----------------------|--------------------|------------------|
| CIFAR-10  | original CW-L2 | 0.0 | -- | 0.0 |
|           | CW-LID  | 90.2 | 80.6 | 98.1 |
|           | CW-MBF  | 0.0 | 98.9 | 98.9 |
| SVHN      | original CW-L2 | 0.0 | -- | 0.0 |
|           | CW-LID  | 84.9 | 79.5 | 96.9 |
|           | CW-MBF  | 0.0 | 98.5 | 98.5 |

H2.1 are shown in Table 10 (see the column “(clean, BIM)”). The $p$-values on both the training and testing sets are 0. Thus, hypothesis $H_0$ is rejected, i.e., the MBF features of adversarial and benign examples follow different distributions. The $p$-values of H2.2 are shown in Table 10. We pick two groups of attack methods, i.e., (BIM, DeepFool) and (CW-L2, R-PGD). The $p$-values of (BIM, DeepFool) are 0, while the $p$-values of (CW-L2, R-PGD) are larger than 0.05. It demonstrates that the distributions of adversarial examples crafted from different attack methods can be different. The $p$-values of H2.3 are shown in Table 11. The $p$-values of all types of adversarial examples exceed 0.05. Thus, the MBF features of adversarial examples from different data sources follow the same distribution. The $p$-values of H2.4 are shown in Table 11. Only the $p$-value of “(training set, out-of-sample set)” of clean examples is slightly lower than 0.05, while the values of other cases exceed 0.05. Thus, in most cases, the MBF features of benign examples from different data sources follow the same distribution.

From the above analysis, we obtain the following conclusions: 1) The extracted MBF features of adversarial and benign examples follow different empirical distributions. It explains why MBF features are effective for detecting adversarial and benign examples. 2) The extracted MBF features of adversarial/benign examples from the training, testing, and out-of-sample sets follow the same empirical distribution. Although the extracted MBF features of adversarial examples crafted from different attack methods may not follow the same empirical distribution, the significant difference between benign and different adversarial distributions can still lead to the good detection performance in the attack-transfer case. It explains why the MBF features are robust across different attack methods and data sources. Moreover, we visualize the statistics of each dimension of MBF features, i.e., mean and standard deviation, as shown in Fig. 1. Specifically, the $x$-axis indicates the indices of the Benford-Fourier coefficient in each feature vector $\mathbf{a}^I_n$, corresponding to the notation $n$ in (8). The $y$-axis indicates the value of MBF, which can be calculated according to (8). These visualizations also support the above conclusions. Due to the space limit, more KS tests and visualizations on different databases and networks will be presented in Appendix A.3.

6 Conclusions

This work has proposed a novel adversarial detection method, dubbed MBF. The assumption behind this is that the internal responses of the classification network of
both adversarial and benign examples follow the generalized Gaussian distribution, but with different shape factors. The magnitude of the Benford-Fourier coefficient is a function w.r.t. the shape factor and can be easily estimated based on responses. Thus, it can serve as the discriminative feature between adversarial and benign examples. The extensive experiments conducted on several databases, as well as the empirical analysis via the KS test, demonstrate the superior effectiveness and robustness against different attacks and different data sources of the proposed MBF method, over state-of-the-art detection methods.

Acknowledgements

This work was supported by Natural Science Foundation of China (No. 62076213), Shenzhen Science and Technology Program, China (No. RCYX20210609103057050), the university development fund of The Chinese University of Hong Kong, Shenzhen, China (No. 01001810), and Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen, China.

