Weight-based Semantic Testing Approach for Neural Networks

Amany Alshareef, Nicolas Berthier, Sven Schewe, Xiaowei Huang

Department of Computer Science
University of Liverpool, UK

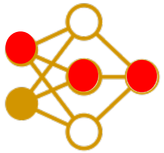
Testing goal

- Evidence that the system meets its requirements
- Evidence that the system is error-free

Machine learned software !!

- ➢ Data-driven NOT requirements-driven
- ➢ Do NOT have a specific control-flow structure

- ➢ Most testing techniques propose structural coverage
- ➢ Tend at transforming in the input data



Real world high-dimensional data lie on low-dimensional manifolds embedded within the high-dimensional.

→ Not thinking about ~~input~~ domain, instead where the data lie and model that

The research presents a testing approach for neural networks that leverages the learned representations and feature importance to evaluate the test data coverage.

Analyse the latent features learnt by the model

Generate additional test cases

Bayesian Network Abstraction Model

A dimensionality reduction technique using feature extraction algorithms to abstract the behaviour of a neural network into a BN.

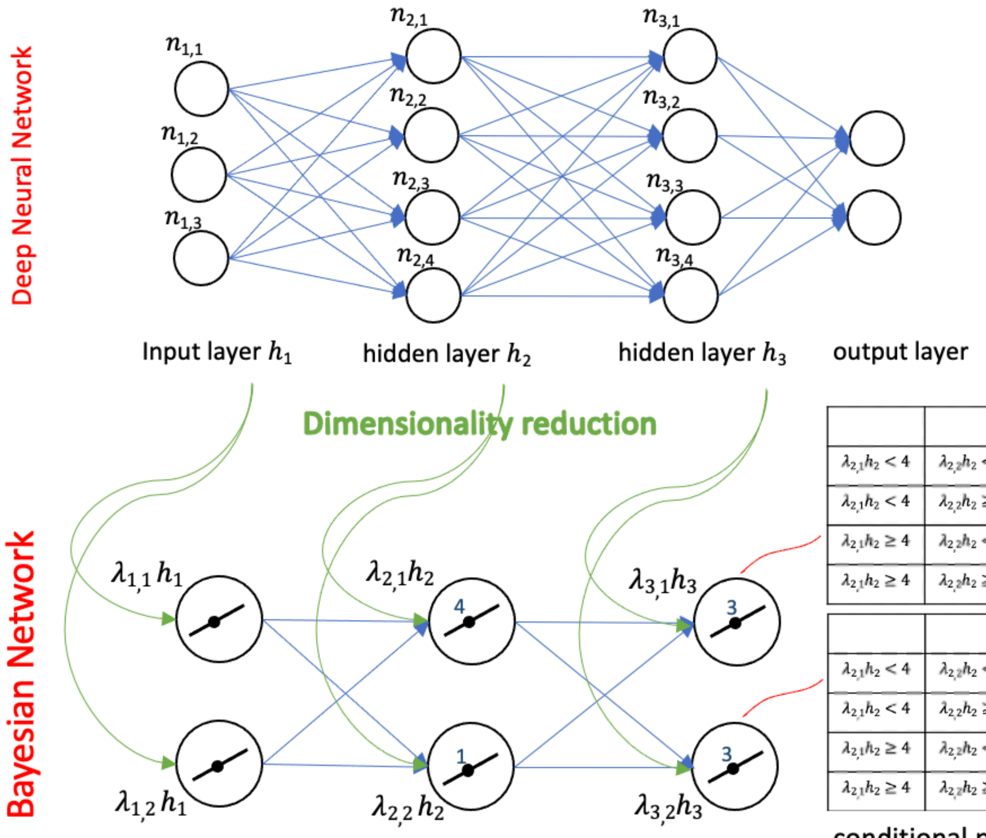
Constructing a Bayesian Network:

I. Hidden features extraction
Map from a high-dimensional space into a feature space.

II. Feature space discretization
Discretise each feature component into finite feature intervals.

III. Probability tables construction

- Represent the probabilistic distribution of each extracted feature with a node in the BN.
- Associate each feature with a marginal or a conditional probability table.



