

Towards Guaranteeing the Reliability and Security of Deep Learning Systems

Zhao, Jianjun

Department of Advanced Information Technology, Faculty of Information Science and Electrical Engineering, Kyushu University : Professor

<https://hdl.handle.net/2324/1937146>

出版情報：九州大学大学院システム情報科学府・研究院先端サマーセミナー．2018，2018-08-10
バージョン：
権利関係：



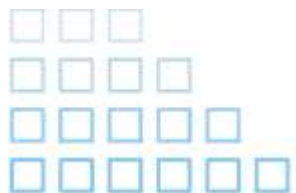
Towards Guaranteeing the Reliability and Security of Deep Learning Systems

(深層学習システムにおける信頼性と安全性保証の最前線)

Jianjun Zhao (趙建軍)

Pangu Research Group

Kyushu University

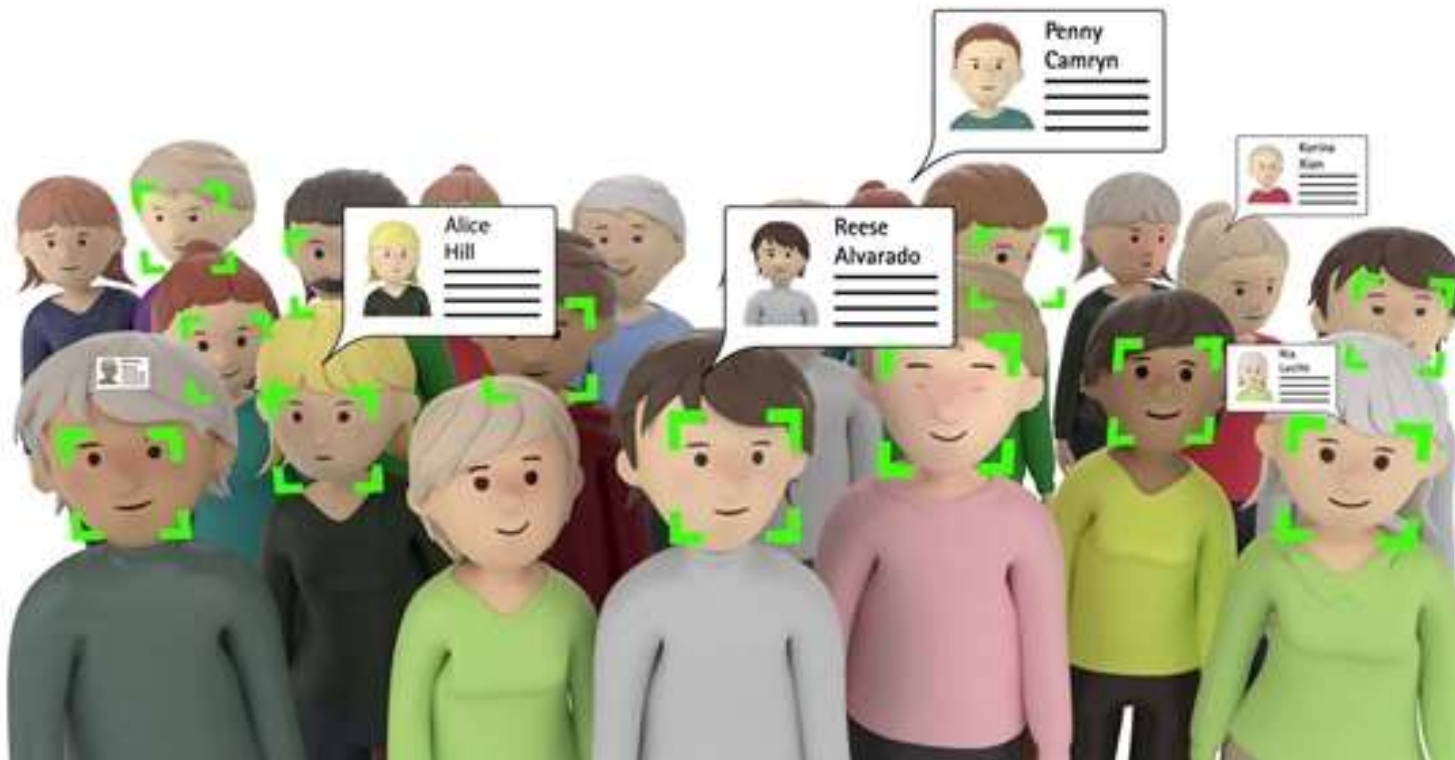


先端サマーセミナー2018 (2018.08.10)

Deep learning has matched human-level intelligence

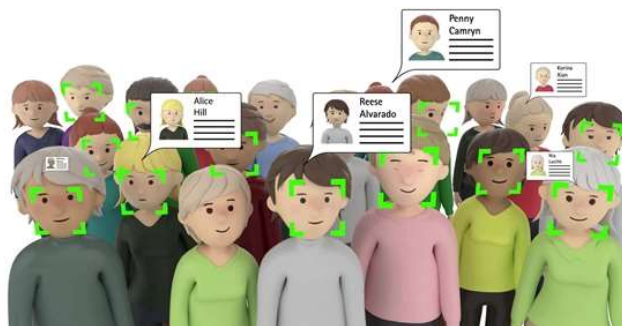
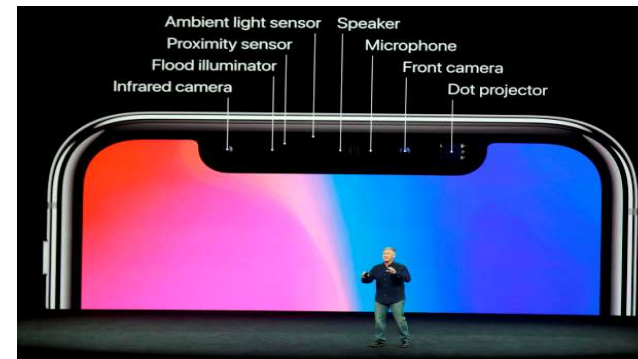


Deep learning has matched human-level intelligence



Deep learning (DL) has human-level intelligence!

- ❑ Image classification, speech recognition, machine translation, intrusion detection...
- ❑ Wide deployment in real-world systems

