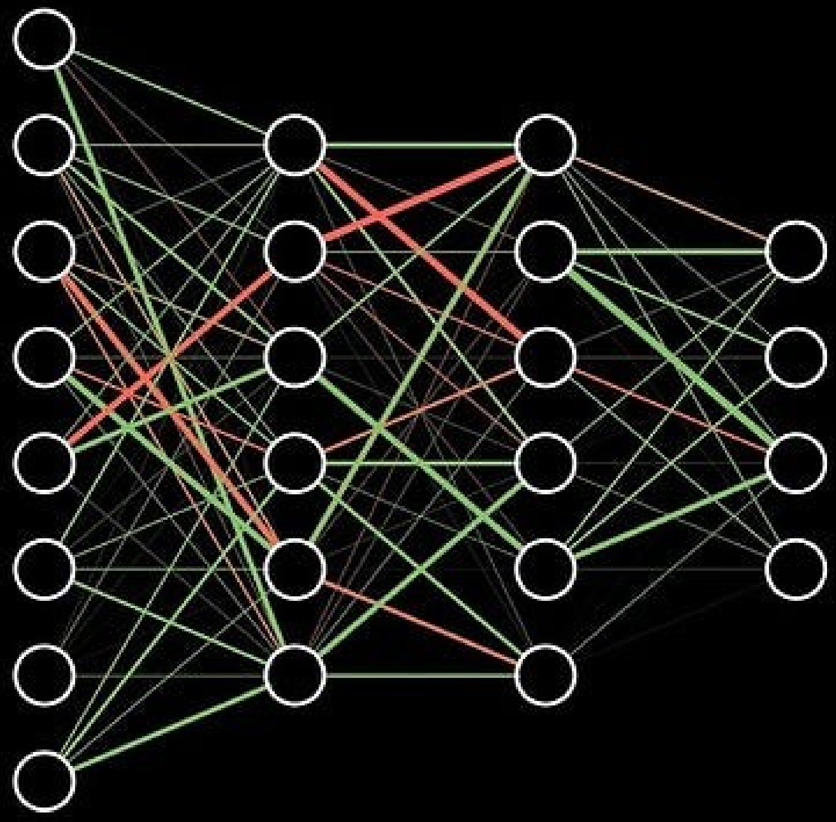
Prioritizing Test Inputs for Deep Neural Networks via Mutation Analysis

Zan Wang ,Hanmo You ,Junjie Cheny ,Yingyi Zhang ,Xuyuan Dong, Wenbin Zhang

Presentation by: Rahil Mehta

DNN



The Problem Addressed

- *Complexity*: DNN models often comprise numerous layers and parameters, necessitating comprehensive testing to uncover potential faults.

- *Resource-Intensive Labeling*: The process of labeling test inputs for DNN model verification is laborious and resource-intensive.

Motivation Behind Solving This Problem

- Improved efficiency in DNN testing enables faster iterations in model development and deployment.

- As DNNs become integral to more applications, ensuring Reliable DNN predictions are crucial for applications in critical domains such as healthcare, autonomous vehicles, and finance.
- For Example, an Uber autonomous vehicle killed a pedestrian in Tempe, Arizona, in 2018.

PRIMA

- It is an input prioritization approach for DNNs via intelligent mutation analysis to label more bug-revealing test inputs

- It is based on the key insight that if a test input that is able to kill many mutated models and produce different prediction results with many mutated inputs, is more likely to reveal DNN bugs, and thus it should be prioritized higher.