Appendix A

A.1 Proof of Theorem 1

Assume that all variables \( \{T_m\}_{m=1, \ldots, M} \) are independent and identically distributed. Applying the central limit theorem[^50] to the real and imaginary parts of \( Y \), we can obtain that both parts asymptotically follow the Gaussian distribution.

\[
Y = \frac{1}{M} \sum_{m=1}^{M} T_m \sim \mathcal{N} \left( E(T_1), \frac{1}{M} D(T_1) \right) \tag{A1}
\]

where

\[
E(T_1) = E \left( e^{-j2\pi n \log|X_1|} \right) = \\
\int_{-\infty}^{+\infty} \mathcal{P}_{X_1}(x) \times e^{-j2\pi n \log|x|} \, dx = \alpha_n \tag{A2}
\]

\[
D(T_1) = E(|T_1|^2) - |E(T_1)|^2 = \\
E \left( |e^{-j2\pi n \log|X_1|}|^2 \right) - |\alpha_n|^2 = 1 - |\alpha_n|^2. \tag{A3}
\]

The PDF of \( Y \) can be rewritten as follows:

\[
Y \sim \mathcal{N}(\alpha_n, \frac{1 - |\alpha_n|^2}{M}). \tag{A4}
\]

Thus, we obtain

\[
\mathcal{E} \sim \mathcal{N}(0, \frac{1 - |\alpha_n|^2}{M}). \tag{A5}
\]

Besides, the pseudo variance[^50] of \( T_1 \) is

\[
\mathcal{J}_{\mathcal{T}_1, \mathcal{T}_1} = E(T_1^2) - E(T_1)^2 = \\
E(\left| e^{-j2\pi n \log|X_1|} \right|^2) - \alpha_n^2 = a_{2n} - \alpha_n^2. \tag{A6}
\]

Correspondingly,

\[
\mathcal{J}_{\mathcal{E}, \mathcal{E}} = \frac{1}{M} \mathcal{J}_{\mathcal{T}_1, \mathcal{T}_1} = \frac{a_{2n} - \alpha_n^2}{M}. \tag{A7}
\]

Since \( a_{2n} - \alpha_n^2 \) is bounded, we have \( \lim_{M \to \infty} \mathcal{J}_{\mathcal{E}, \mathcal{E}} = 0 \). Thus, we find that the random variable \( \mathcal{E} \) follows a circularly-symmetric complex Gaussian distribution because the sufficient and necessary condition is that the mean value and pseudo variance equal zero[^50]. It implies that both the real and imaginary parts of \( \mathcal{E} \) follow the same
Gaussian distribution and are independent. Thus, the magnitude of this complex random variable follows the Rayleigh distribution [39], and the probability density function of $|\mathcal{E}|$ can be formulated as

$$p_{|\mathcal{E}|}(r) = \frac{r}{s^2} e^{-r^2/2s^2}$$  \hspace{1cm} (A8)

where $s$ is the scale parameter. Knowing the properties of the Rayleigh distribution [39], we have the following:

$$s^2 = \frac{1}{2} D(\mathcal{E}) = \frac{1 - |a_n|^2}{2M}.$$  \hspace{1cm} (A9)

Utilizing the fact that $|a_n|^2$ is close to zero when $n$ is a modest number, we obtain that $D(\mathcal{E}) \approx \frac{1}{M}$ leading to $s^2 = \frac{1}{2M}$. Then, we obtain

$$E(|\mathcal{E}|) = \frac{1}{2} \sqrt{\frac{\pi}{M}}, \quad D(|\mathcal{E}|) = \frac{4 - \pi}{4M}.$$  \hspace{1cm} (A10)

It is easy to observe that both $E(|\mathcal{E}|)$ and $D(|\mathcal{E}|)$ are close to zero when the number of samples $M$ increases. It implies that the estimation error $\varepsilon_n$ gets closer to 0 as $M$ increases.

A.2 Display on adversarial examples

In Fig. A1, we display a few adversarial examples and noises generated by four different attacks, as mentioned in Section 5.1.

A.3 Additional empirical analysis

Here, we present additional empirical analysis on more databases and networks, as shown in Tables 12–14. The $p$-values in most cases also support the conclusions. We also present more visualizations in Figs. A2–A4. These visualizations also demonstrate the difference in MBF features between adversarial and benign examples.

