9-2020

Cats are not fish: Deep learning testing calls for out-of-distribution awareness

David BEREND

Xiaofei XIE
Singapore Management University, xfxie@smu.edu.sg

Lei MA

Lingjun ZHOU

Yang LIU

See next page for additional authors

Citation

BEREND, David; XIE, Xiaofei; MA, Lei; ZHOU, Lingjun; LIU, Yang; XU, Chi; and ZHAO, Jianjun. Cats are not fish: Deep learning testing calls for out-of-distribution awareness. (2020). *Proceedings of the 35th IEEE/ ACM International Conference on Automated Software Engineering (ASE): Virtual, 2020 September 21-25*. 1041-1052.

Available at: https://ink.library.smu.edu.sg/sis_research/7076

Author

David BEREND, Xiaofei XIE, Lei MA, Lingjun ZHOU, Yang LIU, Chi XU, and Jianjun ZHAO

Cats Are Not Fish: Deep Learning Testing Calls for Out-Of-Distribution Awareness

David Berend
Nanyang Technological University, Singapore

Xiaofei Xie*
Nanyang Technological University, Singapore

Lei Ma
Kyushu University, Japan

Lingjun Zhou
Tianjin University, China

Yang Liu
Nanyang Technological University, Zhejiang Sci-Tech University, China

Chi Xu
Singapore Institute of Manufacturing Technology, A*Star

Jianjun Zhao
Kyushu University, Japan

ABSTRACT

As Deep Learning (DL) is continuously adopted in many industrial applications, its quality and reliability start to raise concerns. Similar to the traditional software development process, testing the DL software to uncover its defects at an early stage is an effective way to reduce risks after deployment. According to the fundamental assumption of deep learning, the DL software does not provide statistical guarantee and has limited capability in handling data that falls outside of its learned distribution, i.e., out-of-distribution (OOD) data. Although recent progress has been made in designing novel testing techniques for DL software, which can detect thousands of errors, the current state-of-the-art DL testing techniques usually do not take the distribution of generated test data into consideration. It is therefore hard to judge whether the ”identified errors” are indeed meaningful errors to the DL application (i.e., due to quality issues of the model) or outliers that cannot be handled by the current model (i.e., due to the lack of training data). Tofill this gap, we take thefi rst step and conduct a large scale empirical study, with a total of 451 experiment configurations, 42 deep neural networks (DNNs) and 1.2 million test data instances, to investigate and characterize the impact of OOD-awareness on DL testing. We further analyze the consequences when DL systems go into production by evaluating the effectiveness of adversarial retraining with distribution-aware errors. The results confirm that introducing data distribution awareness in both testing and enhancement phases outperforms distribution unaware retraining by up to 21.5%.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Software and its engineering** → *Software testing and debugging*.

KEYWORDS

Deep learning testing, quality assurance, out of distribution

ACM Reference Format:
David Berend, Xiaofei Xie, Lei Ma, Lingjun Zhou, Yang Liu, Chi Xu, and Jianjun Zhao. 2020. Cats Are Not Fish: Deep Learning Testing Calls for Out-Of-Distribution Awareness. In *35th IEEE/ACM International Conference on Automated Software Engineering (ASE '20), September 21–25, 2020, Virtual Event, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3324884.3416609

1 INTRODUCTION

Recently, deep learning (DL) achieved tremendous success and is continuously adopted in many applications, such as image classification [3], speech recognition [47], natural language processing [6], video gaming [7], etc. Service operations are supported by simple administrative tasks outsourced to deep learning software while manufacturing further accelerates through automation via intelligent robotics [5]. Furthermore, an increasing demand for automation and intelligent support is also witnessed in some safety-critical areas, such as autonomous driving [45, 46] and healthcare [1].

As more and more DL software is applied to diverse application domains, impacting our daily activities and lives, its quality and reliability quickly raise lots of concerns, especially in the context of safety-critical and security-critical scenarios. We have already witnessed the accidents and negative social impacts that were caused by quality issues of DL software, e.g., Tesla/Uber accidents [49, 50], wrong diagnosis in healthcare, e.g. cancer or diabetes [1]. Therefore, systematic testing to uncover the incorrect behavior and understand the capability of the DL software is pressing and important, which not only provides confidence in its quality but also reduces the risks after deployment.

However, different from traditional software whose decision logic is mostly programmed by the developer, deep learning adopts a data-driven programming paradigm. In particular, the major tasks of a DL developer are preparing the training data, labeling the data, programming the architecture of the deep neural network (DNN), and specifying the training configuration. All the decision logic is automatically learned during the runtime training phase and encoded in the obtained DNN (e.g., by weights, bias, and their combinations). Due to the differences of programming paradigm, the logic encoding format, and the tasks that a DNN is often developed for (e.g., image recognition), testing techniques for traditional software cannot be directly applied and new testing techniques are needed for DNNs.

While some recent progress has been made in proposing novel testing criteria [17, 25, 33, 35] and test generation techniques for quality assurance of DNNs [8, 33, 35, 43, 48, 55, 58], it still lacks interpretation and understanding on the *detected errors* by such techniques and their impact. For example, it is not clear whether errors are indeed caused by missing training data or insufficient training, etc. The fundamental assumption of deep learning is that

[*] Xiaofei Xie (xfxie@ntu.edu.sg) is the corresponding author.

the training data follows some *distribution*, based on which the learning algorithms train the DNN to obtain its decision logic and are able to handle future data that follow the similar distribution.

If the new unseen input data has a similar distribution as the training data, deep learning provides some statistical guarantee on its prediction correctness in terms of accuracy. However, if the new input data does not follow the training data (i.e., out-of-distribution (OOD)), deep learning does not provide statistical guarantee on its prediction. For example, if a DNN is only trained on cat and dog data for binary classification, given an input data offi sh, the DNN can still produce a prediction result. However, this input data does not follow the distribution of cat and dog data. Hence, handling the fish data goes beyond the capability of this DNN and should not be considered as valid input.

Intuitively, erroneous inputs that follow the distribution of training data may reveal the real weakness of the DNN since the DNN is expected to handle such data. On the other hand, input errors that are considered out-of-distribution may either inherit new information benefitting generalization as well as a domain shift or are simply irrelevant to the DL application. Thereby, the root cause of an error may be identified through analyzing its distribution behavior, which makes us rethink how to define *errors* and how to test the DNN by considering the effect on its trained distribution.

So far, the data in- and out-of-distribution analysis is still an early and active research area [11]. The challenge of OOD detection is that there is no perfect ground truth for validating whether one sample is in-distribution (ID) or out-of-distribution. The common approach of existing techniques to overcome this problem is utilizing significantly different datasets to approximate the ground truth. For example, CIFAR-10 is used as the ID data and MNIST is used as OOD data. When moving into thefi eld of DL testing, the differences between data can become much less as only minor perturbations are employed for generating new test cases [57], making the OOD analysis of DL test data even more difficult. To the best of our knowledge, it is currently still unknown how state-of-the-art DNN testing techniques are performing under consideration of their distribution behavior using existing OOD-detection techniques.

To bridge the gap from data distribution to DL testing activities, we conduct a large scale empirical study of the impact of data distribution awareness on the state-of-the-art DL testing techniques. In particular, we investigate the following research questions along four important perspectives:

- **RQ1. Accuracy on the OOD Detection Techniques.** Can existing OOD detection techniques detect the OOD data that is close and far to the training data? Which technique can achieve the best performance in the context of DL testing?
- **RQ2. Relationship between Mutation Operators and Data Distribution.** Mutation operators are used to generate new test cases. Thus, which mutation operator is more likely to generate OOD data and which one is more likely to generate ID data?
- **RQ3. Relationship between Testing Criteria and Data Distribution.** Testing criteria provide the coverage guidance, which filter new test cases to cover diverse internal behaviors of DNN. Thus, what is the relation of testing criteria and data distribution, i.e., which testing criterion is more likely to keep OOD data and which one is more likely to keep ID data?
- **RQ4. Root Cause Estimation for ID and OOD Errors and Robustness Enhancement.** Finally, we estimate root causes for ID and OOD errors and ask: which type of errors in terms of distribution is more effective when used for retraining in enhancing robustness?

Through answering these questions, we aim to identify the impacts of the data distribution in deep learning testing. In particular, we use three popular datasets from computer vision domain as subject benchmarks and nine OOD datasets, together with a total of 42 DNNs for evaluation, among which we trained 32 DNNs to identify the optimal OOD detection technique for DL testing. Then, to evaluate the effect of OOD for DL testing, we generate a total of over 1.2 million test cases and train 10 DNN for robustness enhancement. Regarding DL testing, our study further focuses on two of its key elements, i.e., 8 mutation operators for new test generation and 6 coverage criteria. All the datasets and results can be found on our website [34].