Overview of PRIMA

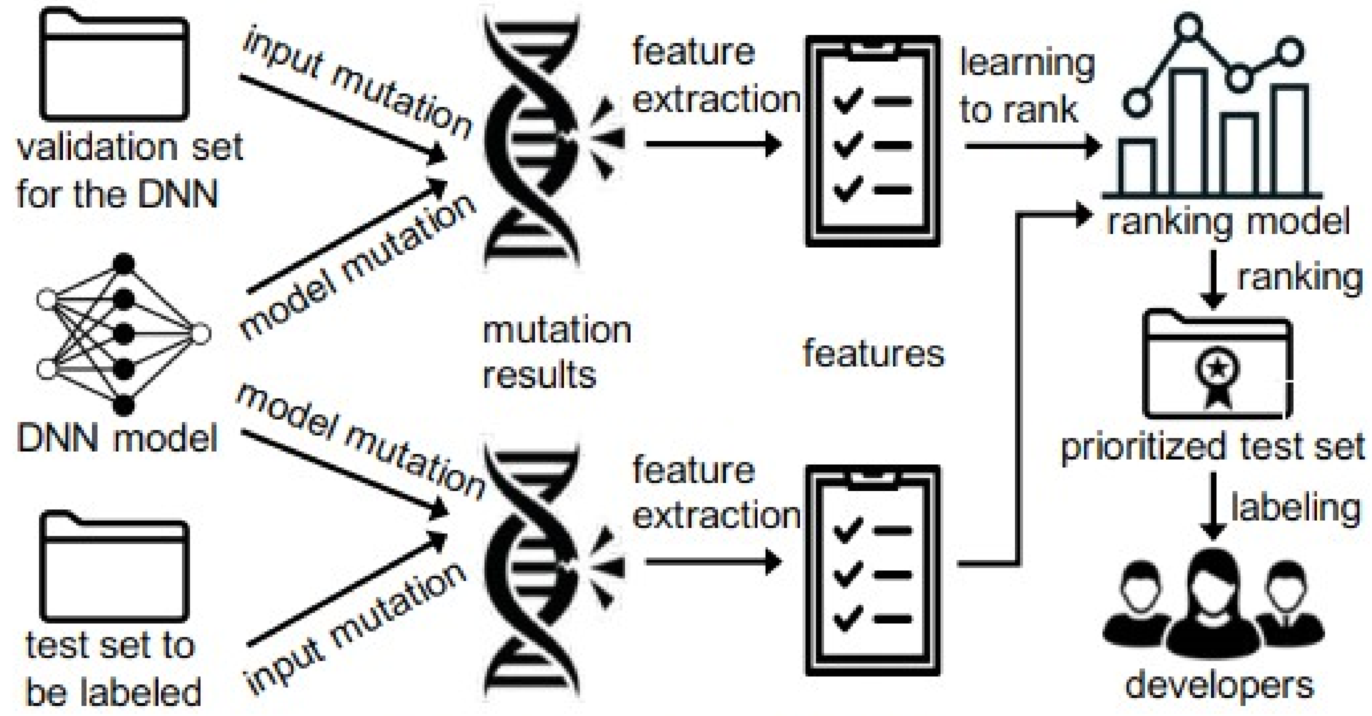


Fig. 1: Overview of PRIMA

Mutation Rules

- **Model Mutation Rules:**
  - Neuron Activation Inverse (NAI): Inverts the activation state of a neuron by changing the sign of the neuron output before passing it to the activation function.
  - Neuron Effect Block (NEB): Blocks the effect of a neuron on the next layers by setting the neuron weights to the next layers to be 0.
  - Weights Shuffling (WS): Shuffles the weights of a neuron with the previous layers.

- **Input Mutation Rules:**
- For images:
  - Pixel Gauss Fuzzing (PGF): Adds noise to the selected pixels following a Gaussian distribution.
  - Pixels Shuffling (PS): Shuffles the selected pixels.
  - Pixel Color Reverse (PCR): Reverses the colors of the selected pixels.
- For sequential data (e.g., text):
  - Character Shuffling (CS): Shuffles the selected characters.
  - Character Replacement (CRL): Replaces the selected characters with other characters randomly selected from the whole set.

Research Question

- RQ1 is to investigate the effectiveness of PRIMA compared with the existing approaches
- RQ2 is to investigate the efficiency of PRIMA

TABLE I: Basic information of subjects

ID	Dataset	Model	#Test	Type	Domain
1	CIFAR-10	VGG-16	10,000	original	image
2	CIFAR-10	VGG-16	10,000	+BIM	image
3	CIFAR-10	VGG-16	10,000	+C&W	image
4	CIFAR-10	VGG-16	10,000	+JSMA	image
5	CIFAR-10	ResNet-20	10,000	original	image
6	CIFAR-10	ResNet-20	10,000	+BIM	image
7	CIFAR-10	ResNet-20	10,000	+C&W	image
8	CIFAR-10	ResNet-20	10,000	+JSMA	image
9	CIFAR-100	VGG-19	10,000	original	image
10	CIFAR-100	VGG-19	10,000	+BIM	image
11	CIFAR-100	VGG-19	10,000	+C&W	image
12	CIFAR-100	VGG-19	10,000	+JSMA	image
13	CIFAR-100	ResNet-32	10,000	original	image
14	CIFAR-100	ResNet-32	10,000	+BIM	image
15	CIFAR-100	ResNet-32	10,000	+C&W	image
16	CIFAR-100	ResNet-32	10,000	+JSMA	image
17	MNIST	LeNet-5	10,000	original	image
18	MNIST-M1	LeNet-5	10,000	original	image
19	MNIST-M2	LeNet-5	10,000	original	image
20	MNIST-M3	LeNet-5	10,000	original	image
21	MNIST_VS_USPS	LeNet-5	1,800	original	image
22	COIL	VGG-11	1,000	original	image
23	PIE27_VS_PIE5	VGG-11	3,332	original	image
24	PIE27_VS_PIE9	VGG-11	1,632	original	image
25	Driving	Dave-orig	5,614	original	image
26	Driving	Dave-drop	5,614	original	image
27	Driving	Dave-orig	5,614	light	image
28	Driving	Dave-drop	5,614	light	image
29	Driving	Dave-orig	5,614	patch	image
30	Driving	Dave-drop	5,614	patch	image
31	TREC	Bi-LSTM	952	original	text
32	IMDB	Bi-LSTM	15,000	original	text
33	SMS Spam	Bi-LSTM	3,000	original	text
34	CoLA	Bi-LSTM	4,000	original	text
35	Hate Speech	Bi-LSTM	14,652	original	text
36	KDDCUP99	CNN	311,027	original	features

