University of Toronto
Department of Computer Science

# DeepMutation: Mutation Testing of Deep Learning Systems

Presented By
Abdul Kawsar Tushar
tushar21@cs.toronto.edu

# Authors

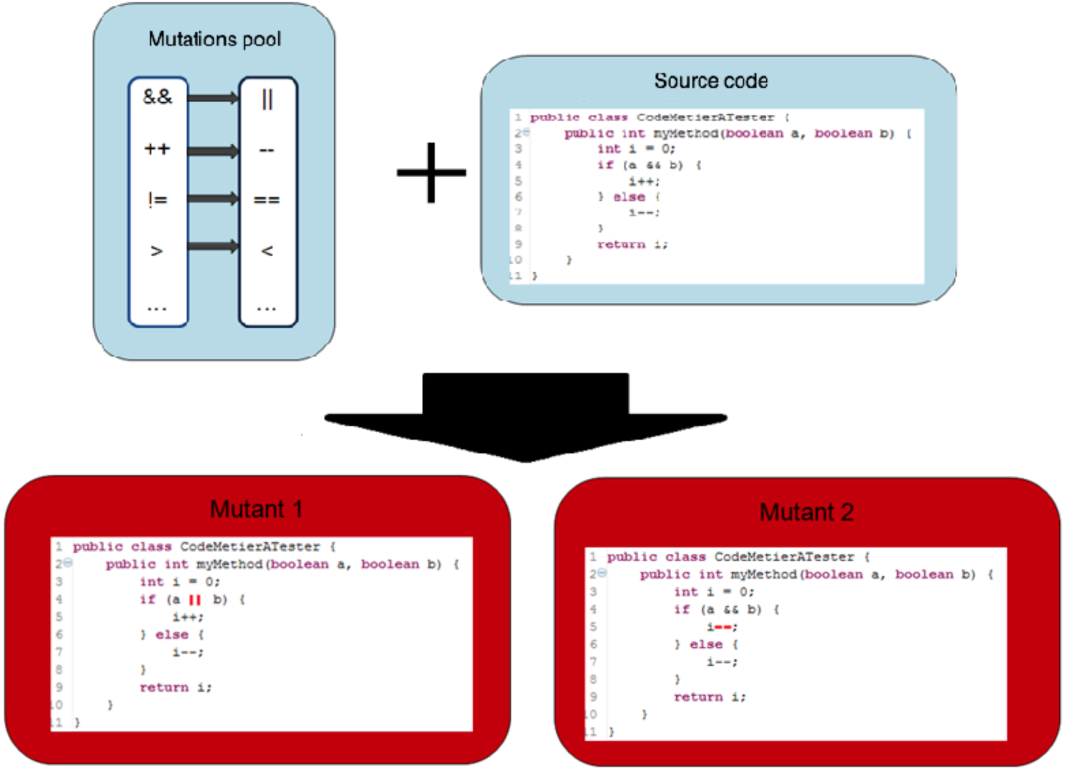

- Lei Ma
- Fuyuan Zhang
- Jiyuan Sun
- Minhui Xue
- Bo Li
- Felix Juefei-Xu
- Chao Xie
- Li Li
- Yang Liu
- Jianjun Zhao
- Yadong Wang

- Harbin Institute of Technology, China
- Nanyang Technological University, Singapore
- Kyushu University, Japan
- University of Illinois at Urbana–Champaign, USA
- Carnegie Mellon University, USA
- Monash University, Australia