![Fig. A1](image-url) Visualization of adversarial examples and noises generated by four attacks. (The perturbations are multiplied by 100 for better views.)

| Table 12 | The $p$-values of the KS hypothesis test among clean samples, noisy samples, and adversarial samples crafted by four methods, respectively |
|----------|---------------------------------------------------------------|
| Dataset  | Clean | Noisy | BIM | CW-L2 | DeepFool | R-PGD |
| CIFAR-10 | Training set | 0.067 | 0.067 | 0.129 | 0.121 | 0.102 | 0.125 |
|          | Testing set | 0.067 | 0.067 | 0.132 | 0.124 | 0.103 | 0.128 |
| SVHN     | Training set | 0.070 | 0.070 | 0.191 | 0.143 | 0.161 | 0.181 |
|          | Testing set | 0.071 | 0.071 | 0.187 | 0.142 | 0.160 | 0.178 |
| ImageNet-AlexNet | Training set | 0.179 | 0.179 | 0.342 | 0.322 | 0.330 | 0.343 |
|          | Testing set | 0.181 | 0.181 | 0.350 | 0.324 | 0.332 | 0.348 |
| ImageNet-VGG-16 | Training set | 0.177 | 0.177 | 0.323 | 0.314 | 0.329 | 0.322 |
|          | Testing set | 0.177 | 0.177 | 0.318 | 0.314 | 0.327 | 0.319 |
Table 13  The p-values of the two-sample KS test among MBF coefficients of different types of examples. We cannot reject the hypothesis that the posterior vectors of both clean and non-malicious noisy images follow the same probability distribution, while clean and adversarial posterior vectors follow a different probability distribution.

|                  | (Clean, noisy) | (Clean, BIM) | (BIM, DeepFool) | (CW-L2, R-PGD) |
|------------------|----------------|--------------|-----------------|----------------|
| CIFAR-10         |                |              |                 |                |
| Training set     | 0.758          | 0.000        | 0.000           | 0.000          |
| Testing set      | 0.465          | 0.000        | 0.000           | 0.000          |
| SVHN             |                |              |                 |                |
| Training set     | 0.840          | 0.000        | 0.000           | 0.000          |
| Testing set      | 0.389          | 0.000        | 0.000           | 0.000          |
| ImageNet-AlexNet |                |              |                 |                |
| Training set     | 0.987          | 0.000        | 0.008           | 0.140          |
| Testing set      | 0.998          | 0.000        | 0.000           | 0.117          |
| ImageNet-VGG-16  |                |              |                 |                |
| Training set     | 0.999          | 0.000        | 0.006           | 0.652          |
| Testing set      | 1.000          | 0.000        | 0.000           | 0.527          |

Table 14  The p-values of the KS test among MBF coefficients from different data sources. Since all the p-values exceed 5%, we cannot reject the hypothesis that the posterior vectors of examples from both training and testing databases follow the same probability distribution.

|                  | Clean | Noisy | BIM  | CW-L2 | DeepFool | R-PGD |
|------------------|-------|-------|------|-------|----------|-------|
| CIFAR-10         |       |       |      |       |          |       |
| (Training set, testing set) | 0.620 | 0.836 | 0.566 | 0.628 | 0.556    | 0.370 |
| SVHN             |       |       |      |       |          |       |
| (Training set, testing set) | 0.901 | 0.848 | 0.503 | 0.147 | 0.491    | 0.416 |
| ImageNet-AlexNet |       |       |      |       |          |       |
| (Training set, testing set) | 0.149 | 0.145 | 0.406 | 0.596 | 0.402    | 0.481 |
| (Training set, out-of-sample) | 0.000 | 0.000 | 0.190 | 0.099 | 0.201    | 0.225 |
| ImageNet-VGG-16  |       |       |      |       |          |       |
| (Training set, testing set) | 0.828 | 0.848 | 0.613 | 0.834 | 0.240    | 0.555 |
| (Training set, out-of-sample) | 0.048 | 0.050 | 0.178 | 0.398 | 0.726    | 0.263 |

Fig. A2  Statistics (mean ± standard deviation) of the MBF coefficients in the training (top row) and testing (bottom row) sets of CIFAR-10

Fig. A3  Statistics (mean ± standard deviation) of the MBF coefficients in the training (top row) and testing (bottom row) sets of SVHN
Fig. A4 Statistics (mean ± standard deviation) of the MBF coefficients in the training (top row) and testing (median row) and out-of-sample (bottom row) sets of ImageNet-AlexNet.