36 pairs of datasets and DNN models as subjects

Evaluation of Effectiveness

TABLE II: Overall comparison results across all the subjects

	Approach	#Best cases in RAUC-					Average RAUC-					Improvement of PRIMA (%) in RAUC-				
		100	200	300	500	All	100	200	300	500	All	100	200	300	500	All
C	DeepGini	2	2	2	1	3	0.751	0.753	0.752	0.755	0.847	18.24	16.47	15.69	14.97	8.50
	LSA	0	0	0	0	0	0.568	0.558	0.559	0.571	0.685	56.34	57.17	55.64	52.01	34.16
	DSA	0	0	0	0	0	0.648	0.632	0.625	0.619	0.722	37.04	38.77	39.20	40.23	27.29
	PRIMA	28	28	28	29	27	0.888	0.877	0.870	0.868	0.919	-	-	-	-	-
R	LSA	0	0	0	0	0	0.345	0.357	0.368	0.394	0.689	131.01	122.97	114.67	97.72	17.27
	PRIMA	6	6	6	6	6	0.797	0.796	0.790	0.779	0.808	-	-	-	-	-

- **Columns 3-7**: These columns contain the number of subjects (i.e., distinct DNN models or datasets) where each test input prioritization approach achieved the best performance for each of the five metrics considered in the study.
- **Columns 8-12**: These columns provide the average results obtained by each prioritization approach across all subjects in terms of each metric.
- **Columns 13-27**: These columns detail the average improvement percentage of PRIMA over the compared approaches for each metric. This shows how much better PRIMA performs relative to other methods like DeepGini, LSA, and DSA.

Evaluation of Efficiency

Approach	Mean	Std.	Min.	Max.
DeepGini	0.1	0.1	< 0.1	1.4
LSA	1.5	1.0	< 0.1	2.8
DSA	23.5	110.1	< 0.1	616.2
PRIMA	10.4	6.2	2.4	27.7

Practical Evaluation

ID	Approach	RAUC-				
		100	200	300	500	All
$M1_t$	DeepGini	0.512	0.535	0.541	0.556	0.688
	PRIMA	0.740	0.651	0.605	0.555	0.746
$M2_t$	DeepGini	0.491	0.596	0.621	0.659	0.765
	PRIMA	0.889	0.830	0.776	0.720	0.766
$M3_{t1}$	DeepGini	0.847	0.903	0.917	0.827	0.656
	PRIMA	1.000	1.000	0.987	0.857	0.651
$M3_{t2}$	DeepGini	0.983	0.981	0.980	0.976	0.941
	PRIMA	1.000	0.995	0.990	0.986	0.935

Assumptions made

- The features extracted for use in the ranking algorithm are relevant and sufficiently informative for assessing the fault-revealing potential of test inputs, contributing to the accuracy of the prioritization.

- The paper assumes that the ranking algorithm can correctly learn from the features it sees and correctly identify which test inputs are most likely to show problems.

Limitations

- The proposed method, dependent on domain knowledge of mutation rules, requires expertise in the DNN architecture to apply mutation rules for specific DNN architecture.

- The feature extraction process and ranking framework introduce additional computational overhead, impacting the overall efficiency of the testing process.

Discussion points

- Can the PRIMA approach be extended to other machine learning models beyond DNNs, such as support vector machines (SVMs) or random forests?
- Can there be ethical implications and potential biases in Prima's approach, especially in scenarios where prioritizing certain inputs over others could lead to biases or unintended consequences?
- Are there practical scalability and implementation considerations for applying PRIMA to large-scale models?