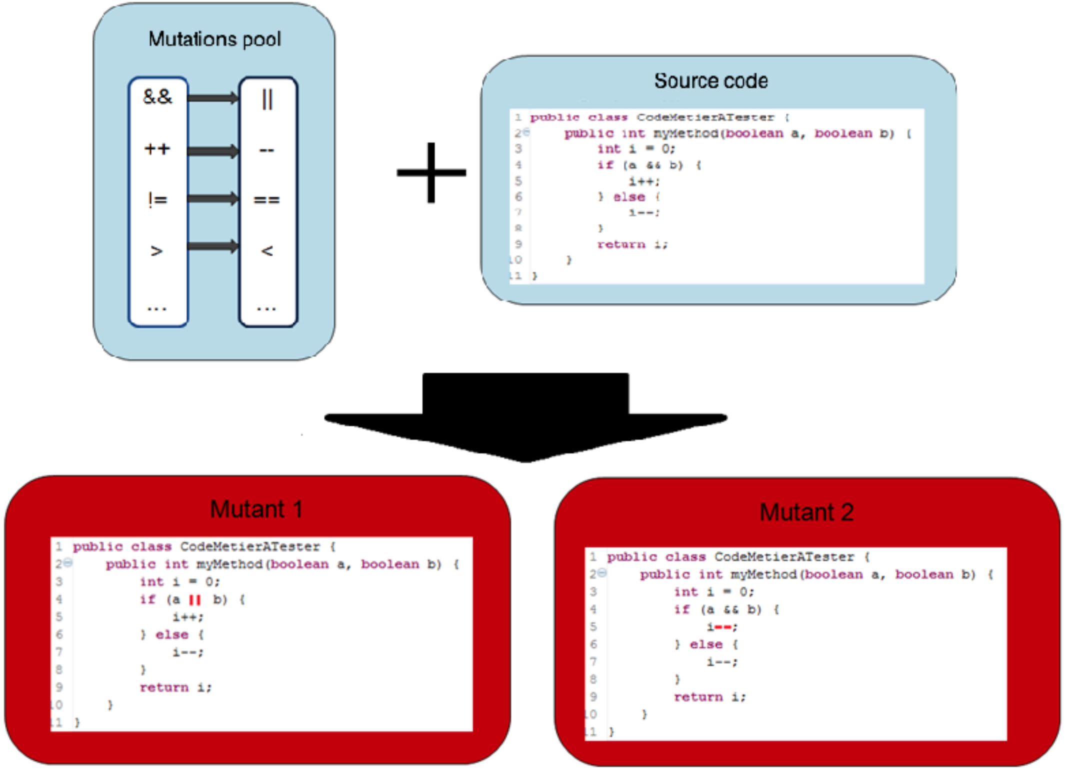
# Contents

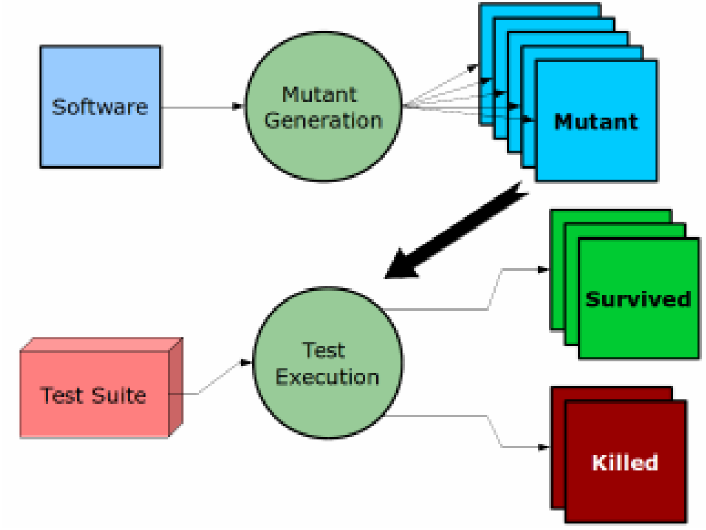
- What is mutation testing (MT)?
- MT in traditional software vs DL
- Proposed approach for MT
- Parameters used
- Performance
- Discussion points
- Related work