		$\lambda_{3,1}h_3 < 3$	$\lambda_{3,1}h_3 \geq 3$
$\lambda_{2,1}h_2 < 4$	$\lambda_{2,2}h_2 < 1$	1.0	0
$\lambda_{2,1}h_2 < 4$	$\lambda_{2,2}h_2 \geq 1$	0.8	0.2
$\lambda_{2,1}h_2 \geq 4$	$\lambda_{2,2}h_2 < 1$	0.5	0.5
$\lambda_{2,1}h_2 \geq 4$	$\lambda_{2,2}h_2 \geq 1$	0.6	0.4

		$\lambda_{3,2}h_3 < 3$	$\lambda_{3,2}h_3 \geq 3$
$\lambda_{2,1}h_2 < 4$	$\lambda_{2,2}h_2 < 1$	0.4	0.6
$\lambda_{2,1}h_2 < 4$	$\lambda_{2,2}h_2 \geq 1$	0.9	0.1
$\lambda_{2,1}h_2 \geq 4$	$\lambda_{2,2}h_2 < 1$	0	1.0
$\lambda_{2,1}h_2 \geq 4$	$\lambda_{2,2}h_2 \geq 1$	0.3	0.7

conditional probability table

UNIVERSITY OF LIVERPOOL

❶ BN-based Latent Feature Analysis

Estimate the **importance** of a neural network's **latent features** by analysing an associated **Bayesian network's sensitivity** to **distributional shifts**

1. Probability calculation for input sample under BN

   Perform the feature projection and discretisation step to input sample to obtain the associated feature intervals, and then calculate their probability belonging to the BN distribution.