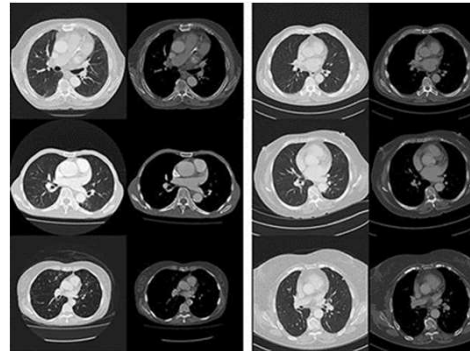


Deep learning is increasingly used in safety-critical systems

- ❑ Deep learning reliability and security is crucial



Self-driving car



Medical diagnosis

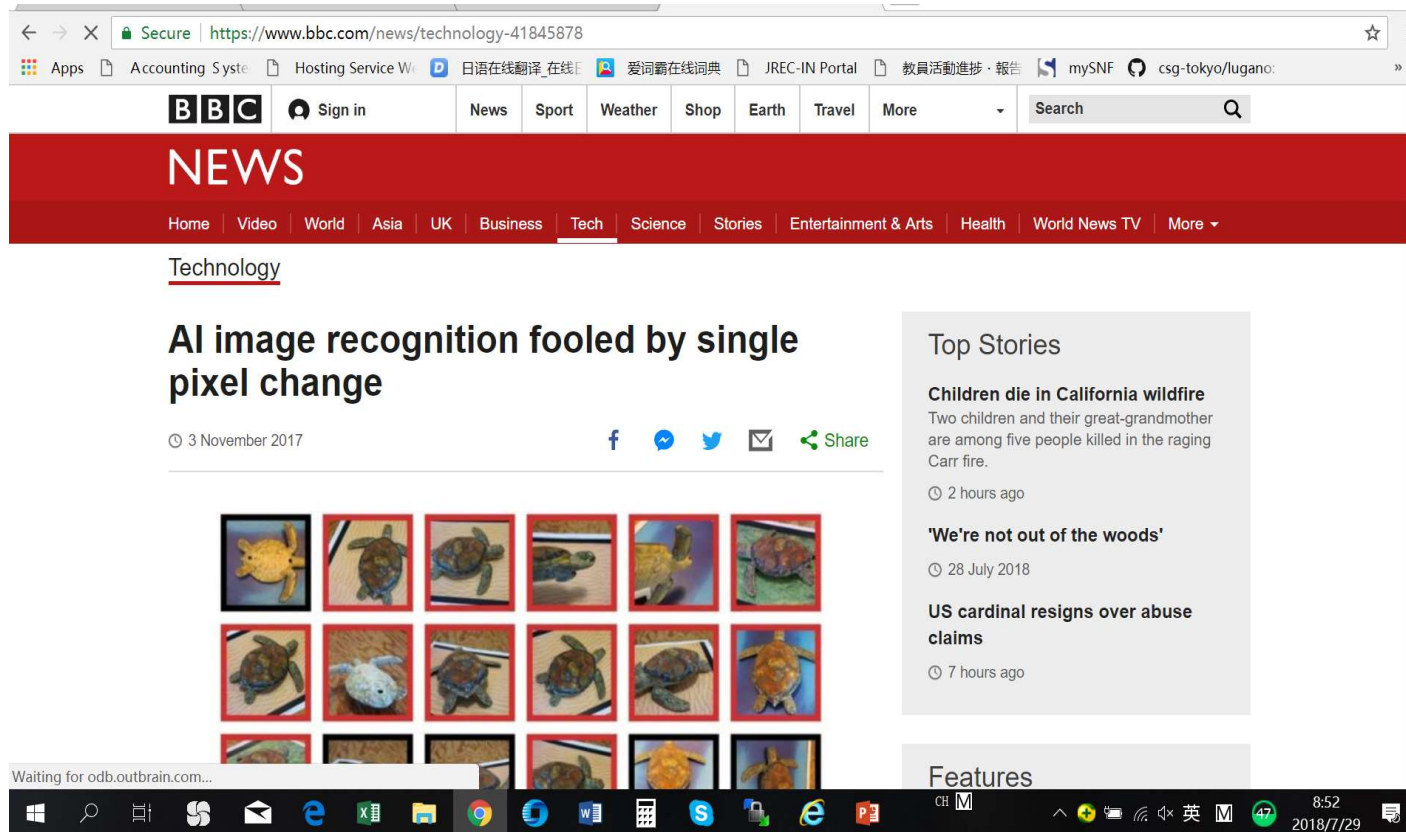


Malware detection

Major problems and critiques for DL systems

- ❑ Un-safe, e.g., lack of robustness (this talk)
- ❑ Hard to explain to human users
- ❑ Ethics, trustworthiness, accountability, etc.

Safety in Image Classification Networks



Safety in Natural Language Processing Networks

MOTHERBOARD

ARTIFICIAL INTELLIGENCE | By Jordan Pearson | Apr 28 2017, 10:00pm

AI Can Be Fooled With One Misspelled Word

When artificial intelligence is dumb.

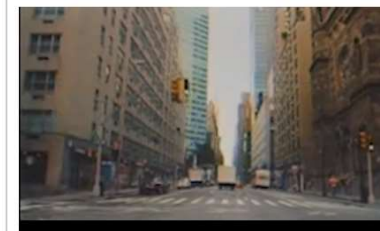
SHARE



TWEET



ADVERTISEMENT



Safety in Voice Recognition Networks

[Log in](#) | [Sign up](#) | [Forums](#)[Serverless](#) | [MP](#) | [CLL](#) | [Events](#) | [Whitepapers](#) | [The Next Platform](#)



The Register
Biting the hand that feeds IT



 [DATA CENTRE](#) [SOFTWARE](#) [SECURITY](#) [DEVOPS](#) [BUSINESS](#) [PERSONAL TECH](#) [SCIENCE](#) [EMERGENT TECH](#) [BOOTNOTES](#) [LECTURES](#) 

Security

Drowning Dalek commands Siri in voice-rec hack attack

Boffins embed barely-audible-to-humans commands inside vids to fool virtual assistants

By [Darren Pauli](#) 11 Jul 2016 at 07:48 40  SHARE ▼

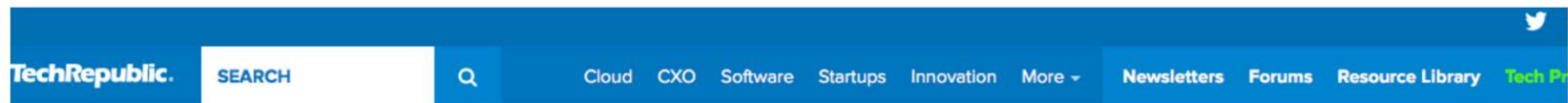
University boffins have brewed one of the most complex mechanisms for loading malware onto phones by way of surreptitious Google Now and Siri voice commands hidden in YouTube videos.

For the attack to work, phones need to be in a state where they can receive voice commands - a feature often left unlocked - and close enough to the audio source for the instructions to be understood.

There's also the trouble of masking the phone's beep which is an acknowledgment that voice commands have been received and could

Ad closed by Google

Safety in Security Systems



ARTIFICIAL INTELLIGENCE

AI vs AI: New algorithm automatically bypasses your best cybersecurity defenses

Researchers have created an AI that tweaks malware code, and it easily bypassed an anti-malware AI undetected. Is machine learning ready to face down cybersecurity threats?

By Brandon Vigliarolo | August 2, 2017, 12:25 PM PST

Unreliable deep learning contributed to Tesla fatal crash



Tesla autopilot failed to recognize a white truck against bright sky leading to fatal crash

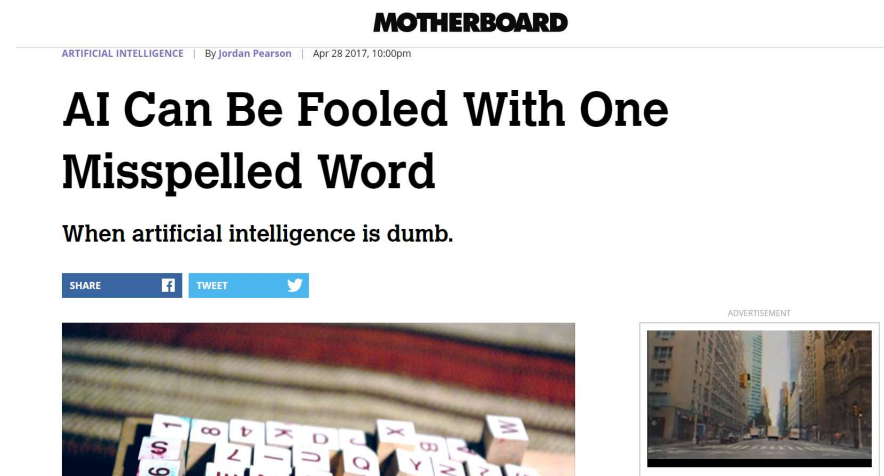
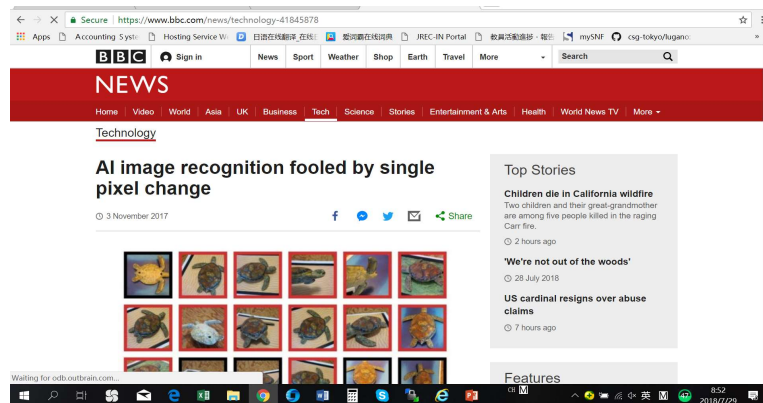
Unreliable deep learning contributed to Uber fatal crash