# Mutation Testing



# Mutation Testing



# Software System vs DL System

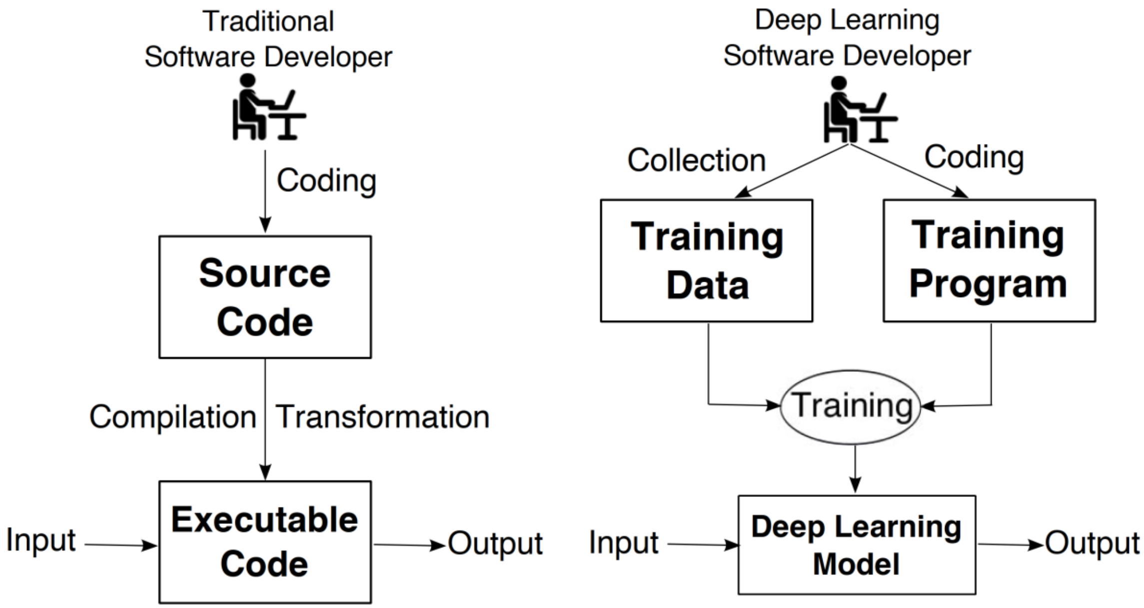



**Fig. 1:** A comparison of traditional and DL software development.

# General Mutation Testing

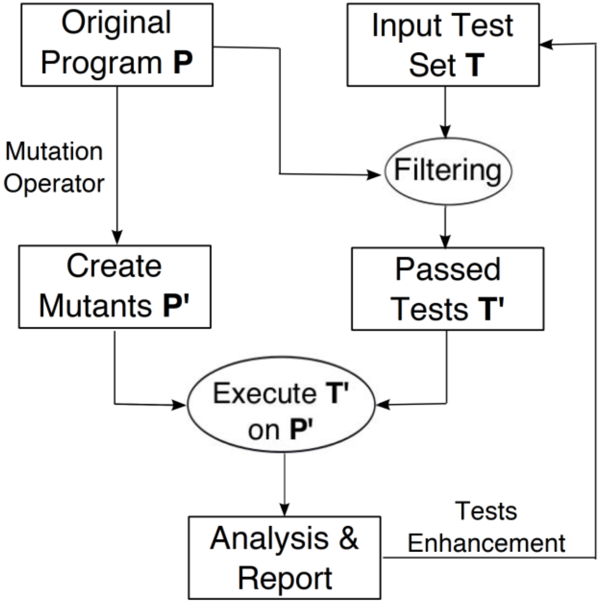



**Fig. 2:** Key process of general mutation testing.

# Proposed Mutation Testing - Source Level

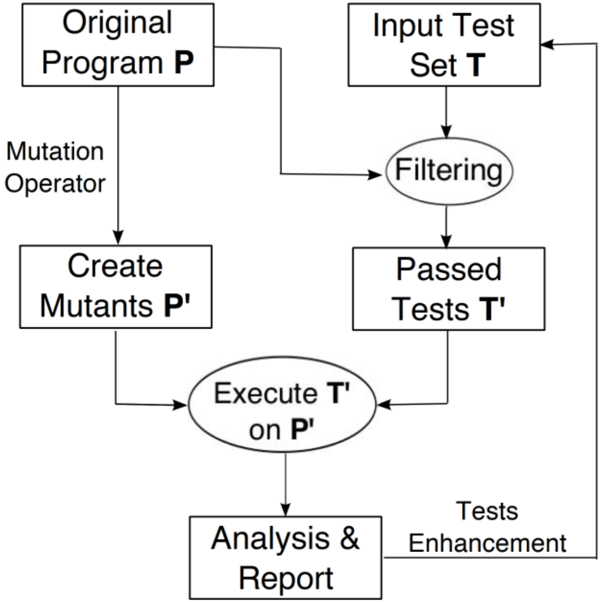



**Fig. 2:** Key process of general mutation testing.

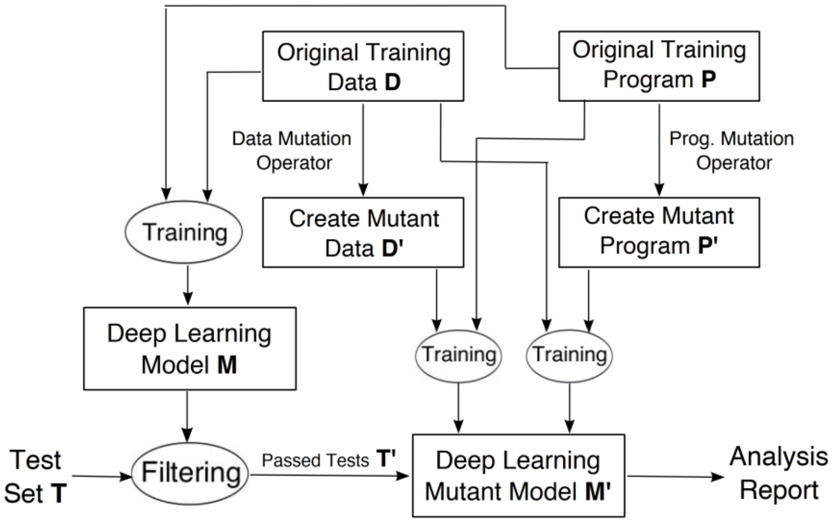



**Fig. 3:** Source-level mutation testing workflow of DL systems.

# Proposed Mutation Testing - Model Level

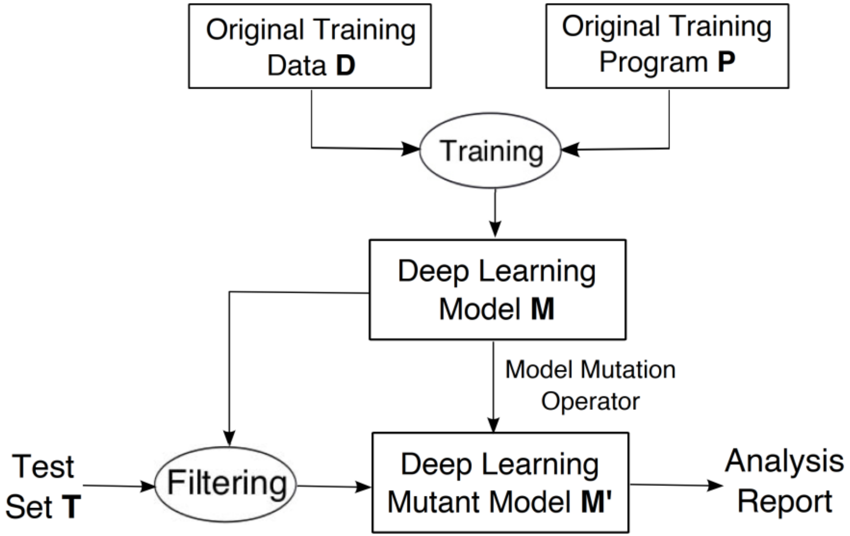