Declarations of conflict of interest

The authors declared that they have no conflicts of interest to this work.

References

[1] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, USA, 2015.

[2] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, USA, pp. 1097–1105, 2012.

[3] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger. Densely connected convolutional networks. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Honolulu, USA, pp. 2261–2269, 2017. DOI: 10.1109/CVPR.2017.243.

[4] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[5] X. O. Tang, Z. F. Li. Video based face recognition using multiple classifiers. In Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition, IEEE, Seoul, Republic of Korea, pp. 345–349, 2004. DOI: 10.1109/AFGR.2004.1301555.

[6] D. H. Gong, Z. F. Li, J. Z. Liu, Y. Qiao. Multi-feature canonical correlation analysis for face photo-sketch image retrieval. In Proceedings of the 21st ACM International Conference on Multimedia, ACM, Barcelona, Spain, pp. 617–620, 2013. DOI: 10.1145/2502808.2502162.

[7] Z. F. Li, D. H. Gong, Y. Qiao, D. C. Tao. Common feature discriminant analysis for matching infrared face images to optical face images. IEEE Transactions on Image Processing, vol. 23, no. 6, pp. 2436–2445, 2014. DOI: 10.1109/TIP.2014.2315920.

[8] Z. Y. Deng, X. J. Peng, Z. F. Li, Y. Qiao. Mutual component convolutional neural networks for heterogeneous face recognition. IEEE Transactions on Image Processing, vol. 28, no. 6, pp. 3102–3114, 2019. DOI: 10.1109/TIP.2019.2894272.

[9] H. B. Qiu, D. H. Gong, Z. F. Li, W. Liu, D. C. Tao. End2end occluded face recognition by masking corrupted features. IEEE Transactions on Pattern Analysis and Machine Intelligence, to be published. DOI: 10.1109/TPAMI.2021.3098962.

[10] X. L. Yang, X. H. Jia, D. H. Gong, D. M. Yan, Z. F. Li, W. Liu. LARNets: Lie algebra residual network for face recognition. In Proceedings of the 38th International Conference on Machine Learning, pp. 11738–11750, 2021.

[11] S. Q. Ren, K. M. He, R. Girshick, J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, pp. 91–99, 2015.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg. SSD: Single shot MultiBox detector. In Proceedings of the 14th European Conference on Computer Vision. Springer, Amsterdam, The Netherlands, pp. 21–37, 2016. DOI: 10.1007/978-3-319-46448-0_2.

[13] R. Feinman, R. R. Curtin, S. Shintre, A. B. Gardner. Detecting adversarial samples from artifacts. [Online]. Available: https://arxiv.org/abs/1703.00410, 2017.

[14] X. J. Ma, B. Li, Y. S. Wang, S. M. Erfani, S. N. R. Wijesickrema, G. Schoenebeck, D. Song, M. E. Houle, J. Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, Canada, 2018.

[15] M. K. Varanasi, B. Aazhang. Parametric generalized Gaussian density estimation. The Journal of the Acoustical Society of America, vol. 86, no. 4, pp. 1404–1415, 1989. DOI: 10.1121/1.396700.

[16] C. Pasquini, F. Pérez-González, G. Boato. A Benford-Fourier JPEG compression detector. In Proceedings of IEEE International Conference on Image Processing, IEEE, Paris, France, pp. 5322–5326, 2014. DOI: 10.1109/ICIP.2014.7026077.

[17] V. N. Vapnik. The Nature of Statistical Learning Theory,
New York, USA: Springer, 1999.

[18] F. J. Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951. DOI: 10.1080/01621459.1951.10500769.

[19] X. Li, F. Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5764–5772, 2017.

[20] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Journal and Magazine of Science*, vol. 2, no. 11, pp. 559–572, 1901. DOI: 10.1080/14786440109462720.