Uber self-driving test car was involved in a fatal collision
on March 19th , 2018

Challenge:

How to guarantee the reliability and security of DL systems?



To design reliable systems, engineers typically engage in both testing and verification

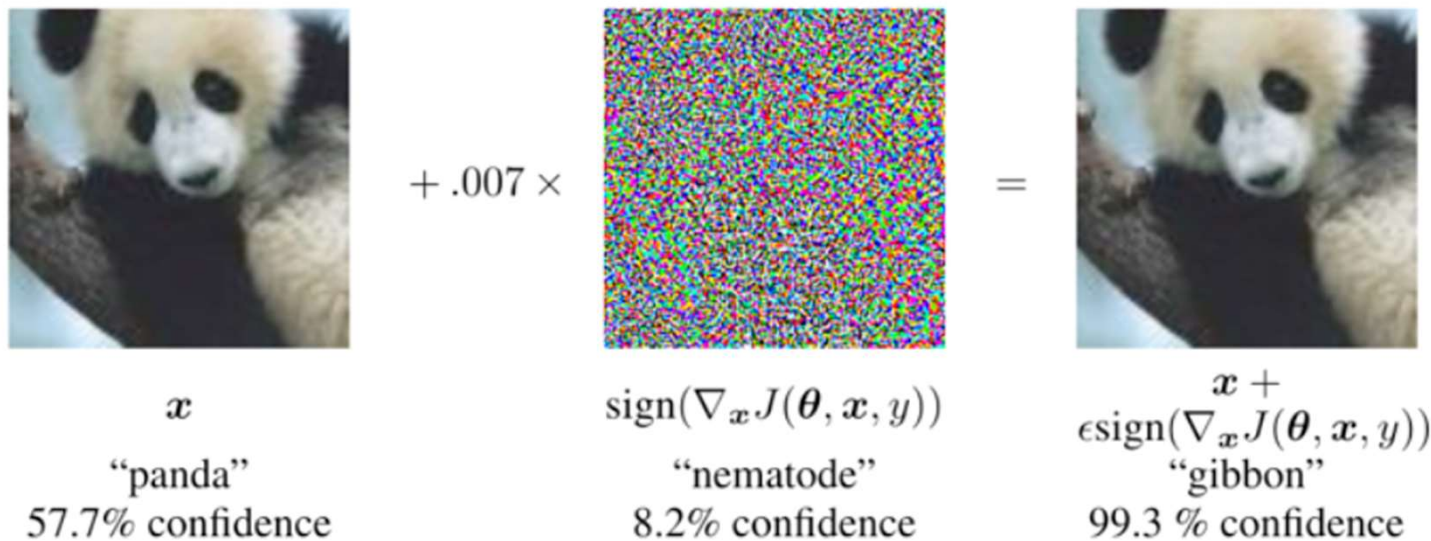
- ❑ **By testing:** we mean evaluating the system in several conditions and observing its behavior, watching for defects. (this talk focuses on testing)
- ❑ **By verification:** we mean producing a compelling argument that the system will not misbehave under a very broad range of circumstances.

Existing DL testing methods are seriously limited

- ❑ Common practice: measure accuracy on a test input set of randomly chosen data
- ❑ Problem 1: how good is the coverage of the test set?
 - DL decision logic is incredibly complex
 - More fundamentally, what is testing coverage metric for DL?
- ❑ Problem 2: it requires expensive labeling effort
 - Data in test set must be manually labelled
 - To enlarge the test set, we need to manually label more data

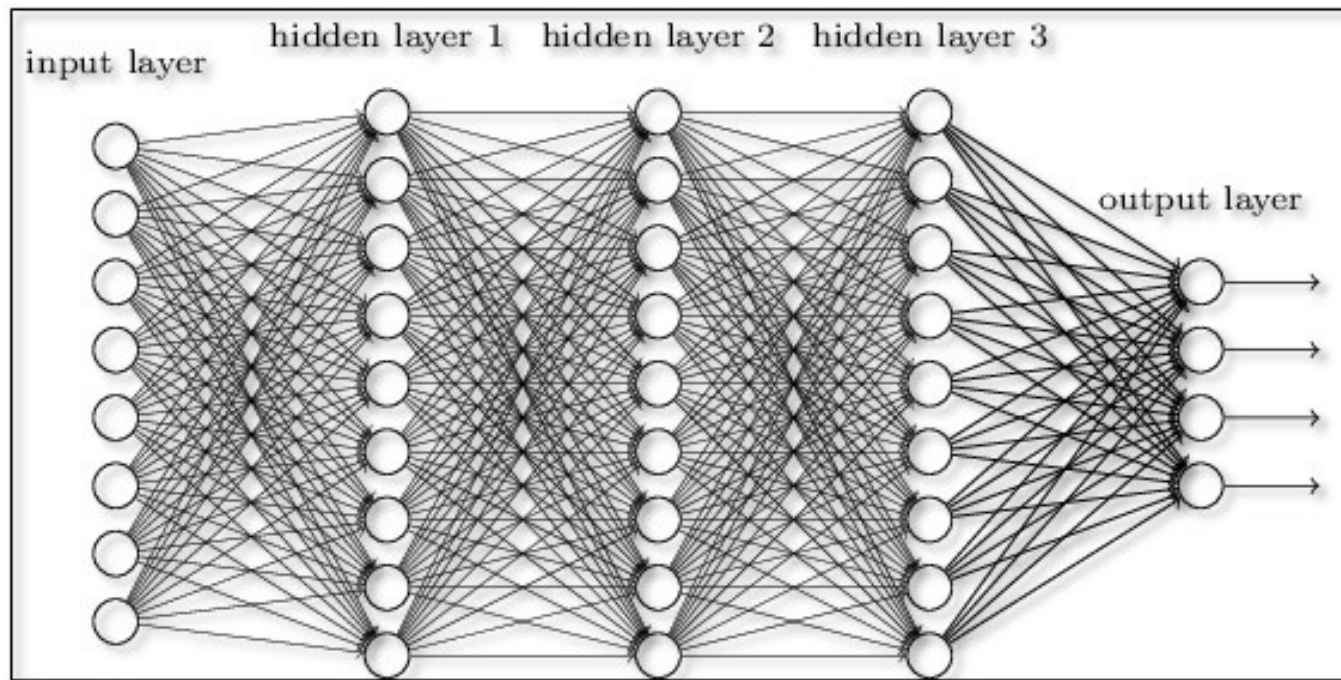
Existing DL testing methods are seriously limited (cont.)