To summarize, this paper makes the following contributions:

- We perform a large scale empirical study on how deep learning testing affects the data distribution of the generated test cases; and how distribution aware testing influences DNN model robustness.
- Our study identifies the impact of mutation operators and coverage criteria on the distribution of the generated test cases. We find that image rotation, contrast and brightness tend to generate more ID data while image blur is more likely to generate OOD data. In terms of the coverage criteria, NBC and SNAC facilitate to generate more OOD data than others.
- We demonstrate the effectiveness of distribution aware retraining, outperforming the state-of-the-art by up to 21.5%. Based on our results, we provide guidelines on distribution-aware error selection for robustness enhancement, by studying the effect of root cause of ID and OOD errors.

To the best of our knowledge, this is thefi rst work that performed a large-scale study on the impact of data distribution behavior on DL testing. This work points out an important direction and calls for the attention of data-awareness when designing new DL testing techniques.

2 BACKGROUND

2.1 Deep Learning Testing

To test the data-driven deep neural networks, a common way is to generate new data inputs so as to capture the DNN model behavior and identify errors (e.g., incorrect prediction). The simplistic form of deep learning testing involves splitting the collected data into a training and testing set. After training the DNN model with the training set, its accuracy is tested with the testing set. One drawback is that it relies solely on the initially collected spectrum of information that usually does not cover all of the observable cases for an intended application.

Currently, quite a few techniques [8, 25, 33, 35, 43, 48, 55, 58] are proposed to test the new data-driven DL software. Coverage-guided testing is a representative and widely used technique, which usually contains three main components: the *mutation operator*, the *coverage criteria*, and the *oracle*. The mutation operator is used to

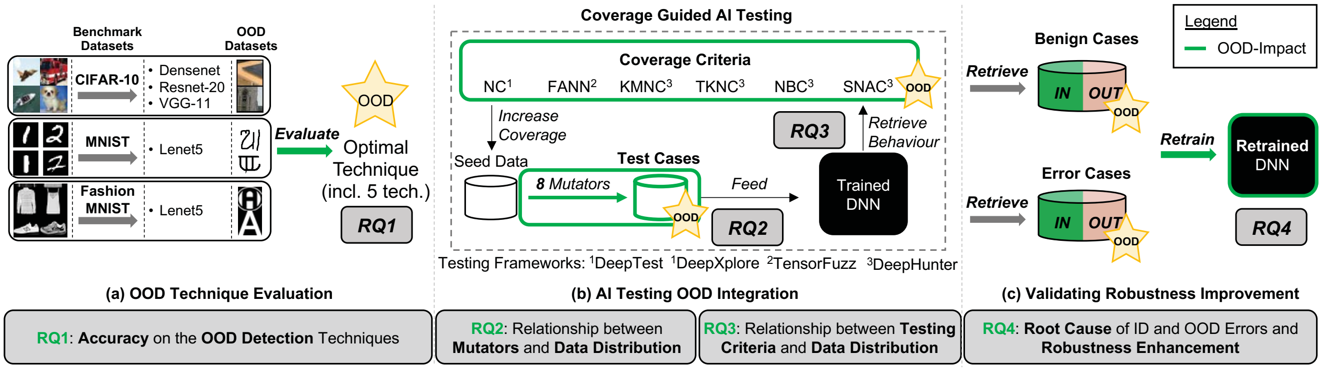
Benchmark Datasets
OOD Datasets
CIFAR-10
• Densenet
• Resnet-20
• VGG-11
MNIST
• Lenet5
Fashion MNIST
• Lenet5
Evaluate
OOD
Optimal Technique (incl. 5 tech.)
RQ1
Coverage Guided AI Testing
Coverage Criteria
NC1
FANN2
KMNC3
TKNC3
NBC3
SNAC3
OOD
Increase Coverage
RQ3
Retrieve Behaviour
Seed Data
Test Cases
8 Mutators
Feed
OOD
RQ2
Trained DNN
Testing Frameworks: 1DeepTest 1DeepXplore 2TensorFuzz 3DeepHunter
Benign Cases
Retrieve
IN
OUT
OOD
Legend
OOD-Impact
Retrain
Retrained DNN
Error Cases
Retrieve
IN
OUT
OOD
RQ4
(a) OOD Technique Evaluation
(b) AI Testing OOD Integration
(c) Validating Robustness Improvement
RQ1: Accuracy on the OOD Detection Techniques
RQ2: Relationship between Mutators and Data Distribution
RQ3: Relationship between Testing Criteria and Data Distribution
RQ4: Root Cause of ID and OOD Errors and Robustness Enhancement

Figure 1: Study Workflow and Research Questions

generate diverse test cases such that more behaviors can be tested. For example, in image classification, mutation operators such as image brightness, blur or contrast are applied under consideration of realistic settings. The coverage criteria measure the degree of how much the DNN is tested. The newly generated test cases are kept when they have achieved new coverage of the DNN. At last, the oracle is used to judge whether a new test case is a benign test case, (i.e., correctly predicted), or an error test case, (i.e., incorrectly predicted).

The assumption is that the test cases are generated by adding minor perturbations on the original input, so they should have the similar prediction result. However, the existing testing tools do not consider the distribution of the training data, which determines what data can or cannot be handled by the target DNN. The errors may be caused by the defects of the DNN model itself (e.g., inappropriate model architecture, learning process) or the lack of the training data. Hence, it is important to distinguish the different types of errors (e.g., the ID and OOD errors), which provides more feedback for the developers.

2.2 Data Distribution and Analysis Techniques

Given a dataset Given two databsets A and B, which follow the data distribution of D_A and D_B, respectively, a DNN is trained on A. If A and B have similar distributions, the well trained DNN is more likely to handle data from B correctly. If they have a totally different distribution (e.g., cat and fish images), the DNN is not expected to handle the data from B. Out-of-distribution techniques are mostly evaluated by distinguishing two totally different datasets from one another, where a large gap in corresponding distributions of scores between D_A and D_B is considered OOD and a large overlap is considered ID. More specifically, it calculates an OOD score for the new input. If the score is below the defined threshold, it is ID. Otherwise, it is OOD.

In practice, detecting the out-of-distribution data is a challenging problem, especially for the high-dimensional data. Some OOD detection has been recently proposed to address the high-dimensional issues, such as [4, 15, 16, 20, 22, 22–24, 31, 36, 40, 44, 51]. These techniques provide different ways to evaluate the distribution of training data. This work inherits those techniques and studies the distribution of the test cases generated by different DL testing strategies (e.g., mutation operators, coverage criteria).

3 OVERVIEW OF OUR STUDY

Fig. 1 shows the overview of our study that focuses on the data distribution and its effect on test cases generated by the coverage guided testing (CGT). Specifically, we focus on the three main components of the CGT for DL: 1) the effect of *mutation* on data distribution of the test cases, 2) the effect of *coverage criteria* and 3) the effectiveness of the *output* test cases on robustness enhancement.

To perform the study, we select three widely used datasets (i.e., MNIST, CIFAR-10 and FashionMNIST [18, 21, 53]) and five state-of-the-art OOD detection techniques (i.e., Baseline [15], ODIN [24], Mahalanobis [23], Outlier Exposure [16] and Likelihood-Ratio [40]). These OOD detection techniques are mainly proposed to distinguish two totally different datasets (e.g., CIFAR-10 and MNIST). However, in this work, the generated test cases are often similar to the training data. Therefore, we first design an experiment to investigate the effectiveness of existing OOD techniques in a novel and more challenging scenario where the difference between datasets for comparison is low (i.e., **RQ1**).

Based on the results of **RQ1**, we select the best OOD metric to evaluate the relationship between the data distribution and the mutation operators. In this work, we select the datasets in the image classification domain. Hence, we select 8 popular image transformations, which are mainly used in the existing CGT tools (e.g., DeepTest [48], DeepHunter [55], and TensorFuzz [33]). Then, we study which mutators tend to generate ID data and which ones tend to generate OOD data (i.e., **RQ2**). Next, we evaluate the relationship between the data distribution and the coverage criteria guidance in CGT. We select 6 popular testing criteria [25, 33, 35] to study which coverage criteria are more likely to guide the generation of OOD or ID data (i.e., **RQ3**). Adversarial training is a common way to enhance the robustness of DNNs by including the detected error data during training. Therefore, we finally study the possible root cause for ID and OOD errors and study the effectiveness of the OOD and ID data for DNN robustness enhancement (i.e., **RQ4**).

3.1 Subject Datasets and DNN Models

We select three publicly available datasets (i.e., *MNIST* [21], *CIFAR-10* [18] and *Fashion-MNIST* [53]), that are widely used in previous work. For each dataset, we follow the best DL practice and choose diverse DNN models that are able to achieve competitive results in

Table 1: Subject Datasets and DNN Models.