**Fig. 5:** The model level mutation testing workflow for DL systems.

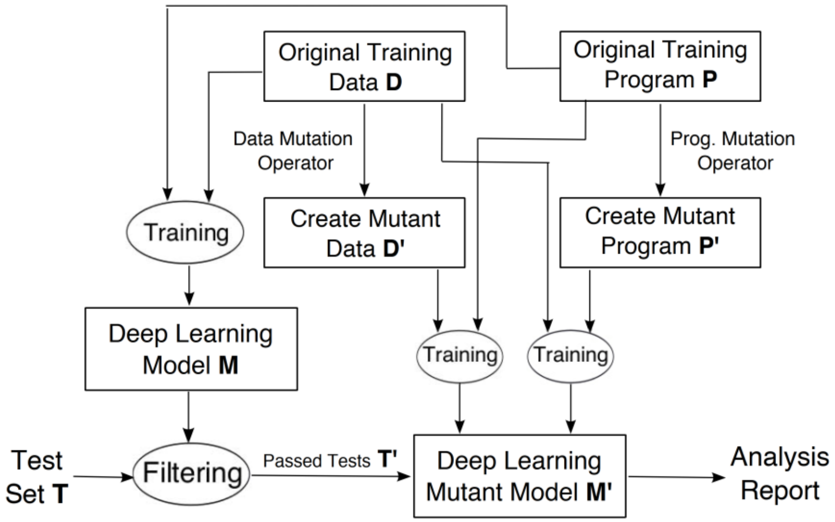



**Fig. 3:** Source-level mutation testing workflow of DL systems.

# Decision Boundary

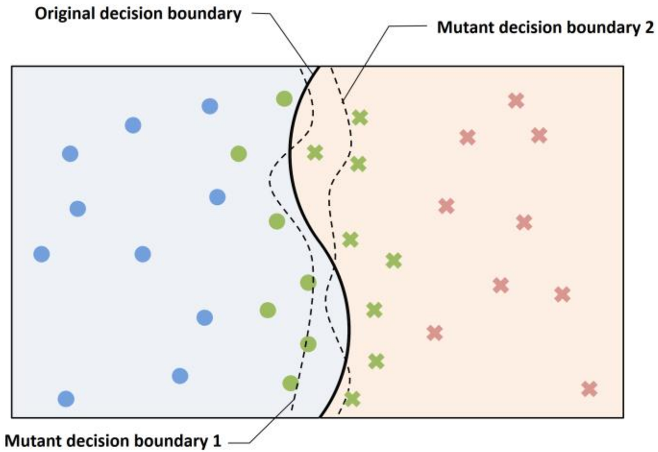



**Fig. 4:** Example of DL model and its two generated mutant models for binary classification with their decision boundaries. In the figure, some data scatter closer to the decision boundary (in green color). Our mutation testing metrics favor to identify the test data that locate in the sensitive region near the decision boundary.