- ❑ Adversarial testing: find corner-case inputs imperceptible to human but induce errors
 - **Problem 1:** how good is the coverage of the test set?
 - **Problem 2:** it requires expensive labeling effort



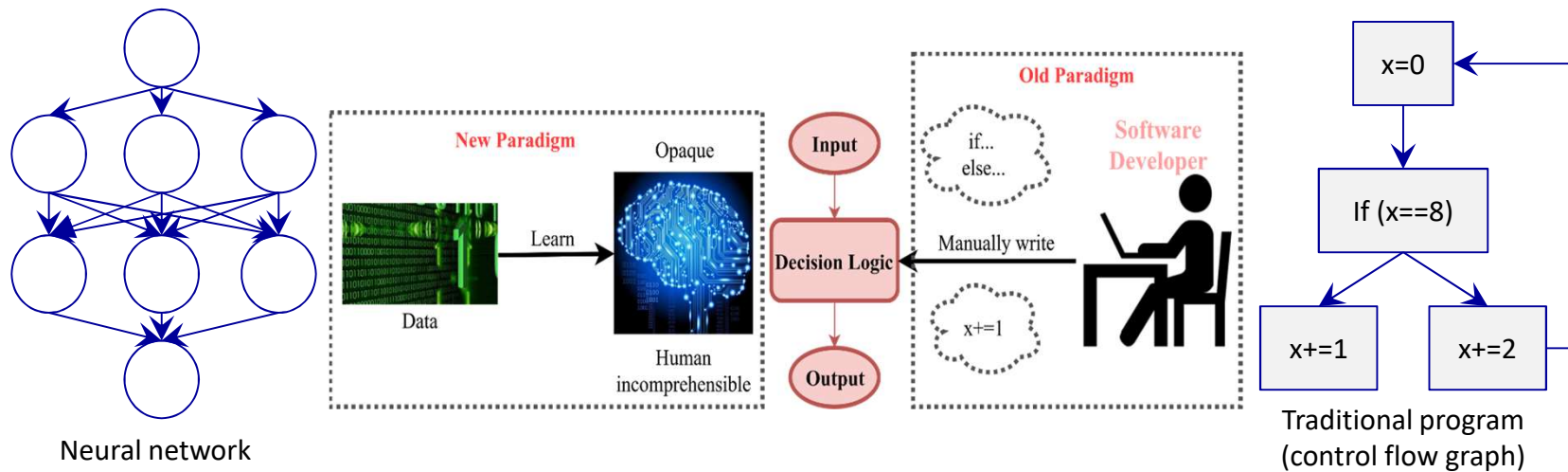
Traditional software testing techniques don't apply to DL

- ❑ DL decision logic is embedded in neurons and layers, not in code



Traditional software testing techniques don't apply to DL

- ❑ DL decision logic is embedded in neurons and layers, not in code

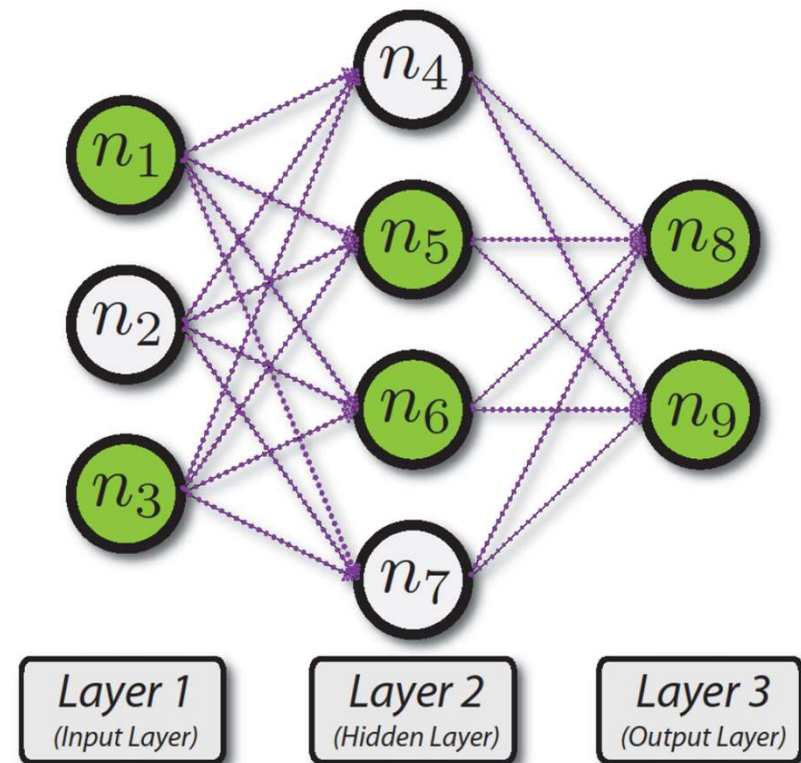


Novel testing techniques are needed for DL systems

- ❑ Problem 1: How good is the coverage of the test set?
 - *DeepGauge*: multi-granularity testing criteria for DL systems
 - Neuron-Level Coverage Criteria
 - Layer-Level Coverage Criteria
- ❑ Problem 2: it requires expensive labeling effort
 - *DeepMutation*: mutation testing of DL systems
 - Definition of mutation operators in neural network
 - Basic Operators and advanced mutation

Deep neural network architecture

- ❑ **A neural network** is a function $f(X) \rightarrow Y$
 - Trainable parameters (W_i) on each edge and nonlinear activation function at each neuron
 - DNN learns the weights during training
- ❑ **Inference:** Simply propagates X through layers (fast)
- ❑ **Training:** Given training set (X, Y) , adjust W to minimize the prediction error (slow)



The rest of the talk

- ❑ *DeepGauge*: multi-granularity testing criteria for DL systems
 - Motivation
 - Multi-granularity testing criteria
 - Experimental results

- ❑ *DeepMutation*: mutation testing of DL systems
 - Motivation
 - Multi-granularity testing criteria
 - Experimental results

The rest of the talk

- ❑ *DeepGauge*: multi-granularity testing criteria for DL systems
 - Motivation
 - Multi-granularity testing criteria
 - Experimental results

- ❑ *DeepMutation*: mutation testing of DL systems
 - Motivation
 - Multi-granularity testing criteria
 - Experimental results

Background and challenge

- ❑ Current studies focus only on pursuing high accuracy of DL systems as a testing criterion
 - Measuring the software quality from DL output alone is superficial in the sense that fundamental understanding of the DL internal neuron activities and network behaviors is not touched upon.
 - Criterion solely based on DL output will rely heavily on how representative the test data are.
 - Any DL system that passes systematic testing should be able to withstand all types of adversarial attacks to some extent.
- ❑ What is still lacking is a systematic way of gauging the testing adequacy of given DL systems.

Contributions

- ❑ Our proposed criteria facilitate the understanding of DNNs as well as the test data quality from different levels and angles.
- ❑ We evaluate the usefulness of our coverage criteria as indicators to quantify defect detection ability of test data on DNNs, through generating new adversarial test data using 4 well-known adversarial data generation algorithms.
- ❑ The various criteria proposed behave differently on DNNs w.r.t. network complexity and dataset under analysis.

Multi-granularity testing criteria for DL systems

- ❑ **Neuron-level coverage criteria**
 - K-multisection neuron coverage
 - Neuron boundary coverage
 - Strong neuron activation coverage

- ❑ **Layer-level coverage criteria**
 - Top-k neuron coverage
 - Top-k neuron patterns

Experiment Setting

- To test the effectiveness of the criteria proposed , we did experiment on 2 datasets and 5 models with 3 adversarial algorithms.