Dataset	Description	DNN	Train/Test Acc. (%)
CIFAR-10	General images (e.g., cats, dogs)	VGG-11	97.16 / 87.92
		DenseNet-121	99.97 / 94.46
		ResNet-18	98.14 / 91.45
MNIST	Digit images	LeNet-5	99.49 / 98.94
FashionMNIST	Fashion Images	LeNet-5	92.53 / 90.25

terms of training and testing accuracy. Table 1 shows the details about the datasets and the DNN models.

3.2 OOD Detection Techniques

We select 5 state-of-the-art OOD-detection techniques that are commonly used among related literature [4, 16, 23, 24, 36, 37, 40]. OOD techniques use different approaches to retrieve an OOD score. Some use input perturbation, and others require a specifically trained new DNN. Therefore, this work includes techniques with various approaches as follows:

- **Simple Baseline [15]**. The baseline identifies that in and out-of-distribution samples are classified with different probability distributions. The softmax prediction probability is used to determine whether an input is ID or OOD.
- **ODIN [24]**. In addition to calculate the softmax prediction probability proposed by the baseline, ODIN adds temperature scaling to the input as well as small input perturbations. They show that small perturbations have stronger effects on in-distribution samples rather than out-of-distribution samples, achieving higher ID/OOD classification performance.
- **Mahalanobis [23]**. Mahalanobis detection technique integrates the information from all layers into the score calculation. It takes the closest class for each layer, adds small noise to the test sample and finally computes the score by measuring the Mahalanobis distance [29] between the test sample and the closest class-conditional Gaussian distribution.
- **Outlier Exposure [16]**. Outlier Exposure stands out by classifying inputs with a separately trained DNN which is exposed to the same training data as the DNN used for the application. However, in addition, out-of-distribution data is integrated into the training procedure of the outlier exposure DNN model. Afterward, the maximum softmax probability is taken similar to the baseline for out-of-distribution detection.
- **Likelihood-Ratio [40]**. The latest contribution of the field utilizes a separately trained DNN, namely a generative DNN model with PixelCNN++ architecture [38]. They use an estimate of input complexity to derive a parameter-free OOD score, which can be seen as a likelihood-ratio [40].

3.3 Evaluation Metrics of OOD Detection

Out-of-distribution detection for DL testing imposes new challenges to the OOD-detection field as the compared data inherits more similarities, while the OOD-detection techniques are designed on datasets with significant differences such as comparing images of birds (CIFAR-10) and street signs of houses (SVHN). Therefore, we first select AUROC to compare the effectiveness of different OOD detection techniques (for *RQ1*) in general, and additionally *TPRN* to select a threshold based on which the OOD detector can distinguish ID data and OOD data (for *RQ2, 3, 4*). As we will see later, having multiple thresholds available is beneficial for analyzing differences for more similar data.

Table 2: Mutation Operators and Coverage Criteria.

	Pixel-Level	Affine Trans.	Tools
Mutation	Contrast, Blur Brightness, Noise	Translation, Scale Shear, Rotation	DeepTest [48], DeepHunter TensorFuzz
Criteria	Neural Coverage (NC)		DeepXplore[35], DeepTest
	k-Multisection Neuron Coverage (KMNC) Strong Neuron Activation Coverage (SNAC)		DeepGauge[25]
	Top-*k* Neuron Coverage (TKNC) Neuron Boundary Coverage (NBC)		DeepHunter[55]
	Fast Approximate Nearest Neighbor (FANN)		TensorFuzz[33]

- **AUROC**. Given an unknown input, OOD detection techniques need to identify a threshold to classify it as ID or OOD. The area under the receiver operating characteristic curve (AUROC) [14] is usually used to evaluate the performance of a classification method across multiple thresholds. The AUROC can be thought of as the probability that an anomalous example is given a higher OOD score than an in-distribution example [16]. Thus, the higher AUROC, the better the OOD detector.
- **TPR*N***, which is the true positive rate at N% true negative rate (*TPRN*). We regard OOD data as the positive class. First, we use N% true negative rate to select one threshold for the OOD detector. Then, with this threshold, we evaluate the true positive rate of the detector.
  Note that, for the parameter *N* in *TPRN*, a larger *N* means we select a bigger threshold such that more data is perceived under the threshold as ID (i.e., higher true negative rate). *Thus, a larger N provides more confident measurement for detecting OOD data while a smaller N provides more confident measurement for detecting ID data.*

3.4 Mutation Operators and Coverage Criteria

For a thorough analysis of DL testing, we select 8 mutation operators and 6 coverage criteria, which represent the state-of-the-art and are widely used in the existing testing tools, i.e., DeepTest [48], DeepHunter [55] and TensorFuzz [33]. Table 2 shows the detailed information about the selected mutation operators and coverage criteria. Column *Tools* represents which techniques are utilized by which tools. All mutation operator parameters are carefully chosen by following previous work and maintaining realistic bounds, e.g. rotation is capped at 40 degree. All configuration can be found on our website [34].

3.5 Study Design

The empirical evaluation for each RQ is designed as follows:

RQ1. Accuracy on OOD Detection Techniques. For RQ1, we select five aforementioned OOD detection techniques, which are widely used to distinguish two totally different datasets. To compare their effectiveness, we design three different experiments as follows:

1. First, like the usual way, we evaluate the techniques in distinguishing the ID data and OOD data in two different datasets, i.e., the training data as ID data and another dataset as OOD data with significantly different features. For the target datasets CIFAR-10, MNIST and FashionMNIST, we select 4 different datasets that are regarded as OOD datasets.
2. Second, we evaluate the inverse extreme case, by taking the distribution difference between the training data and test data

of the same benchmark dataset (e.g. CIFAR-10 train vs CIFAR-10 test). Since we can expect both datasets to follow the similar distribution, we validate that the OOD-techniques are able to identify in-distributions inputs, too, highlighting that the trained distribution encompasses unknown data, which is relevant to the DL application.

(3) Third, we present an evaluation technique by splitting the benchmark's training dataset into 2 subsets based on their classes, i.e., half of the classes are taken as the training data and the rest half of the classes are taken for OOD test set. Even though the other half of classes are not trained, overall similarities exist as they are from the same domain. Thereby, we present a similar scenario as encountered for deep learning testing.

The three settings are designed to showcase a difference in data distribution. In Setting 1, the two datasets are expected to have totally different distributions. In Setting 2, the two distributions should be almost the same. Finally, Setting 3 should lie between the first two settings, as the classes are not known, however, the compared datasets are from the same domain.

RQ2. Relationship Between Mutation Operators and Distribution. In DL testing, mutation operators are used to generate diverse test cases. Hence, RQ2 aims to study the effect of the mutation operator by observing the data distribution of the generated test cases. We use each of the aforementioned mutation operators to randomly generate 2,000 test cases based on 200 seed inputs from the test set. Based on the results of RQ1, we select the best OOD detection technique to compare the distribution of generated test cases with the original training data.

RQ3. Relationship Between Testing Criteria and Distribution. In deep learning testing, coverage criteria provide the guidance for test case generation. Specifically, they are used to filter some test cases from randomly generated mutants and keep only the mutants that can improve the coverage. RQ3 aims to study which coverage criteria are more likely to generate ID or OOD data. To answer this question, we generate for each DNN model 2,000 test cases for each mutation operator for each coverage criterion based on the seeds of RQ2. Then, we compare the distribution of the test cases generated with different coverage criteria guidance.

RQ4. Root Cause for ID and OOD Errors and Robustness Enhancement. After identifying the optimal OOD detection technique setting for deep learning testing (from RQ1) and analyzing the effect of mutation operators and coverage criteria (from RQ2 and RQ3), we aim to study the root cause for ID and OOD errors and whether distribution-aware test cases are more effective in enhancing robustness by retraining.

For robustness enhancement we select 1,000 uniformly class-distributed seeds extracted from the training set. Based on these seeds, we generate 5 different sets of data: *ID-errors where* $TPR85 = 0$, *ID-errors where* $TPR95 = 0$, *OOD-errors where* $TPR95 = 1$, *OOD-errors where* $TPR99 = 1$ and *random errors*. Each set contains 10,000 error test cases. Then, we add each set of error test cases into the original training set consisting of 50,000 samples for retraining the DNN.

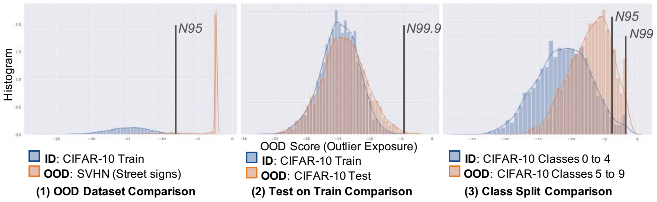

Figure 2: Visualization of the distribution difference between different datasets on DenseNet-121

4 EMPIRICAL STUDY

We summarize the important results and findings in this paper, while complete results can be found on our website [34].