# Source-level Operators

**TABLE I:** Source-level mutation testing operators for DL systems.

| Fault Type | Level | Target | Operation Description |
|---|---|---|---|
| Data Repetition (DR) | Global<br>Local | Data | Duplicates training data<br>Duplicates specific type of data |
| Label Error (LE) | Global<br>Local | Data | Falsify results (e.g., labels) of data<br>Falsify specific results of data |
| Data Missing (DM) | Global<br>Local | Data | Remove selected data<br>Remove specific types of data |
| Data Shuffle (DF) | Global<br>Local | Data | Shuffle selected training data<br>Shuffle specific types of data |
| Noise Perturb. (NP) | Global<br>Local | Data | Add noise to training data<br>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. ($AFR_s$) | Global | Prog. | Remove activation functions |

# Model-level Operators

**TABLE II:** 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 | Invert 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. Fun. Remov. ($AFR_m$) | Layer | Remove activation functions |

# Baseline Models

**TABLE III:** Evaluation subject datasets and DL models. Our selected subject datasets MNIST and CIFAR-10 are widely studied in previous work. We train the DNNs model with its corresponding original training data and training program. The obtained DL model refers to the original DL (*i.e.*, the DL model $M$ in Figure 3 and 5), which we use as the baseline in our evaluation. Each studied DL model structure and the obtained accuracy are summarized below.

| MNIST | | CIFAR-10 |
|---|---|---|
| A (LeNet5) [23] | B [38] | C [39] |
| 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) |

| | MNIST | | CIFAR-10 |
|---|---|---|---|
| | A (LeNet5) [23] | B [38] | C [39] |
| #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% |

# Settings

1. Test Data:
   30 pairs

**TABLE IV:** The controlled experiment data preparation settings.

| Controlled Data Set | MNIST/CIFAR-10 | | | |
|---|---|---|---|---|
| | Setting 1 | | Setting 2 | |
| | Group 1 | Group 2 | Group 1 | Group 2 |
| Source | Train. data | Train. data | Test data | Test data |
| Sampling | Uniform | Non-uniform | Uniform | Non-uniform |
| #Size | 5,000 | 5,000 | 1,000 | 1,000 |

2. Mutant Model:
   1. Source-level mutant: 10*2 and 20
   2. Model-level mutant: 50 and 50

# Source-level Mutant Model Generation

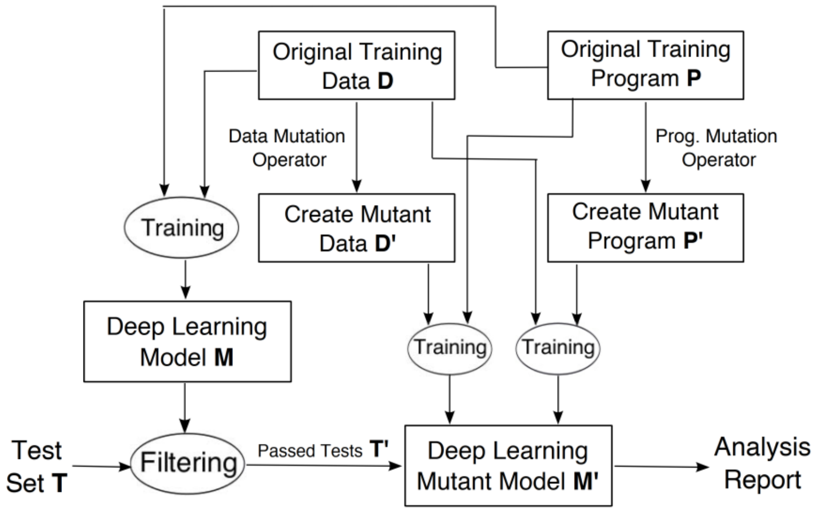



**Fig. 3:** Source-level mutation testing workflow of DL systems.

**TABLE I:** Source-level mutation testing operators for DL systems.

| Fault Type | Level | Target | Operation Description |
|---|---|---|---|
| Data Repetition (DR) | Global<br>Local | Data | Duplicates training data<br>Duplicates specific type of data |
| Label Error (LE) | Global<br>Local | Data | Falsify results (e.g., labels) of data<br>Falsify specific results of data |
| Data Missing (DM) | Global<br>Local | Data | Remove selected data<br>Remove specific types of data |
| Data Shuffle (DF) | Global<br>Local | Data | Shuffle selected training data<br>Shuffle specific types of data |
| Noise Perturb. (NP) | Global<br>Local | Data | Add noise to training data<br>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. ($AFR_s$) | Global | Prog. | Remove activation functions |