Dataset	Dataset Description	DL Name	#Neuron	#Layer	Test Data for Eval.
MNIST	Hand written digits recog.	LeNet-1	52	7	Test orig.
		LeNet-4	148	8	FGSM/BIM/JSMA
		LeNet-5	268	9	
ImageNet	General image with 1K-class	VGG-19	16,168	25	Test orig.
		ResNet50	94,056	175	FGSM/BIM

Table 1: Details of datasets and DNN models



Fig. From MNIST Dataset

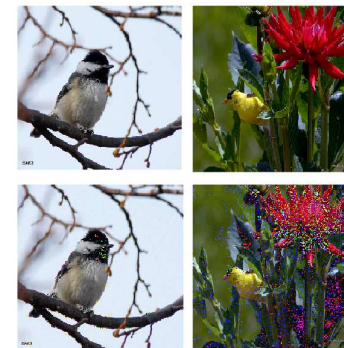
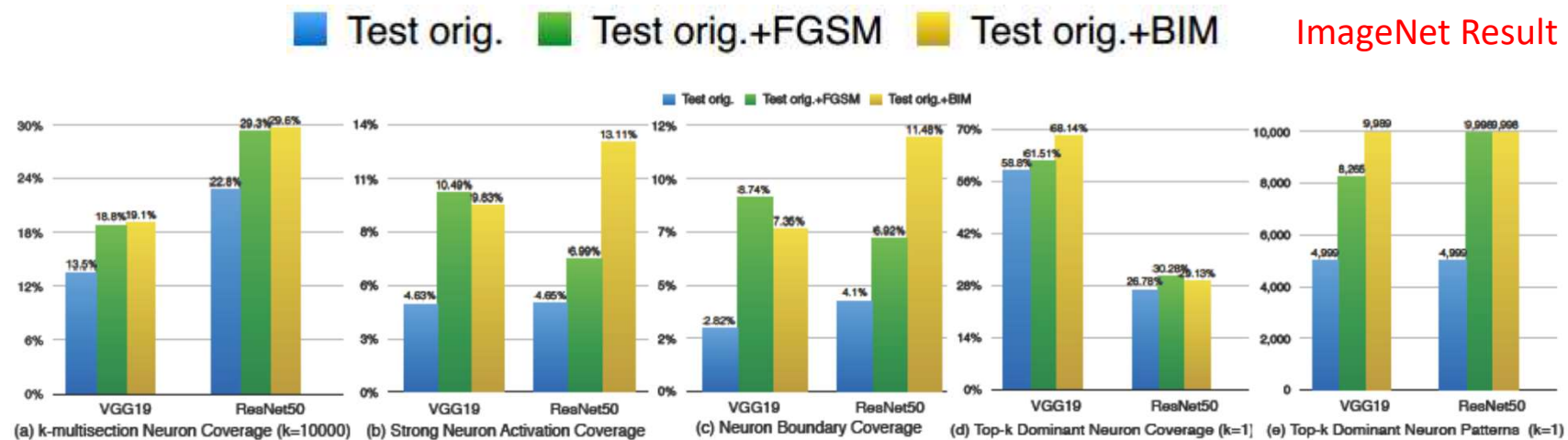
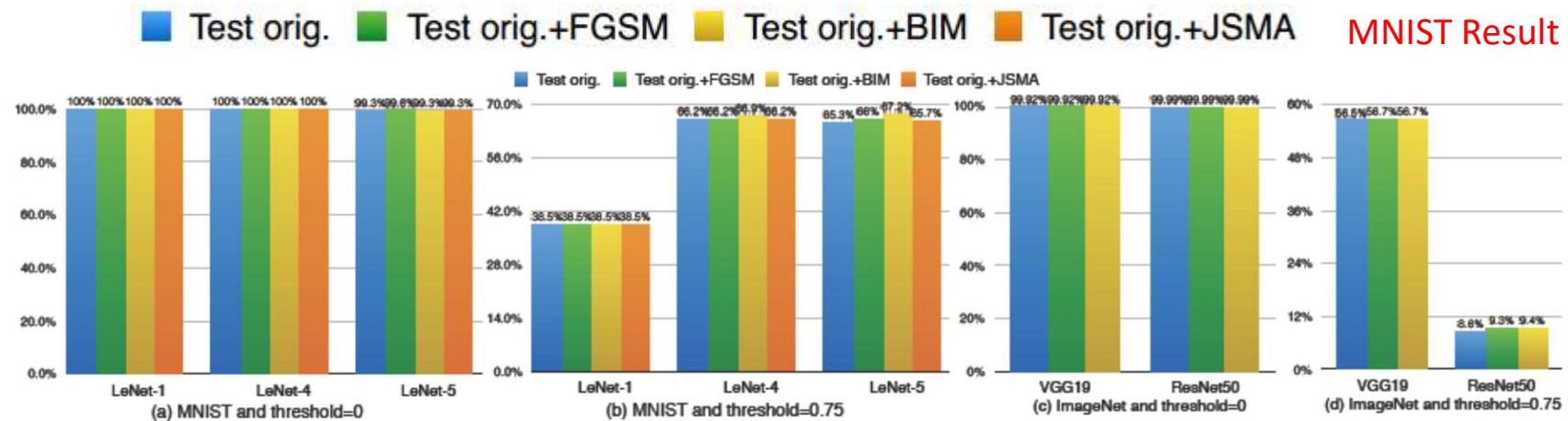


Fig. From ImageNet Dataset (ILSVRC2012)

Experimental Results

□ X_axis: Models Y_axis: Coverage



Experimental Results

- ❑ The original test data of MNIST and ImageNet cover both the DNNs major function region (see KMNC) as well as corner-case region (see NBC and SNAC)
- ❑ The test data generated by 4 studied adversarial techniques (combined with original test data) generally boost the coverage measured by our criteria.
- ❑ Test data (including the generated test data by adversarial) evaluated on both MNIST and ImageNet mostly obtain a higher k-multisection neuron coverage than the neuron boundary coverage and strong neuron activation coverage, revealing that the test data cover more of the major function region than the corner-case region of a DNN.
- ❑ For most of the evaluated configurations, we find that a higher strong neuron activation coverage is more achieved than its corresponding neuron boundary coverage.

The rest of the talk

- ❑ *DeepGauge*: multi-granularity testing criteria for DL systems
 - Motivation
 - Multi-granularity testing criteria
 - Experimental results

- ❑ *DeepMutation*: mutation testing of DL systems
 - Motivation
 - Multi-granularity testing criteria
 - Experimental results