Table 3: Average Results (AUROC) of different OOD detection techniques (in %)

Dataset	DNN	Base.	ODIN	Maha.	OE	Like.
CIFAR-10	DenseNet	98.8	98.9	94.1	**99.5**	68.0
	ResNet	95.8	97.1	**99.7**	97.3	68.0
	VGG-11	92.5	94.3	87.5	**97.9**	68.0
MNIST	LeNet-5	98.7	98.7	98.6	**99.7**	93.4
Fashion-MNIST	LeNet-5	91.2	91.9	**99.9**	99.4	63.0

4.1 RQ1: OOD Detection Accuracy.

We first evaluate the state-of-the-art of OOD detection techniques to identify the optimal technique. The results (see Section 3.5) are visualized by examples in Figure 2 and are as follows:

4.1.1 Results on totally different datasets. First, we perform a comparative study to investigate the performance of existing OOD-detection techniques. We prepare frequently used OOD datasets [15, 16, 23, 24, 40] and the training data of the DNN as ID data. We select the OOD datasets: *SVHN [32]*, *iSUN [54]*, *Picsum [39]* and *Omniglot [19]*. Detailed description of each OOD dataset can be found in [34]. In addition, we scale the benchmark datasets Imagenet, CIFAR-10, MNIST and FashionMNIST to the same dimensions as the trained dataset and convert them into grayscale when necessary, e.g., for MNIST or FashionMNIST. Due to the significant difference in datasets, a ground truth of ID and OOD can be assumed.

For each DNN, we evaluate the OOD detection performance on different OOD datasets. Table 3 shows the average AUROC score of each OOD detection technique on each DNN. The best results are in bold. We can observe that, except for the Likelihood-Ratio, other OOD detection techniques are effective (91%+) in detecting significantly different OOD data. Overall, Outlier Exposure shows the highest overall performance while Mahalanobis is the second best. For example, for VGG-11, OE achieves 97.9% AUROC score that is much higher than others and Mahalanobis has 87.5% AUROC score. Likelihood-Ratio has the same results for the three DNNs of CIFAR-10 as its DNN-independent. Our results show that Likelihood-Ratio performs the worst for CIFAR-10, MNIST and FashionMNIST (the average score of AUROC is 74.8%).

4.1.2 Results on the training data and test data. In the previous section, we evaluate the OOD techniques for detecting OOD data

Table 4: Results of OOD detection on the test set. (in %)

OOD Tech.	Base.		ODIN		Maha.		OE		Like.	
*TPR**N***	99	99.9	99	99.9	99	99.9	99	99.9	99	99.9
DenseNet-121	5.92	3.02	0.74	0.02	0.80	0.03	4.94	0.10	1.52	0.22
ResNet-18	2.06	0.08	0.64	0.00	1.29	0.00	1.46	0.08	1.52	0.22
VGG-11	2.40	0.26	0.90	0.08	1.14	0.00	0.96	0.14	1.52	0.22
MN. LeNet-5	1.48	0.38	1.24	0.06	1.37	0.03	2.00	0.38	0.42	0.02
FMN. LeNet-5	1.44	0.24	0.74	0.00	0.54	0.00	1.22	0.20	1.12	0.12

Table 5: Results of OOD detection when training half of the classes of the training set while testing OOD with the other half of untrained classes. (in %)

OOD Tech.	Base.		ODIN		Maha.		OE		Like.	
*TPR**N***	95	99	95	99	95	99	95	99	95	99
DenseNet-121	22.7	7.6	25.0	9.2	13.7	2.5	23.8	3.4	1.8	0.1
ResNet-18	33.7	14.9	34.4	15.6	15.2	3.9	27.7	8.2	1.8	0.1
VGG-11	30.8	11.6	31.7	11.8	19.6	1.2	10.0	1.8	1.8	0.1
MN. LeNet-5	71.7	45.8	73.7	51.4	92.3	78.7	86.0	64.9	4.9	0.8
FMN. LeNet-5	15.3	2.5	18.5	4.0	98.9	93.3	82.8	76.8	14.5	3.2

that is very different from the training data. In this section, we evaluate the techniques for distinguishing ID/OOD data in the test set of the same benchmark dataset that follows a very similar distribution as the training set.

Table 4 summarizes the results on the test data for each dataset. We select two metrics *TPR99* and *TPR99.9* (in the second row), which means that we select two thresholds with high accuracy in detecting ID data (i.e., 99% and 99.9% high accuracy in detecting ID data). Then, we observe how much of the test data is detected as OOD data under these two thresholds. The results show that the overall values are very small and follow our expectations that there is little OOD data in the test set. It also demonstrates that all techniques are effective in identifying ID data. Most importantly, it demonstrates that the test set data follows almost an identical distribution as the training data, highlighting that the trained distribution of the DNN integrates unknown data, which is considered relevant to the DNN application (middle of Figure 2).

4.1.3 Results of splitting the training data. After evaluating the ability in detecting OOD and ID data in their extreme cases (i.e., Section 4.1.1 and 4.1.2), we design a final study that is between these two extreme cases. We split the training dataset equally in two separate sets based on their labels. Then, we train the same DNN architectures as before but with only five outputs, since we only use half of the classes (e.g., 0-4). Similarly, we evaluate the capability of the detection techniques by distinguishing the trained classes from the non-trained classes of data.

Table 5 shows how much of the non-trained class data is detected as OOD data. We can clearly see that the values lie between the values in Table 3 and Table 4. Another observation is that more OOD data can be identified in the grayscale images (i.e., MNIST and FashionMNIST) while less OOD data is identified in the color images. The reason may be that the grayscale images have lower dimensionality. Thereby, it is easier to capture the content changes between the two subsets. However, for the complex color images, they may share a similar style in the same domain (e.g., the background), which makes it more difficult in distinguishing the two classes. We also found that TPR95 can identify more OOD data than TPR99 as TPR95 selects a smaller threshold of the trained distribution.

Table 6: OOD Data by Different Mutation Operators. (in %)

	Mutators	DenseNet-121		ResNet-18		LeNet-5 (MNIST)	
		TPR85	TPR99	TPR85	TPR99	TPR85	TPR99
Benign	Translation (-3, 3)	32	**7**	36	**9**	41	**13**
	Scale (0.7, 1.2)	75	36	74	43	46	10
	Shear (-0.6, 0.6)	65	18	53	17	84	54
	Rotation (-40, 40)	63	13	45	11	67	21
	Contrast (0.5, 0.13)	28	8	47	6	80	41
	Brightness (-32, 32)	19	**4**	17	**5**	88	68
	Blur (1, 10)	87	77	91	83	96	87
	Noise (1, 4)	37	**0**	17	**0**	29	**0**
	Average	50.6	20.5	47.3	21.7	66.2	36.8
Error	Translation (-3, 3)	77	48	59	**21**	97	84
	Scale (0.7, 1.2)	99	90	96	85	96	66
	Shear (-0.6, 0.6)	78	31	76	**27**	99	87
	Rotation (-40, 40)	76	**25**	79	**25**	99	76
	Contrast (0.5, 0.13)	77	66	54	28	100	99
	Brightness (-32, 32)	94	**1**	64	43	100	99
	Blur (1, 10)	96	86	98	91	100	100
	Noise (1, 4)	97	77	83	64	100	18
	Average	86.9	53.1	76.3	48.1	98.9	78.7

The overall results give us directions on how to select the threshold for different datasets. If the two datasets are very similar but suspected to be from different distributions, we can select a smaller *N* in TPR**N**, which can detect more OOD data. If the two datasets are very different, we can select a larger *N* that can distinguish the two datasets.

> **Answer to RQ1:** Overall, our results show that Outlier Exposure on Densenet-121 architecture performs the best and the results are consistent on all benchmark datasets. The existing techniques can detect the ID data effectively where most of the test data are correctly classified as in-distribution. Splitting the classes of the training set imposes a challenge to the detection techniques and grants a new perspective on their performance for application-realistic settings.

4.2 RQ2. Relationship between Mutation Operators and Data Distribution.

In the following experiments, we select Outlier Exposure as the optimal OOD-detection technique for DL testing based on the results of RQ1. In addition, due to the space limit, for mutation operator evaluation we only show the results of DenseNet-121, ResNet-18 and LeNet-5 for MNIST. VGG-11 has lower complexity and Fashion-MNIST is very similar to MNIST. To evaluate the effect of the mutation operators, we randomly select 200 data samples with uniformly distributed classes from the benchmark's test set as the seed images. Then, we apply each mutation operator to the seeds, generating 2,000 benign test cases and 2,000 error test cases. Table 6 shows the results of the OOD data generated by each mutation operator. Column *Mutators* shows each mutation operator. The parameters are chosen very conservatively and follow previous contributions while changing the original image slightly [55]. The exact settings for realistic mutation is at [34].