[21] J. J. Lu, T. Issaranon, D. Forsyth. SafetyNet: Detecting and rejecting adversarial examples robustly. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Venice, Italy, pp. 446–454, 2017. DOI: 10.1109/ICCV.2017.56.

[22] J. H. Metzen, T. Genewein, V. Fischer, B. Bischoff. On detecting adversarial perturbations. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[23] I. J. Goodfellow, J. Shlens, C. Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA, 2014.

[24] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami. The Limitations of Deep Learning in Adversarial Settings. In *Proceedings of IEEE European Symposium on Security and Privacy*, IEEE, Saarbrucken, Germany, pp. 372–387, 2016. DOI: 10.1109/EuroSP.2016.36.

[25] N. Carlini, D. Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of IEEE Symposium on Security and Privacy (SP)*, IEEE, San Jose, USA, pp. 39–57, 2017. DOI: 10.1109/SP.2017.49.

[26] K. Grose, P. Manoharan, N. Papernot, M. Backes, P. McDaniel. On the (statistical) detection of adversarial examples. [Online]. Available: https://arxiv.org/abs/1702.06280, 2017.

[27] R. Z. Gao, F. Liu, J. F. Zhang, B. Han, T. L. Liu, G. Niu, M. Sugiyama. Maximum mean discrepancy test is aware of adversarial attacks. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 3564–3575, 2021.

[28] N. Carlini, D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, ACM, Texas, USA, pp. 3–4, 17. DOI: 10.1145/3128572.3140444.

[29] Z. H. Zheng, P. Y. Hong. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 7924–7933, 2018.

[30] K. Lee, K. Lee, H. Lee, J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 7167–7177, 2018.

[31] K. Roth, Y. Kilcher, T. Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp. 5498–5507, 2019.

[32] J. Raghuwar, V. Chandrasekaran, S. Jha, S. Banerjee. A general framework for detecting anomalous inputs to DNN classifiers. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8764–8775, 2021.

[33] D. Hendrycks, K. Gimpel. Early methods for detecting adversarial images. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[34] T. Y. Pang, C. Du, Y. P. Dong, J. Zhu. Towards robust detection of adversarial examples. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 4584–4594, 2018.

[35] P. Samangouei, M. Kabbab, R. Challappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, 2018.

[36] S. Y. Hu, T. Yu, C. Guo, W. L. Chao, K. Q. Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, Canada, pp. 1633–1644, 2019.

[37] F. Pérez-González, G. L. Heileen, C. T. Abdallah. Benford’s Law in image processing. In *Proceedings of 2007 IEEE International Conference on Image Processing*, IEEE, San Antonio, USA, pp. I-405–I-408, 2007. DOI: 10.1109/ICIP.2007.4378977.

[38] I. S. Gradhsteyn, I. M. Ryzhik. *Table of Integrals, Series, and Products*, Cambridge, UK: Academic Press, 2014.

[39] A. Papoulis. Probability, Random Variables, and Stochastic Processes, New York, USA: McGraw-Hill, 1965.

[40] A. Kurakin, I. J. Goodfellow, S. Bengio. Adversarial examples in the physical world. *Artificial Intelligence Safety and Security*, R. V. Yampolskiy. Ed., New York, USA: Chapman and Hall/CRC, pp. 99–112, 2018.

[41] A. Krizhevsky, V. Nair, G. Hinton. Cifar-10 (Canadian institute for advanced research), [Online]. Available: https://academic torrents.com/details/4e3ba7e7f3737d414c12fbb1ef6431ead22d7a.

[42] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Proceedings of NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, Granada, Canada, 2011.

[43] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Miami, USA, pp. 248–255, 2009. DOI: 10.1109/CVPR.2009.5206848.

[44] S. M. Moosavi-Dezfooli, A. Fawzi, P. Frossard. DeepFool: A simple and accurate method to fool deep neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 2574–2582, 2016. DOI: 10.1109/CVPR.2016.282.

[45] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, 2018.