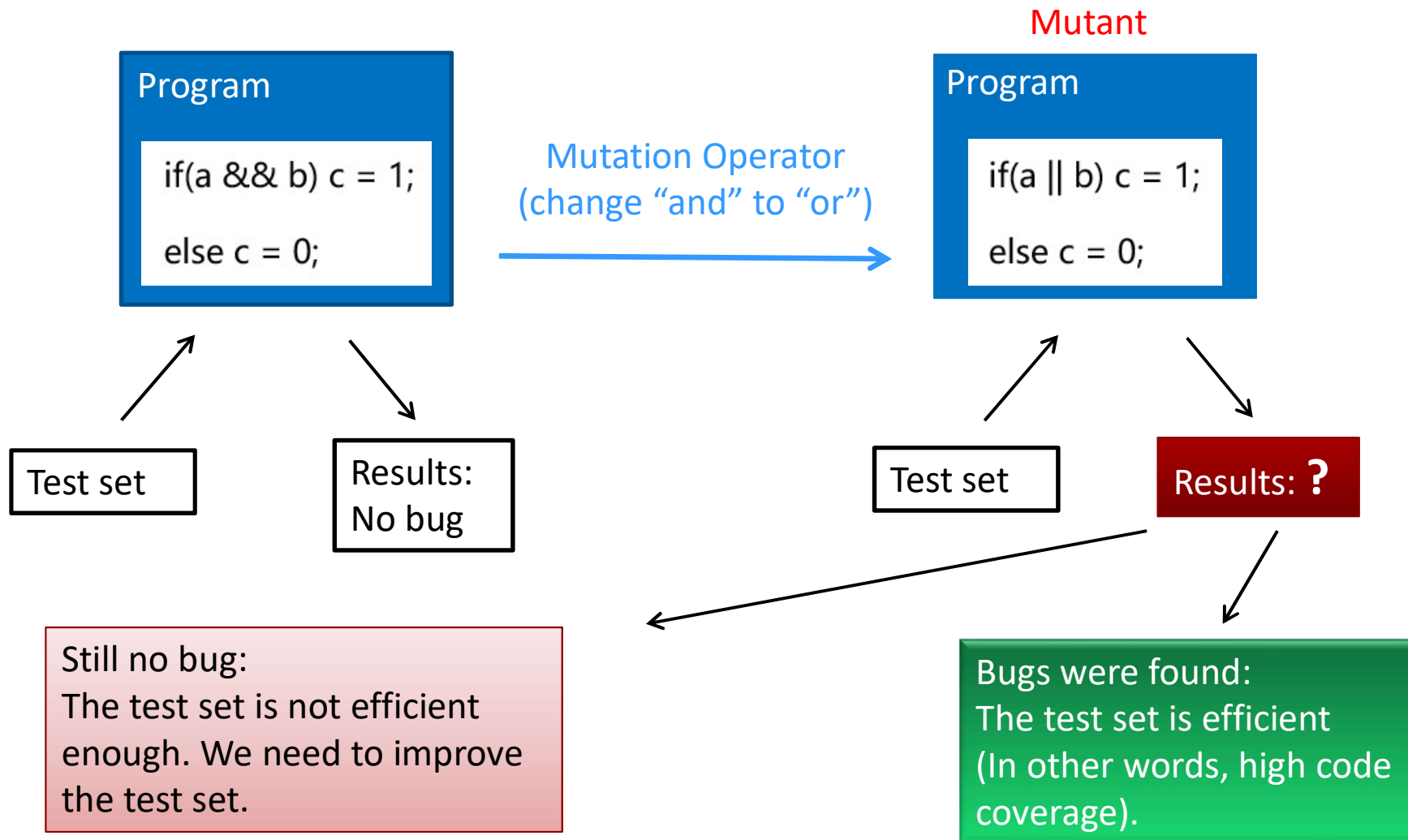
Background and challenge

- ❑ Current studies focus only on pursuing high accuracy of DL systems as a testing criterion
 - Measuring the software quality from DL output alone is superficial in the sense that fundamental understanding of the DL internal neuron activities and network behaviors is not touched upon.
 - Criterion solely based on DL output will rely heavily on how representative the test data are.
 - Any DL system that passes systematic testing should be able to withstand all types of adversarial attacks to some extent.
- ❑ What is still lacking is a systematic way of gauging the testing adequacy of given DL systems.

Contributions

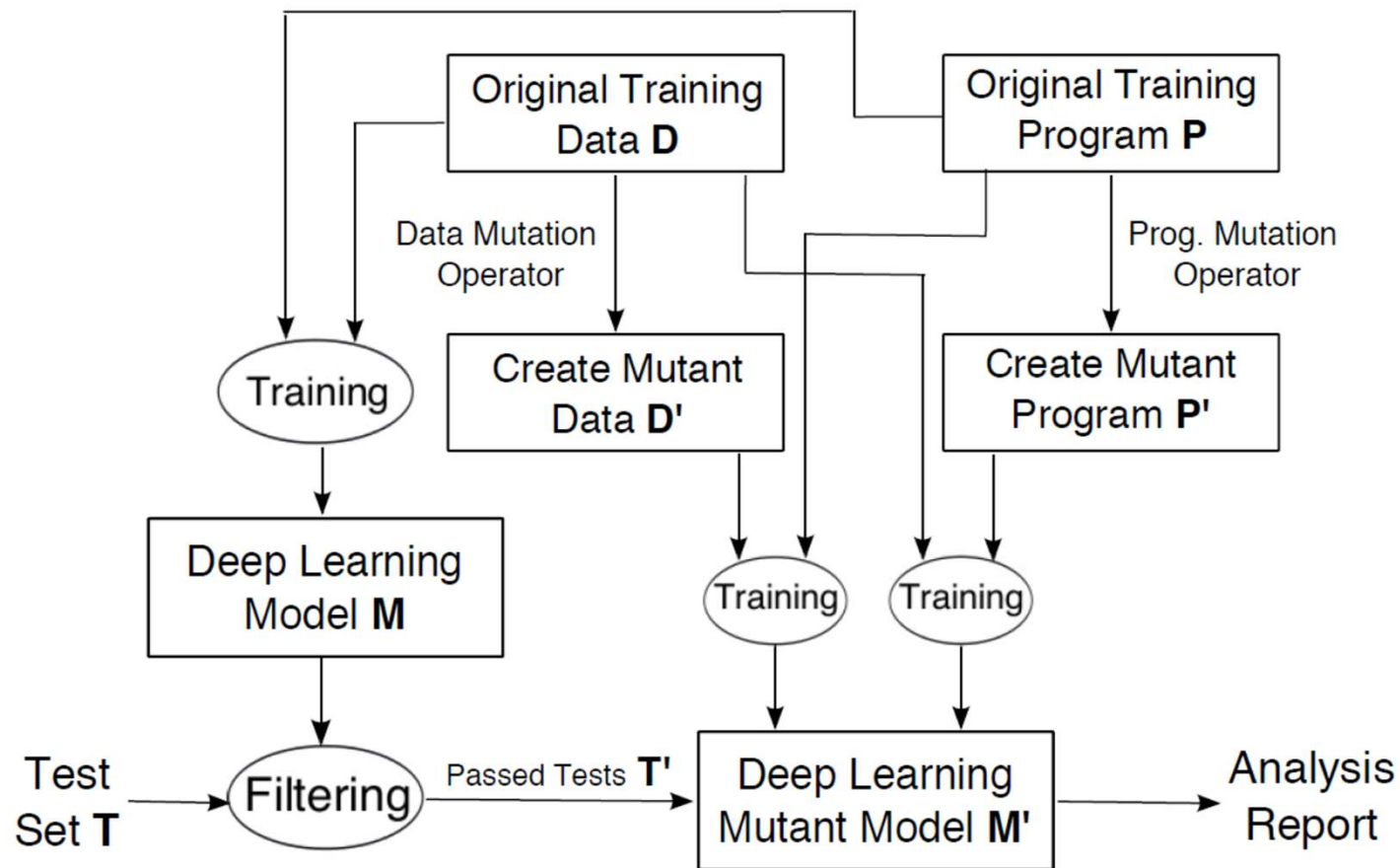
- ❑ **A mutation testing framework and workflow** specialized for DL systems, which enables the quality evaluation and weakness localization of test dataset.
- ❑ **The design of eight source-level mutation operators and eight mutation operators** that potentially or directly introduce potential faults into DL models.
- ❑ **Two DL-specific mutation testing metrics** to allow quantitative measurement for test quality.
- ❑ **Evaluation of the proposed mutation testing framework** on widely studied real-world data sets and DL models, to demonstrate the usefulness of the technique, which could also potentially facilitate the test set enhancement