The generated test cases and the original dataset are similar. Therefore, we build on our findings from RQ1 and introduce both TPR85 and TPR99 settings to detect the OOD test cases. We can be more certain that samples tend to be OOD with the threshold of 99%. Nevertheless, if TPR85 shows a low score, the likelihood is more samples tend to be in-distribution.

Considering the results of the benign test cases and error test cases, we find that the errors test cases are considered out-of-distribution at a higher rate than the benign test cases. Error test cases for DNNs trained on CIFAR-10, namely DenseNet-121 and ResNet-18, have an average TPR99-score of 50.6%, while the benign test cases only show 21.1%. For example, focusing on the mutation operator Image Noise, we can observe that benign test cases seem to be entirely in-distribution ($TPR99 = 0$) while the error test cases show a TPR99-score of 77%, 64% and 18% respectively for all three DNNs. This behavior indicates that error test cases are more likely to be out-of-distribution and thus they are more likely to be predicted incorrectly.

Considering the results between different DNNs, we find that DenseNet-121 and ResNet-18 (trained on CIFAR-10) share similar averages between all three evaluation metrics. However, they have different trends when compared with LeNet-5 which is trained on MNIST. This behavior shows that the results are data-driven, highly depending on the trained datasets in general. For a simple grayscale image (MNIST) which has low dimensionality, the mutation operators may change it a lot and generate OOD data. However, for more complex color images, the mutation operators (with the defined conservative parameter setting) will change little on the high dimensional data, which makes lower OOD data in CIFAR-10. For example, in MNIST, the average results are 36.8% and 78.7% for benign test cases and error test cases, respectively. While for ResNet-18 on CIFAR-10, the results are 21.7% and 48.1%.

Comparing different mutation operators individually, wefind that image blur tends to generate the most OOD data. The benign and error test cases of Blur are considered 77% and 86% as OOD data (based on TPR99). Image Scale follows a similar pattern especially for error test cases on CIFAR-10 (85% and 90% for Densenet-121 and ResNet-18 respectively). For Image Scale, after the image is becoming smaller, the black color is used to complement the background, and therefore is more likely to be OOD. However, in error test cases of MNIST, the background of the original images is black already. Hence, the Image Scale only generates 66% OOD data, which is the smallest value in all mutation operators. In addition, we find that mutation operators, i.e., Translation, Shear, Rotation and Brightness tend to generate fewer OOD data. For example, Brightness only generates 1% OOD data for DenseNet-121 and Rotation generates 25% OOD data for both, DenseNet-121 and ResNet-18.

> **Answer to RQ2**: The data distribution generated by mutation operators is dependent on the datasets. Considering the same mutation operators, more test cases tend to be more OOD for grayscale images and less for color images. Image blur and Image Scale are the mutations strategies where the highest OOD-score is observed, whereas Image Rotation, Shear, Brightness and Contrast generate fewer OOD data. The error test cases are more likely to be OOD than benign test cases.

Table 7: OOD data (under TPR99) generated by different coverage guidance. (in %)

			Rand	NC	KMNC	NBC	SNAC	TKNC	FANN
Benign	DenseNet-121	Rotation	13	8	14	**33**	22	7	8
		Contrast	8	24	9	52	**59**	17	21
		Brightness	4	14	5	40	**42**	13	14
		Blur	**77**	36	58	**77**	51	40	53
		All	14	38	11	34	**42**	14	13
		Average	23	24	20	**47**	43	18	22
	ResNet-18	Rotation	11	6	**20**	13	10	10	12
		Contrast	6	8	9	43	**44**	14	9
		Brightness	5	4	3	11	**15**	11	5
		Blur	83	24	69	**81**	71	34	47
		All	8	33	11	**59**	52	18	11
		Average	23	15	22	**41**	38	17	17
	LeNet-5	Rotation	**21**	1	11	17	7	12	15
		Contrast	**41**	1	3	16	4	15	25
		Brightness	**68**	2	39	29	1	28	32
		Blur	**87**	3	43	4	1	32	55
		All	31	3	25	**69**	37	25	22
		Average	**49**	2	24	27	10	23	30
Error	DenseNet-121	Rotation	13	26	29	32	**46**	36	34
		Contrast	8	**95**	40	75	85	72	62
		Brightness	4	13	31	48	**85**	40	38
		Blur	77	**93**	87	86	90	89	87
		All	14	66	55	67	**72**	58	57
		Average	23	59	48	61	**76**	59	56
	ResNet-18	Rotation	25	35	26	26	34	**37**	29
		Contrast	28	65	50	95	**92**	67	56
		Brightness	**43**	20	9	13	10	33	36
		Blur	91	86	89	**93**	89	87	85
		All	56	60	52	66	**68**	59	55
		Average	48	53	45	**59**	**59**	57	52
	LeNet-5	Rotation	76	56	59	68	59	73	**77**
		Contrast	99	**100**	87	**100**	**100**	98	98
		Brightness	99	**100**	**100**	**100**	**100**	100	100
		Blur	**100**	81	95	91	80	97	97
		All	82	73	74	**91**	86	82	78
		Average	**91**	82	83	90	85	90	90

4.3 RQ3. Relationship between Testing Criteria and Data Distribution.

In RQ2, we have identified the behavior of the mutation operators without applying any guidance to the DL testing flow. Now, we add coverage criteria to the testing flow to guide the test case generation. Therefore, we evaluate the effect of coverage criteria on the data distribution and compare it with the results of RQ2 as the baseline.

Based on the results of RQ2, we select 4 mutation operators, i.e., Rotation, Contrast and Brightness which are more likely to generate ID data and Blur which is more likely to generate OOD data. Table 7 shows the results of how many test cases are OOD data with different coverage guidance. The results are evaluated by Outlier Exposure with TPR99. The third column shows the mutation operators. Row *All* means we use all default mutation operators for test case generation. Column *Rand* shows the results without coverage criteria guidance which is from Table 6. The columns following *Rand* show the results for each coverage criterion. Note that, in coverage-guided testing, benign test cases are generated with the guidance of the coverage criteria while error test cases are not filtered by the coverage criteria.

Considering the results of benign test cases (i.e., under coverage guidance), we find that, compared with the random generation (i.e., without coverage guidance), KMNC TKNC and FANN decrease the OOD data ratio, whereas NBC and SNAC increase the OOD data ratio. For example, in DenseNet-121, the average value with random generation is 23% while the average values with KMNC, TKNC and FANN guidance are 20%, 18% and 22%, respectively. The average values with NBC and SNAC are 47% and 43%, respectively.

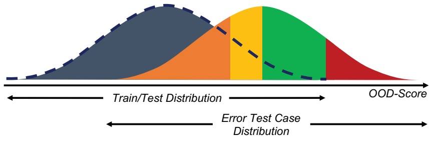

Figure 3: Train/Test and Error Test Case Distributions colored in different ranges defined by quantiles of train/test distribution

It is consistent with their definitions. For example, KMNC mainly considers major behaviors of the DNN and FANN generates test cases that are near the original seeds. Thus, ID data is more likely to be generated with the guidance of these two coverage criteria. However, NBC and SNAC mainly consider the boundary of the DNN. Hence, they can generate more OOD data.

Considering the specific mutation operators individually, we find that mutation and coverage criteria influence each other. For example, for DenseNet-121 and ResNet-18 when using Image Contrast, most of the coverage criteria increase the OOD data ratio including KMNC, TKNC and FANN, where usually these coverage criteria tend to decrease the OOD data ratio in all data. It indicates that the coverage criteria guide the test case generation by covering more diverse DNN behaviors for this particular mutation operator. In the contrary, Image Blur changes the image a lot causing most test cases to be OOD under the random mutation setting already. In this case, all coverage criteria decrease the OOD data ratio compared to the random mutation setting. This behavior means that most OOD test cases generated by Image Blur may be filtered by the coverage criteria. For example, the coverage criteria in DeepGauge [25] are defined based on the profiling of training data. The blurred images are far from the training data, thus they cannot achieve new coverage under these criteria. The results demonstrate that existing coverage criteria have obvious effects on mutation operators but vary in their behavior depending on the underlying mutation operator.

Another interesting observation is found for MNSIT trained on LeNet-5 which is different from the results for CIFAR-10. For the benign test cases of LeNet-5, almost all coverage criteria decrease the OOD data ratio, where for CIFAR-10 this behavior could only be observed for selected mutation operators. It seems that for simple black and white or grayscale images, the random mutation requires much change to produce a test case, which is why a lot of OOD data are generated. However, the testing criteria filter most of these data points as they do not contribute to increasing the coverage.