**TABLE IV:** The controlled experiment data preparation settings.

| Controlled Data Set | MNIST/CIFAR-10 | | | |
|---|---|---|---|---|
| | Setting 1 | | Setting 2 | |
| | Group 1 | Group 2 | Group 1 | Group 2 |
| Source | Train. data | Train. data | Test data | Test data |
| Sampling | Uniform | Non-uniform | Uniform | Non-uniform |
| #Size | 5,000 | 5,000 | 1,000 | 1,000 |

# Model-level Mutant Model Generation

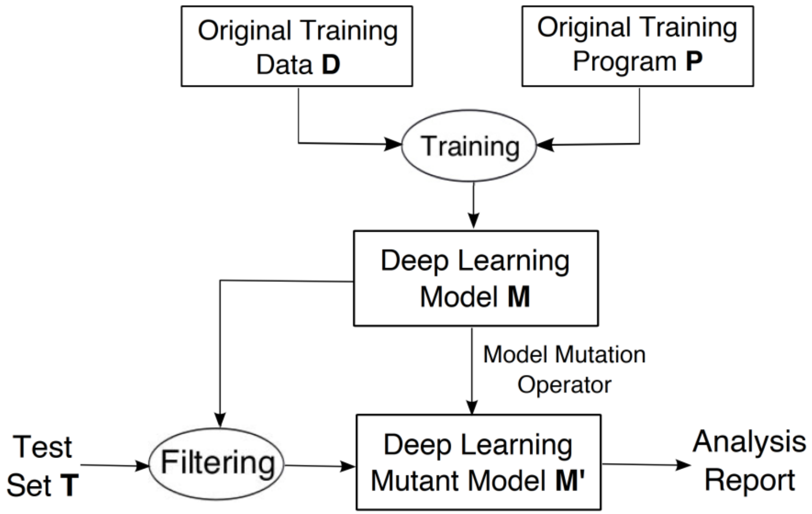



**Fig. 5:** The model level mutation testing workflow for DL systems.

**TABLE II:** 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 | Invert 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. Fun. Remov. ($AFR_m$) | Layer | Remove activation functions |

# Performance

# Metrics

$$\mathrm{MutationScore}(T', M') = \frac{\sum_{m' \in M'} |\mathrm{KilledClasses}(T', m')|}{|M'| \times |C|}$$

$$\mathrm{AveErrorRate}(T', M') = \frac{\sum_{m' \in M'} \mathrm{ErrorRate}(T', m')}{|M'|}$$

# Average Error Rate

**TABLE V:** The average error rate of controlled experiment data on the DL mutant models. We control the sampling method and data size to be the same, and let the data selection scope as the variable. The first group sample data from all classes of original passed test data, while the second group sample data from a single class.

| Model | Source Level (%) | | | | Model Level (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | 5000 train. | | 1000 test. | | 5000 train. | | 1000 test. | |
| Samp. | Uni. | Non. | Uni. | Non. | Uni. | Non. | Uni. | Non. |
| A | 2.43 | 0.13 | 0.23 | 0.17 | 4.55 | 4.30 | 4.38 | 4.06 |
| B | 0.49 | 0.28 | 0.66 | 0.21 | 1.67 | 1.56 | 1.55 | 1.47 |
| C | 3.84 | 2.99 | 17.20 | 13.44 | 9.11 | 7.34 | 11.48 | 9.00 |

# Average Error Rate

**TABLE V:** The average error rate of controlled experiment data on the DL mutant models. We control the sampling method and data size to be the same, and let the data selection scope as the variable. The first group sample data from all classes of original passed test data, while the second group sample data from a single class.

| Model | Source Level (%) | | | | Model Level (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | 5000 train. | | 1000 test. | | 5000 train. | | 1000 test. | |
| Samp. | Uni. | Non. | Uni. | Non. | Uni. | Non. | Uni. | Non. |
| A | 2.43 | 0.13 | 0.23 | 0.17 | 4.55 | 4.30 | 4.38 | 4.06 |
| B | 0.49 | 0.28 | 0.66 | 0.21 | 1.67 | 1.56 | 1.55 | 1.47 |
| C | 3.84 | 2.99 | 17.20 | 13.44 | 9.11 | 7.34 | 11.48 | 9.00 |

# Average Error Rate

**TABLE V:** The average error rate of controlled experiment data on the DL mutant models. We control the sampling method and data size to be the same, and let the data selection scope as the variable. The first group sample data from all classes of original passed test data, while the second group sample data from a single class.