Example of Software Mutation Testing



(Not in detailed here)

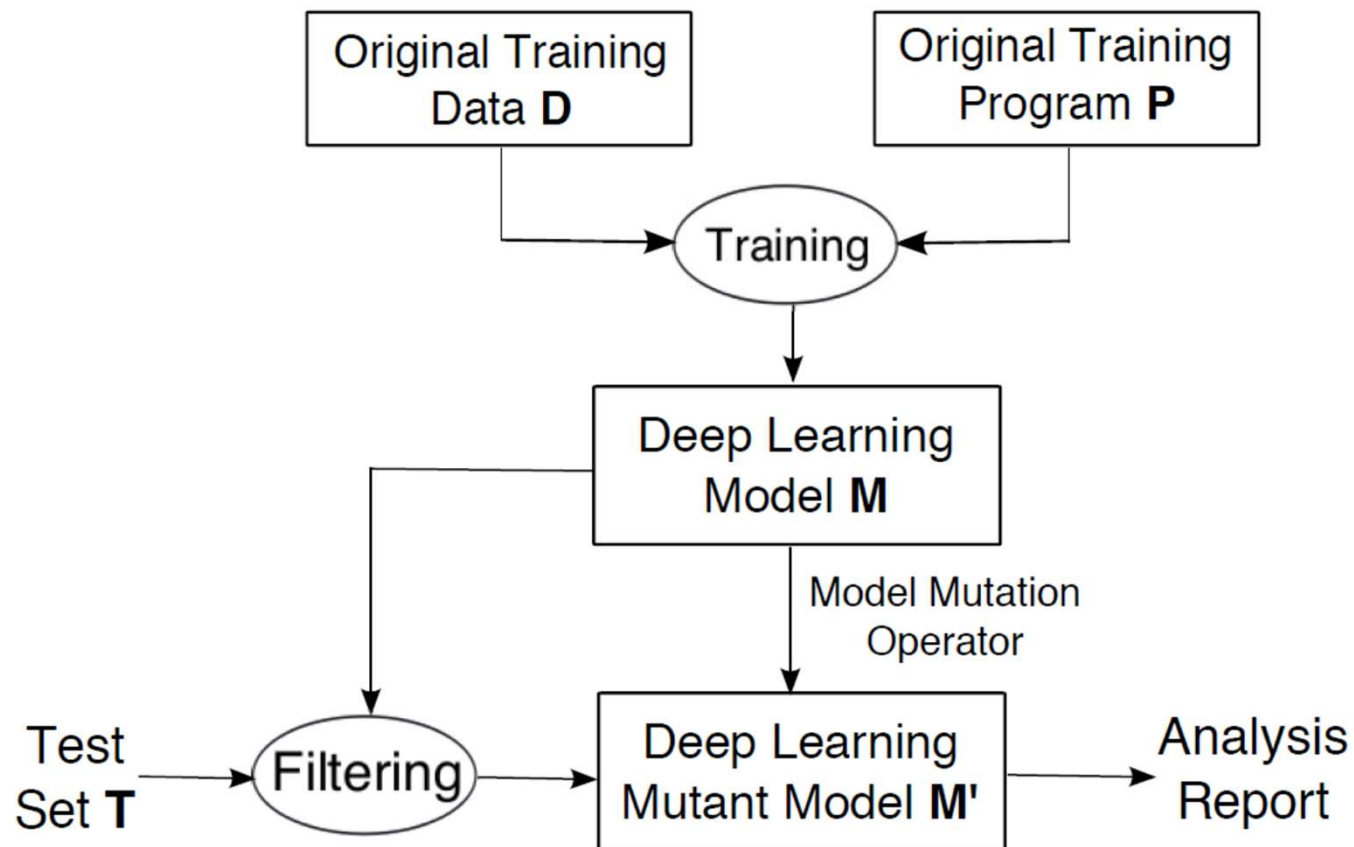
Source-level mutation testing workflow of DL systems



Source-level mutation testing operators for DL systems

Fault Type	Level	Target	Operation Description
Data Repetition (DR)	Global	Data	Duplicates training data
	Local		Duplicates specific type of data
Label Error (LE)	Global	Data	Falsify results (e.g., labels) of data
	Local		Falsify specific results of data
Data Missing (DM)	Global	Data	Remove selected data
	Local		Remove specific types of data
Data Shuffle (DF)	Global	Data	Shuffle selected training data
	Local		Shuffle specific types of data
Noise Perturb. (NP)	Global	Data	Add noise to training data
	Local		Add noise to specific type of data
Layer Removal (LR)	Global	Prog.	Remove a layer
Layer Addition (LA _s)	Global	Prog.	Add a layer
Act. Fun. Remov. (AFM _s)	Global	Prog.	Remove activation functions

Model-level mutation testing workflow of DL systems



Model-level mutation testing operators for DL systems

Mutation Operator	Level	Description
Gaussian Fuzzing (GF)	Weight	Fuzz weight by Gaussian Distribution
Weight Shuffling (WS)	Neuron	Shuffle selected weights
Neuron Effect Block. (NEB)	Neuron	Block a neuron effect on following layers
Neuron Activation Inverse (NAI)	Neuron	Change the activation status of a neuron
Neuron Switch (NS)	Neuron	Switch two neurons of the same layer
Layer Deactivation (LD)	Layer	Deactivate the effects of a layer
Layer Addition (LA_m)	Layer	Add a layer in neuron network
Act. Func. Remov. (AFR_m)	Layer	Remove neuron activation function