For the error test cases, we find that, compared with random mutation, almost all the coverage criteria increase the OOD data ratio in ResNet-18 and DenseNet-121. In LeNet-5, most of the erroneous test cases tend to be OOD data (TPR99). Actually, error test cases have no direct relationship with the coverage criteria as they are not filtered by the criteria directly. However, they are generated by mutating the benign data that are generated under the coverage guidance.

Answer to RQ3: Our results show that, existing coverage criteria affect the data distribution of generated test cases, which is important to address when designing a test scenario. KMNC, TKNC, NC and FANN tend to decrease the number of OOD benign test cases while NC and NBC tend to increase the OOD benign test cases. For the mutation operators that tend to generate fewer OOD data such as rotation and contrast, the existing coverage criteria can increase the number of OOD data by covering more behaviors of the DNN. For the mutation that tends to generate more OOD data such as blur, the existing coverage criteria can decrease the number by filtering some data with the coverage guidance. For grayscale images, the coverage criteria may decrease the number of OOD data with random mutation operators. The coverage criteria may increase the OOD data for generated error test cases.

4.4 RQ4. Root Cause of ID and OOD Errors and Robustness Enhancement.

We hypothesize that ID error test cases tend to be a result of defects in the DNN model while OOD error test cases tend to be a result of missing data in the training set. Therefore, two experiments are designed to study the hypothesis. First, we use other DNN models with different architectures but trained on the same training data to predict the ID and OOD errors of the model under test. Following our hypothesis, we expect other DNN models to predict ID errors more correctly than OOD errors. Second, we retrain the model under test with additional ID and OOD error test cases. We expect that the newly added data helps in correctly predicting OOD errors more effectively.

4.4.1 Robustness Enhancement with Adjusting Models. For the first hypothesis, i.e., ID error test cases tend to be a result of defects in the DNN model, we select six other DNN variants (VGG-11, VGG-13, ResNet-18, ResNet-34, DenseNet-121, DenseNet-169), which differ in their architecture but are learned from the same training dataset. Note that these models can be regarded as the simulation of the potential improvement of the original model (e.g., finetune or change in architecture). We expect errors found on, e.g. ResNet-18, to be predicted correctly by some of the other five DNN variants with special attention on whether ID errors tend to be predicted correctly more likely than OOD errors.

Table 8 shows the results of the cross validation on other five models. Full evaluation can be found on the website [34]. For each model, we collect 10,000 ID errors and 10,000 OOD errors, respectively. The accuracy shows how much data is correctly predicted on average by the other models (i.e., it could be handled well by improving the model). Overall, the results indicate that ID errors tend to be fixed at a higher likelihood than OOD errors through DNN adjustments such as changing its neural architecture, which is consistent with our hypothesis. For example, by changing the models, 32.4% ID errors could be fixed while only 20.4% OOD errors could be fixed.

4.4.2 Robustness Enhancement by Adding Training Data. For the second hypothesis, i.e., OOD error test cases tend to be a result

Table 8: DNN Model Agreement on ID and OOD Errors.

Test Model TPR85	Error Type TPR99	Cross Validation Accuracy (%)
ResNet-18	ID-Error	29.7
	OOD-Error	21.2
DenseNet-121	ID-Error	32.4
	OOD-Error	20.4

Table 9: Results for robustness enhancement on different dataset and DNNs. (in %)

	Test Set	CIFAR-10	RANDOM	ID85	ID95	OOD95	OOD99
ResNet-18	CIFAR-10 Test	91.5	90.1	90.5	90.2	90.1	89.8
	ID Error	0.0	50.9	56.9	50.5	**60.4**	13.3
	OOD Error	0.0	**64.1**	54.3	54.3	62.6	10.9
	RAND Error	0.0	58.1	53.5	50.0	**61.9**	12.0
DenseNet	CIFAR-10 Test	94.5	89.2	89.6	89.9	89.7	89.6
	ID Error	0.0	**60.7**	47.3	47.0	**60.5**	49.1
	OOD Error	0.0	46.6	55.1	49.9	**59.0**	58.3
	RAND Error	0.0	48.7	53.5	47.7	**59.2**	55.0
	Total Average	0	54.9	53.4	49.9	**60.6**	33.1

of missing training data, we generate multiple ID/OOD data and evaluate the robustness by retraining with them.

Specifically, we propose five datasets, each of which contains original training data and 10,000 error test cases, which are generated from 1,000 initial seed inputs but vary in their OOD score (presented by color and threshold in Figure 3: Orange and yellow areas indicate ID error test cases with TPR85 and TPR95 to be 0% respectively (related to as ID85 and ID95). Green and red areas indicate OOD error test cases with TPR95 and TPR99 to be 100%, respectively (related to as OOD95 and OOD99). We further use errors drawn randomly from the distribution as another dataset.

In addition, we prepare four test sets: the original test set, 2,000 ID errors, 2,000 OOD errors and random errors, which are used to test the new DNNs. Here, we only include two DNNs trained on CIFAR-10 due to most of the errors for LeNet-5 on MNIST are considered OOD.

Table 9 shows the results of the retrained DNNs. Overall, for the new DNNs, the accuracy on test set is reduced on average by 1.5%. At the same time, the performance on correctly classifying random errors is improved up to 61.9% percentage points. However, the results vary quite a lot with the data distribution. OOD error test cases (green area, column OOD95) show the highest overall accuracy with 60.6% average accuracy, while ID85 and ID95 only classify 53.4% and 49.9% correctly. This promotes the idea that OOD errors are more effective in generalizing the model towards new data. However, not all OOD errors can be considered effective for retraining. Column OOD99 shows the lowest total average, which indicates, that at some point error test cases can not be considered directly benefiting the overall DNN application as they are too different from the overall distribution.

Compared with random retraining which can be considered a baseline of recent work, distribution aware retraining increases robustness on average by 10.2% and up to 21.5% for Random Error Test Set on DenseNet-121.

Answer to RQ4: The results demonstrate that ID-errors tend to be fixed via DNN adjustments, while OOD-errors seem to require further training data for being correctly classified. When retraining, OOD errors tend to be on average 10.4% more effective in improving the robustness of the DNN than ID errors or randomly chosen ones. Furthermore, not all OOD errors help the model to generalize, indicating that the OOD-score distance towards the trained/tested DNN distribution matters when choosing the right data for enhancing robustness.

4.5 Discussion and Research Guidance

Based on our results, we pinpoint the following research directions:

- **OOD Detection for DL Testing (RQ1).** In DL testing, it is still challenging to distinguish ID and OOD data especially when more similarities between the two tested data types exist. Therefore, fine-grained thresholds seem helpful in gaining a better understanding in similar cases. Our results in Fig. 2 provide the following guidance: if the testing tool aims at generating ID test cases, a smaller N should be selected. If we want to generate OOD test cases, a larger N should be selected.
  Research Guidance: a possible direction is to develop OOD techniques, which can effectively detect fine-grained OOD data for deep learning testing.
- **Mutation Operators and Coverage Criteria (RQ2&3).** Our results show that the existing mutation and coverage criteria have different effects on ID data or OOD data generation. To build the distribution-aware DL testing tools, we could develop distribution-based coverage criteria that can filter some OOD data or ID data.
  Research Guidance: DL testing tools should be aware of distribution. A promising direction is to develop more fine-grained distribution-aware criteria for the test selection.
- **Robustness Enhancement (RQ4.)** Our initial results have shown that distribution-aware retraining is more effective in robustness enhancement than the distribution-unaware retraining. It seems that root causes for ID errors are partially model dependent while OOD errors can be effectively fixed with new training data.
  Research Guidance: A future research direction is to further analyze the root cause of ID and OOD errors, especially in an even more fine grained setting which can provide guidance for repairing the model from a data and DNN architecture perspective under regard of the presented threshold of this work.

4.6 Threat to Validity

The selection of the datasets and DNNs could be a threat to validity. We try to counter this by using eight publicly available and popularly used datasets and cross-validating results on three different DNN architectures. OOD detection is a very challenging problem as there is no perfect ground truth, which could be a threat to validity. To this end, we select multiple state-of-the-art OOD techniques for a comparative study. In addition, we also design three fine-grained experiments (in RQ1) where the ground truth can be approximated by the inherited difference. Thereby, we can identify the optimal OOD detection technique for DL testing to compare the distribution performance for DL testing associated data. The results of Tables 4, 5, 6, 7, 8 and 9 may be biased on the seeds and the generated data. We try to counter this by randomly selecting the same number of seeds for each class, generate a large number of mutants, and compare the averaged results.