[46] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath. Generative adversarial net-
works: An overview. IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53–65, 2018. DOI: 10.1109/MSP.2017.2765202.

[47] L. Deng. The MNIST database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141–142, 2012. DOI: 10.1109/MSP.2012.2211477.

[48] N. E. Lasmar, Y. Stitou, Y. Berthoumieu. Multiscale skewed heavy tailed model for texture analysis. In Proceedings of the 16th IEEE International Conference on Image Processing, IEEE, Cairo, Egypt, pp. 2281–2284, 2009. DOI: 10.1109/ICIP.2009.5414404.

[49] M. Rosenblatt. A central limit theorem and a strong mixing condition. Proceedings of the National Academy of Sciences of the United States of America, vol. 42, no. 1, pp. 43–47, 1956. DOI: 10.1073/pnas.42.1.43.

[50] N. R. Goodman. Statistical analysis based on a certain multivariate complex Gaussian distribution (An introduction). The Annals of Mathematical Statistics, vol. 34, no. 1, pp. 152–177, 1963. DOI: 10.1214/aoms/1177704250.

Cheng-Cheng Ma received the B.Sc. degree in automation from Northwestern Polytechnical University, China in 2017. Currently, he is a Ph.D. degree candidate of Institute of Automation, Chinese Academy of Sciences, China. His research interests include adversarial learning and machine learning.

E-mail: machengcheng2017@ia.ac.cn
ORCID ID: 0000-0002-0502-3960

Bao-Yuan Wu received the Ph.D. degree in pattern recognition and intelligent systems from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China in 2014. From November 2016 to August 2020, he was a senior and principal researcher at AI Lab, Tencent Inc., China. Currently, he is an associate professor of School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China. He is also the director of the Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data (SBRID), China. He has published more than 50 top-tier conference and journal papers, including TPAMI, IJCV, NeurIPS, CVPR, ICCV, ECCV, ICLR, AAAI, and one paper was selected as the Best Paper Finalist of CVPR 2019. He is currently serving as an Associate Editor of Neurocomputing, Area Chair of ICLR 2022, AAAI 2022 and ICIG 2021, Senior Program Committee Member of AAAI 2021 and IJCAI 2020/2021. His research interests include AI security and privacy, machine learning, computer vision and optimization.

E-mail: wubaoyuan@cuhk.edu.cn (Corresponding author)
ORCID ID: 0000-0003-2183-5990

Yan-Bo Fan received the B.Sc. degree in computer science and technology from Hunan University, China in 2013, and the Ph.D. degree in pattern recognition and intelligent systems from Institute of Automation, Chinese Academy of Sciences, China in 2018. He is currently a senior researcher at AI Lab, Tencent Inc., China. His research interests include computer vision and machine learning.

E-mail: fanyanbo0124@gmail.com
ORCID ID: 0000-0002-8530-485X

Yong Zhang received the Ph.D. degree in pattern recognition and intelligent systems from Institute of Automation, Chinese Academy of Sciences, China in 2018. From 2015 to 2017, he was a visiting scholar with the Rensselaer Polytechnic Institute, USA. He is currently with AI Lab, Tencent Inc., China. His research interests include computer vision and machine learning.

E-mail: zhangyong201303@gmail.com
ORCID ID: 0000-0003-0066-3448

Zhi-Feng Li received the Ph.D. degree from The Chinese University of Hong Kong, China in 2006. After that, he was a postdoctoral fellow at The Chinese University of Hong Kong, China, and Michigan State University, USA for several years. He is currently a top-tier principal research scientist with Tencent, China. Before joining Tencent, he was a full professor with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. He was one of the 2020 Most Cited Chinese Researchers (Elsevier-Scopus) in computer science and technology. He is currently serving on the editorial boards of Neurocomputing, IEEE Transactions on Circuits and Systems for Video Technology, and Pattern Recognition. He is a Fellow of British Computer Society (FBCS).

His research interests include deep learning, computer vision and pattern recognition, and face detection and recognition.

E-mail: michaelzfli@tencent.com
ORCID ID: 0000-0001-5902-5067