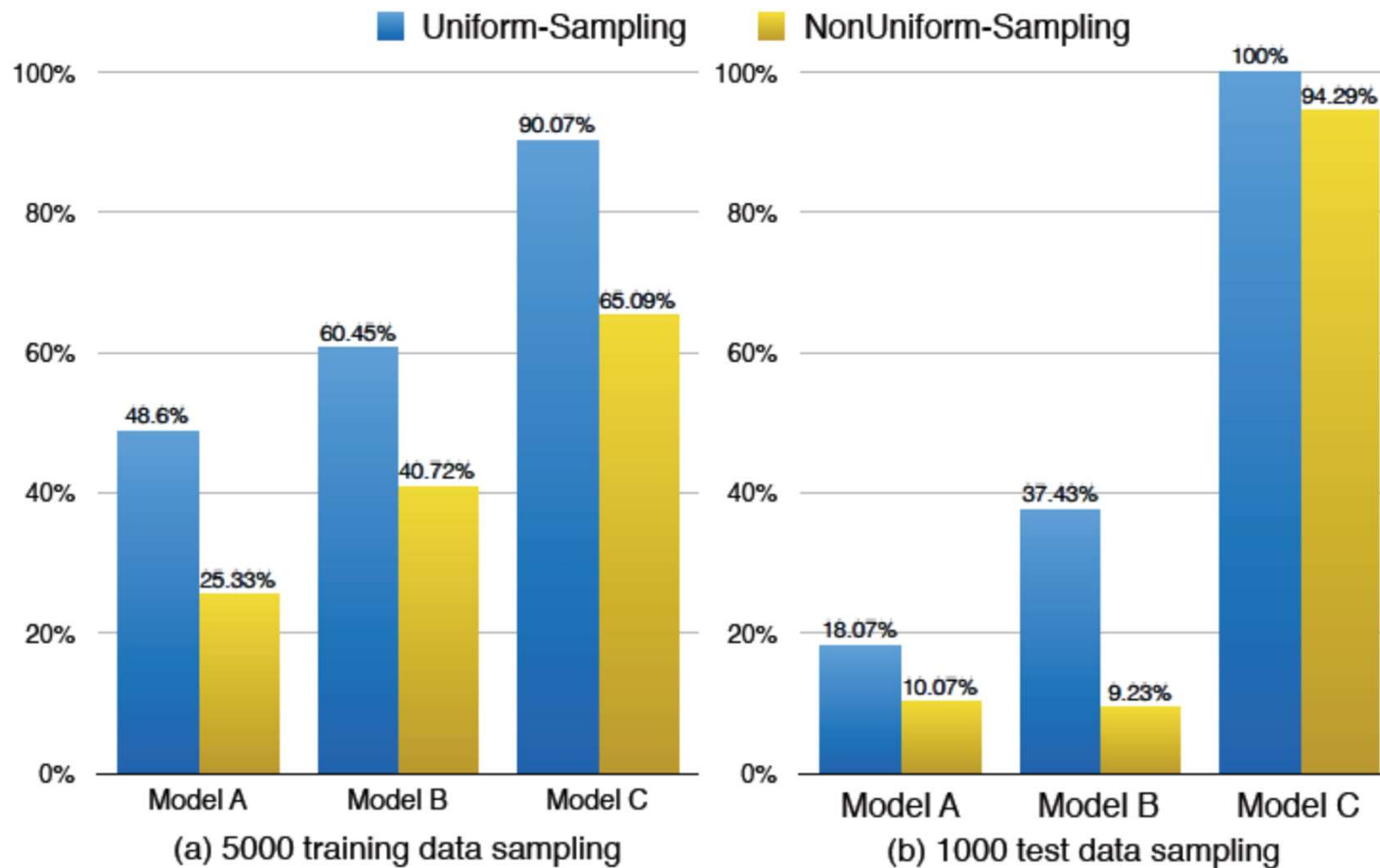
Experiment Setting

- The usefulness of the proposed mutation testing techniques is demonstrated on two public datasets, MNIST and CIFAR-10, with three DL models.

MNIST		CIFAR-10
A (LeNet5)[21]	B[35]	C[36]
Conv(6,5,5)+Relu	Conv(32,3,3)+Relu	Conv(64,3,3)+Relu
MaxPooling (2,2)	Conv(32,3,3)+Relu	Conv(64,3,3)+Relu
Conv(16,5,5)+Relu	MaxPooling(2,2)	MaxPooling(2,2)
MaxPooling(2,2)	Conv(64,3,3)+Relu	Conv(128,3,3)+Relu
Flatten()	Conv(64,3,3)+Relu	Conv(128,3,3)+Relu
FC(120)+Relu	MaxPooling(2,2)	MaxPooling(2,2)
FC(84)+Relu	Flatten()	Flatten()
FC(10)+Softmax	FC(200)+Relu	FC(256)+Relu
	FC(10)+Softmax	FC(256)+Relu
		FC(10)
#Train. Para. 107,786	694,402	1,147,978
Train. Acc. 97.4%	99.3%	97.1%
Test. Acc. 97.0%	98.7%	78.3%

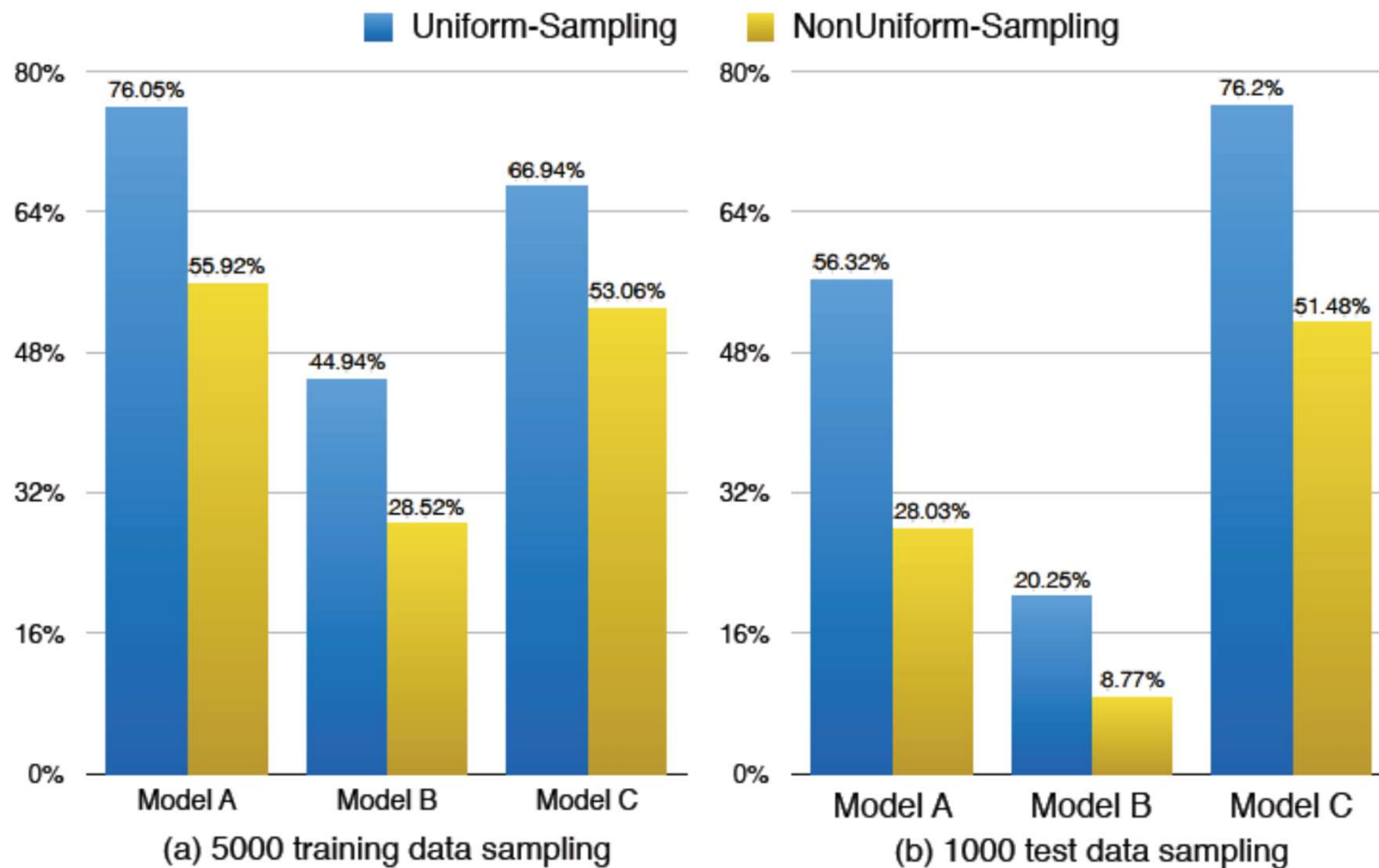
Experimental Results

- The averaged mutation score of source-level mutation testing



Experimental Results

- The averaged mutation score of model-level mutation testing



Exciting new research problems for DL/ML

- ❑ Build analysis tools for testing and verification of DL/ML systems
- ❑ Build better debugging support for opaque DL/ML systems
- ❑ Make DL/ML systems explainable

Thanks for listening
ご清聴ありがとうございました