5 RELATED WORKS

Deep Learning Testing. Adversarial attacks have been extensively studied to perform perturbation on input data to fool a DNN in different applications [2, 10, 13, 30, 41, 52]. However, such perturbations are often obtained through gradient- or optimized-based searching, which may rarely happen in a physical environment. In addition, it has been demonstrated that there are many issues during the DL development and depolyment phases [12], which calls for the requirement of systematic DL testing. Different from the adversarial attack, DL testing considers generating new tests by performing mutations that simulate noise patterns from the physical environment (e.g., image brightness change, rotation) with defined bounds to maintain realism, e.g., rotation is limited to 40 degrees. To estimate the DL testing sufficiency and providing testing guidance, many testing criteria have been proposed. DeepXplore [35] originally proposed the neuron coverage. Inspired by this, DeepGauge [25] proposed a set of morefi ne-grained testing criteria such as KMNC, NBC, etc. DeepConcolic proposed MC/DC test criteria [42]. Furthermore, combinatorial testing criteria [26] and Surprise Adequacy [17] are also proposed. The testing criteria above mainly focus on feed-forward neural networks, while DeepStellar [8] proposed the model-based testing criteria for recurrent neural networks.

These proposed testing criteria for DNN are used to guide the test generation process, such as in [27, 28, 33, 35, 48, 55, 56, 59]. In addition, DeepTest [48] and DeepRoad [58] also generate images with Generative Adversarial Networks (GANs). Compared with the basic transformations (e.g., adding noise, rotation), the GAN-based techniques can perform advanced scene transformation, but are computation-intensive requiring training a GAN, the quality of the generated images can not be easily guaranteed.

Similarly, we also leverage the basic mutation operators and coverage criteria in existing testing tools for the study. However, this paper is orthogonal to existing DL testing work in that our focus is to investigate the importance of data distribution and how it impacts existing testing techniques. Our results show that, although existing testing techniques are able to detect *thousands of errors* as discussed in their original papers, a large portion of these errors may not contribute directly to the desired result when retraining or to the overall DL application. Therefore, considering the data distribution during DL testing is of great importance to properly identify the real weakness of a DNN for further processing.

Out-of-Distribution-Detection Techniques. While being important, the out-of-distribution analysis is challenging especially for high-dimensional data. Dan Hendrycks et. al introduced a baseline approach [15], which utilizes the maximum Softmax probability. Correctly classified examples tend to have greater maximum Softmax probabilities than erroneously classified and out-of-distribution examples, allowing for their detection. The ODIN [24] and Mahalanobis technique [23] propose to apply input perturbations by adding noise or temperature to the input, by which they intensify the ability of the baseline algorithm to differentiate confidence between in and out-of-distribution errors. Outlier Exposure [16] takes a different approach. Here, a separate DNN is taken, and trained with an additional infusion of declared OOD-samples, such as large scale data images TinyImages [9], while the score is calculated in a similar fashion to the baseline. One advantage of the technique is independence towards the DNN used for the application. Just towards the data. Thereby, a bias for OOD-detection caused by a given DNN may be overcome more efficiently.

Finally, more recent contributions propose to use likelihood-ratios at their core [36, 40] and utilize generative PixelCNN++ architecture to retrieve bits per dimension to calculate OOD scores. Other techniques, which are not capable of classifying single inputs [31], require heavy DNN architectural adjustments, such as adding an additional class [51] or taking multiple techniques as ensemble [4, 20]. This is not considered in this work due to their imposed limitations towards DL testing.

Existing OOD detection methods are mostly proposed to work on datasets with a large difference. Therefore, it is still unclear whether and to what extent existing OOD detection methods can be used for the challenging DL testing scenario, where the generated test data often differs from its original counterpart in a minor way. In this work, we selected the state-of-the-art OOD-detection techniques to investigate their effectiveness, its connection and usefulness for DL testing purposes. Wefi nd that data-distribution awareness could be a key for more effective and interpretable DL testing towards providing better quality assurance.

6 CONCLUSION

In this paper, we conduct a large-scale empirical study on the state-of-the-art OOD techniques towards understanding the data distribution and its impact on DL testing activities. Our results show that the existing OOD detection techniques can distinguish the OOD data from the newly generated test cases, even for challenging cases where the test data is very similar to the training data. Our study further shows that existing image mutation operators and testing criteria can greatly affect the distribution of the generated test cases. Finally, we demonstrate that distribution-aware dataset tends to be more effective in robustness enhancement. This study makes thefi rst step along this direction towards understanding the data-driven nature of DL software for testing activities. The results of this paper call for the attention of data-distribution awareness during designing testing and analysis techniques for DL software, which builds the foundation towards developing more effective DL testing techniques.

ACKNOWLEDGMENTS

This work was supported by Singapore Ministry of Education Academic Research Fund Tier 1 (Award No. 2018-T1-002-069), the National Research Foundation, Prime Ministers Office, Singapore under its National Cybersecurity R&D Program (Award No. NRF2018 NCR-NCR005-0001), the Singapore National Research Foundation under NCR Award Number NSOE003-0001 and NRF Investigatorship NRFI06-2020-0022. It was also supported by JSPS KAKENHI Grant No.20H04168, 19K24348, 19H04086, and JST-Mirai Program Grant No.JPMJMI18BB, Japan. We also gratefully acknowledge the support of NVIDIA AI Tech Center (NVAITC) to our research.

REFERENCES

[1] BBC. 2020. *AI 'outperforms' doctors diagnosing breast cancer.* https://www.bbc.com/news/health-50857759

[2] Nicholas Carlini and David A. Wagner. 2016. Towards Evaluating the Robustness of Neural Networks. *CoRR* abs/1608.04644 (2016). arXiv:1608.04644 http://arxiv.org/abs/1608.04644

[3] carnegieendowment. 2019. *The Global Expansion of AI Surveillance.* https://carnegieendowment.org/2019/09/17/global-expansion-of-ai-surveillance-pub-79847

[4] Hyunsun Choi and Eric Jang. 2019. Generative Ensembles for Robust Anomaly Detection. https://openreview.net/forum?id=B1e8CsRctX

[5] McKinsey Co. 2019. *AI Adoption Advances but Foundational Barriers Remain.* https://www.mckinsey.com/featured-insights/artificial-intelligence/ai-adoption-advances-but-foundational-barriers-remain

[6] Datamonsters. 2017. *10 Applications of Artificial Neural Networks in Natural Language Processing.* https://medium.com/@datamonsters/artificial-neural-networks-in-natural-language-processing-bcf62aa9151a

[7] Google Deepmind. 2019. *AlphaStar: Grandmaster level in StarCraft II using multi-agent reinforcement learning.* https://deepmind.com/blog/article/AlphaStar-Grandmaster-level-in-StarCraft-II-using-multi-agent-reinforcement-learning

[8] Xiaoning Du, Xiaofei Xie, Yi Li, Lei Ma, Yang Liu, and Jianjun Zhao. 2019. DeepStellar: Model-Based Quantitative Analysis of Stateful Deep Learning Systems. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Tallinn, Estonia) *(ESEC/FSE 2019)*. Association for Computing Machinery, New York, NY, USA, 477–487. https://doi.org/10.1145/3338906.3338954

[9] Rob Fergus, Yair Weiss, and Antonio Torralba. 2009. Semi-Supervised Learning in Gigantic Image Collections. In *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta (Eds.). Curran Associates, Inc., 522–530. http://papers.nips.cc/paper/3633-semi-supervised-learning-in-gigantic-image-collections.pdf

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations.* http://arxiv.org/abs/1412.6572

[11] Google. 2019. *Improving Out-of-Distribution Detection in Machine Learning Models.* https://ai.googleblog.com/2019/12/improving-out-of-distribution-detection.html

[12] Qianyu Guo, Sen Chen, Xiaofei Xie, Lei Ma, Qiang Hu, Hongtao Liu, Yang Liu, Jianjun Zhao, and Xiaohong Li. 2019. An Empirical Study towards Characterizing Deep Learning Development and Deployment across Different Frameworks and Platforms. In *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering (ASE '19)*. 810–822.

[13] Qing Guo, Xiaofei Xie, Felix Juefei-Xu, Lei Ma, Zhongguo Li, Wanli Xue, Wei Feng, and Yang Liu. 2019. SPARK: Spatial-aware Online Incremental Attack Against Visual Tracking. arXiv:cs.CV/1910.08681

[14] J. A. Hanley and B. J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143, 1 (April 1982), 29–36. http://www.ncbi.nlm.nih.gov/pubmed/7063747

[15] Dan Hendrycks and Kevin Gimpel. 2016. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *CoRR* abs/1610.02136 (2016). arXiv:1610.02136 http://arxiv.org/abs/1610.02136

[16] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2019. Deep Anomaly Detection with Outlier Exposure. In *International Conference on Learning Representations.* https://openreview.net/forum?id=HyxCxhRcY7

[17] Jinhan Kim, Robert Feldt, and Shin Yoo. 2019. Guiding Deep Learning System Testing Using Surprise Adequacy. In *Proceedings of the 41st International Conference on Software Engineering* (Montreal, Quebec, Canada) *(ICSE '19)*. 1039–1049.