| Model | Source Level (%) | | | | Model Level (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | 5000 train. | | 1000 test. | | 5000 train. | | 1000 test. | |
| Samp. | Uni. | Non. | Uni. | Non. | Uni. | Non. | Uni. | Non. |
| A | 2.43 | 0.13 | 0.23 | 0.17 | 4.55 | 4.30 | 4.38 | 4.06 |
| B | 0.49 | 0.28 | 0.66 | 0.21 | 1.67 | 1.56 | 1.55 | 1.47 |
| C | 3.84 | 2.99 | 17.20 | 13.44 | 9.11 | 7.34 | 11.48 | 9.00 |

# Average Error Rate

**TABLE V:** The average error rate of controlled experiment data on the DL mutant models. We control the sampling method and data size to be the same, and let the data selection scope as the variable. The first group sample data from all classes of original passed test data, while the second group sample data from a single class.

| Model | Source Level (%) | | | | Model Level (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | 5000 train. | | 1000 test. | | 5000 train. | | 1000 test. | |
| Samp. | Uni. | Non. | Uni. | Non. | Uni. | Non. | Uni. | Non. |
| A | 2.43 | 0.13 | 0.23 | 0.17 | 4.55 | 4.30 | 4.38 | 4.06 |
| B | 0.49 | 0.28 | 0.66 | 0.21 | 1.67 | 1.56 | 1.55 | 1.47 |
| C | 3.84 | 2.99 | 17.20 | 13.44 | 9.11 | 7.34 | 11.48 | 9.00 |

# Average Mutation Score

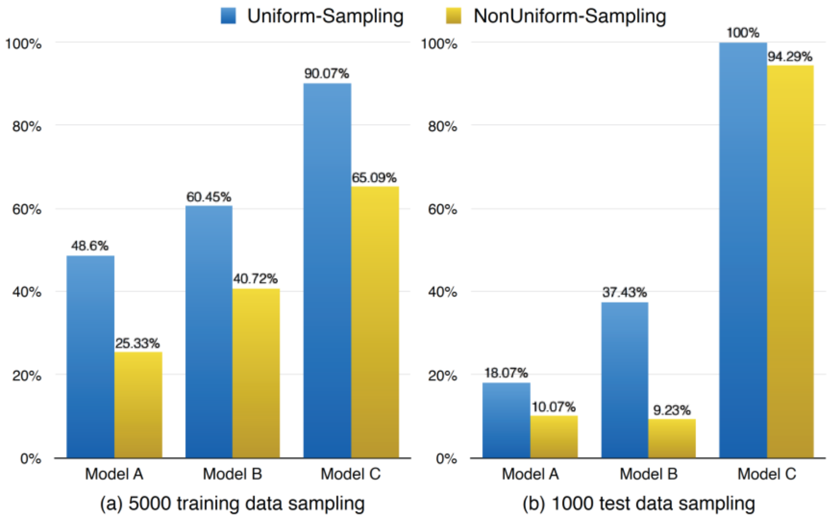



**Fig. 6:** The averaged mutation score of source-level mutation testing.

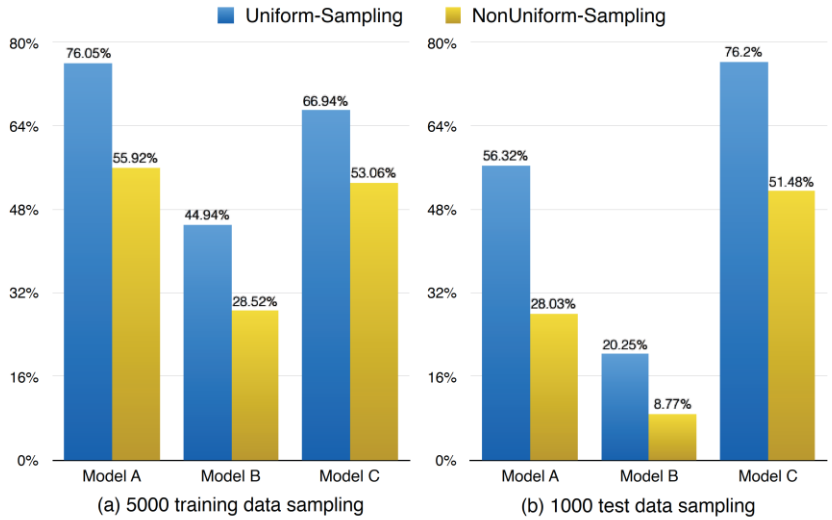



**Fig. 7:** The averaged mutation score of model-level mutation testing.

# Class-wise Performance

**TABLE VI:** The model-level MT score and average error rate of test data by class. According to our mutation score definition, the maximal possible mutation score for a single class is 10%.