[18] Alex Krizhevsky. 2009. *Learning multiple layers of features from tiny images.* Technical Report.

[19] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338. https://doi.org/10.1126/science.aab3050 arXiv:https://science.sciencemag.org/content/350/6266/1332.full.pdf

[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 6402–6413. http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf

[21] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. (2010). http://yann.lecun.com/exdb/mnist/

[22] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In *International Conference on Learning Representations.* https://openreview.net/forum?id=ryiAv2xAZ

[23] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 7167–7177. http://papers.nips.cc/paper/7947-a-simple-unified-framework-for-detecting-out-of-distribution-samples-and-adversarial-attacks.pdf

[24] Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *International Conference on Learning Representations.* https://openreview.net/forum?id=H1VGkIxRZ

[25] Lei Ma, Felix Juefei-Xu, Jiyuan Sun, Chunyang Chen, Ting Su, Fuyuan Zhang, Minhui Xue, Bo Li, Li Li, Yang Liu, et al. 2018. DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems. *ASE*, 120–131.

[26] Lei Ma, Felix Juefei-Xu, Minhui Xue, Bo Li, Li Li, Yang Liu, and Jianjun Zhao. 2019. DeepCT: Tomographic Combinatorial Testing for Deep Learning Systems. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 614–618.

[27] L. Ma, F. Juefei-Xu, M. Xue, B. Li, L. Li, Y. Liu, and J. Zhao. 2019. DeepCT: Tomographic Combinatorial Testing for Deep Learning Systems. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 614–618.

[28] Lei Ma, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Felix Juefei-Xu, Chao Xie, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. [n.d.]. DeepMutation: Mutation Testing of Deep Learning Systems. In *29th IEEE International Symposium on Software Reliability Engineering (ISSRE), Memphis, USA, Oct. 15-18, 2018*. 100–111.

[29] G. Mclachlan. 1999. Mahalanobis Distance. *Resonance* 4 (06 1999), 20–26. https://doi.org/10.1007/BF02834632

[30] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2015. DeepFool: a simple and accurate method to fool deep neural networks. *CoRR* abs/1511.04599 (2015). arXiv:1511.04599 http://arxiv.org/abs/1511.04599

[31] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. 2020. Detecting Out-of-Distribution Inputs to Deep Generative Models Using Typicality. https://openreview.net/forum?id=r1lnxTEYPS

[32] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011.* http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[33] Augustus Odena, Catherine Olsson, David Andersen, and Ian J. Goodfellow. 2019. TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing. In *ICML*. 4901–4911.

[34] Accompanied Anonymous Website of Deep Learning Testing Calls for Out-Of-Distribution Awareness. 2020. *This Work's Website.* https://sites.google.com/view/oodtesting/home

[35] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. *CoRR* abs/1705.06640 (2017). arXiv:1705.06640 http://arxiv.org/abs/1705.06640

[36] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. 2019. Likelihood Ratios for Out-of-Distribution Detection. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 14707–14718. http://papers.nips.cc/paper/9611-likelihood-ratios-for-out-of-distribution-detection.pdf

[37] Ryne Roady, Tyler L. Hayes, Ronald Kemker, Ayesha Gonzales, and Christopher Kanan. 2019. Are Out-of-Distribution Detection Methods Effective on Large-Scale Datasets? *CoRR* abs/1910.14034 (2019). arXiv:1910.14034 http://arxiv.org/abs/1910.14034

[38] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. 2017. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. *CoRR* abs/1701.05517 (2017). arXiv:1701.05517 http://arxiv.org/abs/1701.05517

[39] Vikash Sehwag, Arjun Nitin Bhagoji, Liwei Song, Chawin Sitawarin, Daniel Cullina, Mung Chiang, and Prateek Mittal. 2019. Analyzing the Robustness of Open-World Machine Learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security* (London, United Kingdom) *(AISec'19)*. Association for Computing Machinery, New York, NY, USA, 105–116. https://doi.org/10.1145/3338501.3357372

[40] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. 2020. Input Complexity and Out-of-distribution Detection with Likelihood-based Generative Models. In *International Conference on Learning Representations.* https://openreview.net/forum?id=SyxIWpVYvr

[41] Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. 2020. Stealthy and Efficient Adversarial Attacks against Deep Reinforcement Learning. In *AAAI*. 5883–5891.

[42] Youcheng Sun, Xiaowei Huang, and Daniel Kroening. 2018. Testing Deep Neural Networks. *arXiv preprint arXiv:1803.04792* (2018).

[43] Youcheng Sun, Min Wu, Wenjie Ruan, Xiaowei Huang, Marta Kwiatkowska, and Daniel Kroening. 2018. Concolic Testing for Deep Neural Networks. In *Automated Software Engineering (ASE)*. ACM, 109–119.

[44] Engkarat Techapanurak and Takayuki Okatani. 2019. Hyperparameter-Free Out-of-Distribution Detection Using Softmax of Scaled Cosine Similarity. *CoRR* abs/1905.10628 (2019). arXiv:1905.10628 http://arxiv.org/abs/1905.10628

[45] TechCrunch. 2019. *Didi Chuxing to launch self-driving rides in Shanghai and expand them beyond China by 2021.* https://techcrunch.com/2019/08/30/didi-chuxing-to-launch-self-driving-rides-in-shanghai-and-expand-them-beyond-china-by-2021/

[46] TechCrunch. 2019. *Waymo's robotaxi pilot surpassed 6,200 riders in itsfirst month in California.* https://techcrunch.com/2019/09/16/waymos-robotaxi-pilot-surpassed-6200-riders-in-its-first-month-in-california/

[47] TechCrunch. 2020. *Nearly 70% of US smart speaker owners use Amazon Echo devices.* https://techcrunch.com/2020/02/10/nearly-70-of-u-s-smart-speaker-owners-use-amazon-echo-devices/

[48] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2017. DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars. *CoRR* abs/1708.08559 (2017). arXiv:1708.08559 http://arxiv.org/abs/1708.08559

[49] New York Times. 2018. *Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam.* https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html

[50] New York Times. 2020. *Tesla Autopilot System Found Probably at Fault in 2018 Crash.* https://www.nytimes.com/2020/02/25/business/tesla-autopilot-ntsb.html

[51] Sachin Vernekar, Ashish Gaurav, Taylor Denouden, Buu Phan, Vahdat Abdelzad, Rick Salay, and Krzysztof Czarnecki. 2019. Analysis of Confident-Classifiers for Out-of-distribution Detection. *CoRR* abs/1904.12220 (2019). arXiv:1904.12220 http://arxiv.org/abs/1904.12220

[52] Run Wang, Felix Juefei-Xu, Qing Guo, Yihao Huang, Xiaofei Xie, Lei Ma, and Yang Liu. 2019. Amora: Black-box Adversarial Morphing Attack. arXiv:cs.CV/1912.03829

[53] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.* arXiv:cs.LG/cs.LG/1708.07747

[54] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. 2016. SUN Database: Exploring a Large Collection of Scene Categories. *Int. J. Comput. Vision* 119, 1 (Aug. 2016), 3–22. https://doi.org/10.1007/s11263-014-0748-y

[55] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. 2019. DeepHunter: A Coverage-Guided Fuzz Testing Framework for Deep Neural Networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis* (Beijing, China) *(ISSTA 2019)*. Association for Computing Machinery, New York, NY, USA, 146–157. https://doi.org/10.1145/3293882.3330579

[56] Xiaofei Xie, Lei Ma, Haijun Wang, Yuekang Li, Yang Liu, and Xiaohong Li. 2019. Diffchaser: Detecting disagreements for deep neural networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence.* AAAI Press, 5772–5778.

[57] J. M. Zhang, M. Harman, L. Ma, and Y. Liu. 2020. Machine Learning Testing: Survey, Landscapes and Horizons. *IEEE Transactions on Software Engineering* (2020). https://doi.org/10.1109/TSE.2019.2962027

[58] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. 2018. DeepRoad: GAN-based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE 2018).* 11.

[59] Xiyue Zhang, Xiaofei Xie, Lei Ma, Xiaoning Du, Qiang Hu, Yang Liu, Jianjun Zhao, and Meng Sun. 2020. Towards Characterizing Adversarial Defects of Deep Learning Software from the Lens of Uncertainty.