| M. | Eval. | Classification Class (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| A | mu. sc. | 7.22 | 8.75 | 9.03 | 6.25 | 8.75 | 8.19 | 8.75 | 9.17 | 9.72 | 9.03 |
| | avg.err. | 3.41 | 3.50 | 1.81 | 1.48 | 4.82 | 2.52 | 5.50 | 4.25 | 10.45 | 3.11 |
| B | mu. sc. | 1.59 | 3.29 | 8.29 | 7.44 | 5.49 | 4.02 | 8.17 | 3.66 | 5.85 | 8.41 |
| | avg.err. | 0.41 | 1.42 | 1.12 | 1.55 | 1.07 | 2.92 | 2.95 | 1.21 | 1.24 | 2.11 |
| C | mu. sc. | 8.33 | 7.95 | 8.97 | 9.74 | 9.74 | 9.62 | 9.62 | 8.97 | 9.74 | 7.56 |
| | avg.err. | 3.67 | 6.22 | 14.80 | 8.84 | 9.11 | 11.53 | 6.83 | 11.48 | 8.87 | 8.55 |

# Class-wise Performance

**TABLE VI:** The model-level MT score and average error rate of test data by class. According to our mutation score definition, the maximal possible mutation score for a single class is 10%.

| M. | Eval. | Classification Class (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| A | mu. sc. | 7.22 | 8.75 | 9.03 | 6.25 | 8.75 | 8.19 | 8.75 | 9.17 | 9.72 | 9.03 |
| | avg.err. | 3.41 | 3.50 | 1.81 | 1.48 | 4.82 | 2.52 | 5.50 | 4.25 | 10.45 | 3.11 |
| B | mu. sc. | 1.59 | 3.29 | 8.29 | 7.44 | 5.49 | 4.02 | 8.17 | 3.66 | 5.85 | 8.41 |
| | avg.err. | 0.41 | 1.42 | 1.12 | 1.55 | 1.07 | 2.92 | 2.95 | 1.21 | 1.24 | 2.11 |
| C | mu. sc. | 8.33 | 7.95 | 8.97 | 9.74 | 9.74 | 9.62 | 9.62 | 8.97 | 9.74 | 7.56 |
| | avg.err. | 3.67 | 6.22 | 14.80 | 8.84 | 9.11 | 11.53 | 6.83 | 11.48 | 8.87 | 8.55 |

# Class-wise Performance

**TABLE VI:** The model-level MT score and average error rate of test data by class. According to our mutation score definition, the maximal possible mutation score for a single class is 10%.

| M. | Eval. | Classification Class (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| A | mu. sc. | 7.22 | 8.75 | 9.03 | 6.25 | 8.75 | 8.19 | 8.75 | 9.17 | 9.72 | 9.03 |
| | avg.err. | 3.41 | 3.50 | 1.81 | 1.48 | 4.82 | 2.52 | 5.50 | 4.25 | 10.45 | 3.11 |
| B | mu. sc. | 1.59 | 3.29 | 8.29 | 7.44 | 5.49 | 4.02 | 8.17 | 3.66 | 5.85 | 8.41 |
| | avg.err. | 0.41 | 1.42 | 1.12 | 1.55 | 1.07 | 2.92 | 2.95 | 1.21 | 1.24 | 2.11 |
| C | mu. sc. | 8.33 | 7.95 | 8.97 | 9.74 | 9.74 | 9.62 | 9.62 | 8.97 | 9.74 | 7.56 |
| | avg.err. | 3.67 | 6.22 | 14.80 | 8.84 | 9.11 | 11.53 | 6.83 | 11.48 | 8.87 | 8.55 |

# Discussion

| | |
|---|---|
| Time for Training | Source-level testing needs re-training of the entire model |
| Why First? | Why did no one try this before? |
| Generalization | Imperfection of model |

# Discussion

| | |
|---|---|
| Designing Mutation Operator | Challenging to simulate real world faults on source-level,<br>Impact difference on source-level and model-level |
| CPU vs GPU | Non-deterministic behavior |
| Relation with Accuracy | Training and test accuracy vs proposed metrics |

# Related Works – Testing for DL

- Other papers by the same team
  - DeepGauge, DeepCruiser, DeepCT, DeepHunter
- **DeepXplore, DeepTest**
- DeepLaser: Practical Fault Attack on Deep Neural Networks
- DeepRoad
- DeepCover (Testing Deep Neural Networks)
- Concolic Testing for Deep Neural Networks
- TensorFuzz - by Goodfellow
- DeepFault: Fault Localization for Deep Neural Networks
- Review Paper - On Testing Machine Learning Programs

# Related Works – Verification for DL

- **AI$^2$**
- Reluplex
- DeepSafe
- Towards evaluating the robustness of neural networks
- Safety Verification of Deep Neural Networks

Thank you for your attention