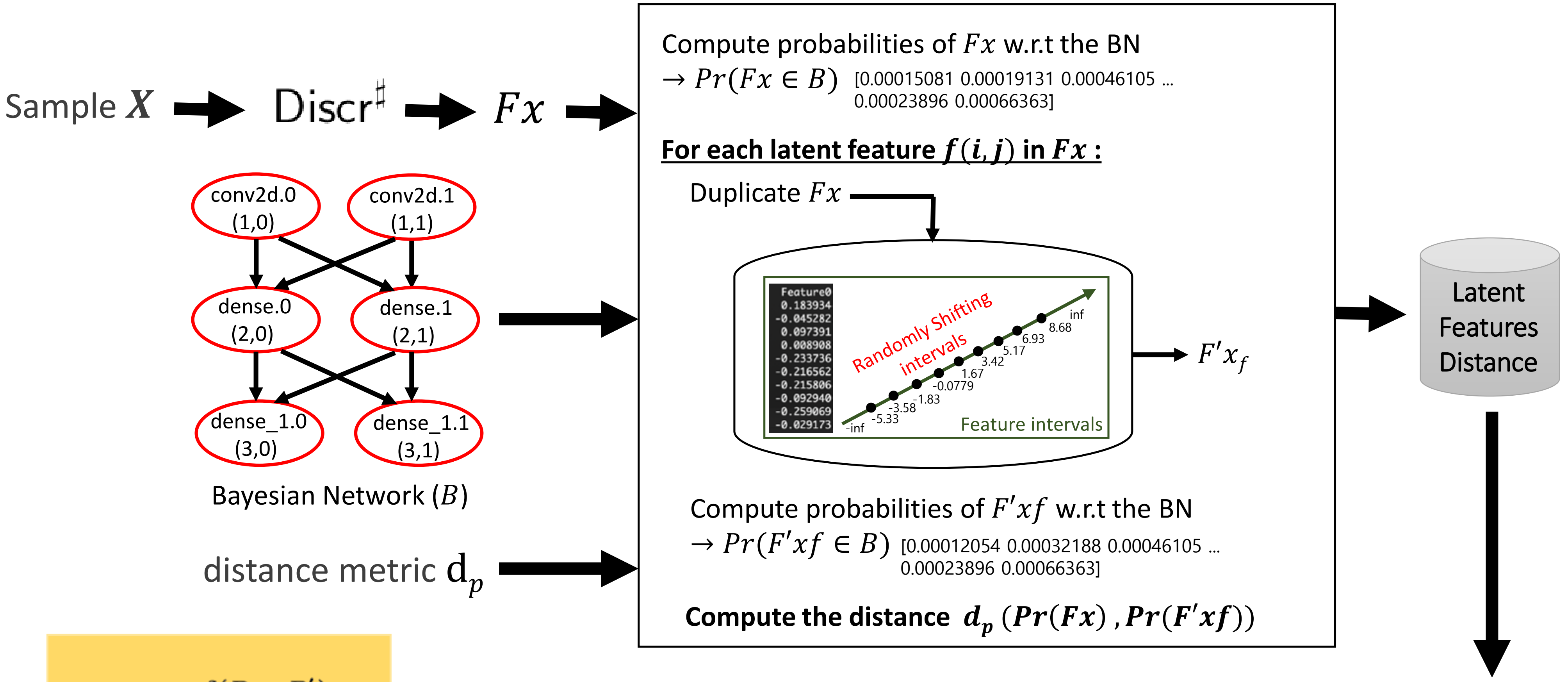
2. Latent features perturbation

   For each latent feature, randomly shifting its intervals in a selected feature space.

3. Distance computation

   Compute the distance between the original probability vector and the probability vector obtained from the perturbed features.

The proposed BN analysis technique to compute the sensitivity of extracted latent features



$$W_{i,j} = \frac{\delta(P_{ref}, P'_f)}{\sum_{f \in \mathbb{F}^\sharp} \delta(P_{ref}, P'_f)}$$

$w_{(1,1)} = 0.192$

$w_{(2,1)} = 0.172$

distance / perturbed feature	d_{L_1}	d_{L_2}	d_{L_∞}	d_{JS}	d_{corr}	d_{cos}	d_{MSE}	d_{RMSE}	d_{MAE}	d_{AF}
(1, 0)	150	0.726	0.009 56	0.224	0.142	0.114	0.000 000 879	0.000 937	0.000 249	0.278
(1, 1)	340	1.18	0.009 89	0.353	0.448	0.361	0.000 002 32	0.001 52	0.000 567	0.735
(2, 0)	325	1.09	0.009 46	0.365	0.332	0.267	0.000 001 98	0.001 41	0.000 541	0.625
(2, 1)	360	1.16	0.0103	0.393	0.395	0.323	0.000 002 24	0.001 50	0.000 600	0.710
(3, 0)	276	0.880	0.008 89	0.258	0.170	0.137	0.000 001 29	0.001 14	0.000 460	0.408
(3, 1)	315	1.07	0.009 60	0.324	0.318	0.264	0.000 001 92	0.001 39	0.000 525	0.608

Higher distribution change → Higher importance score

❷ Weight-based Testing Metric

Transforming the traditional binary coverage approach to a weighted probability problem and define our coverage metric based on the latent features importance.

- Weight-based Feature Coverage

$$\mathrm{WFCov}(\mathcal{B}_{\mathcal{N}},x) \stackrel{\text{def}}{=} \sum_{(\!|f^{\sharp}_{i,j}|\!) \in V_{\mathcal{N},X}} w_{(\!|f^{\sharp}_{i,j}|\!)} \cdot \frac{\left|\left\{ f^{\sharp k}_{i,j} \in \mathbb{F}^{\sharp}_{i,j} \mid \mathcal{P}_i\left(f^{\sharp k}_{i,j}\right) \geq \varepsilon \right\}\right|}{\left|\mathbb{F}^{\sharp}_{i,j}\right|}$$

- Weight-based Feature Dependence Coverage

$$\mathrm{WFdCov}\left(\mathcal{B}_{\mathcal{N}},X\right) \stackrel{def}{=} \sum_{(\!|f^{\sharp}_{i,j}|\!) \in V^{+}_{\mathcal{N},X}} w_{(\!|f^{\sharp}_{i,j}|\!)} \cdot \frac{\left| \begin{array}{c|c} (f^{\sharp k}_{i,j}, F^{\sharp}_{i-1}) \in & \mathcal{CP}_i\left(f^{\sharp k}_{i,j} | F^{\sharp}_{i-1}\right) \geq \varepsilon \\ \mathbb{F}^{\sharp}_{i,j} \times \mathbb{F}^{\sharp}_{i-1} & \vee \quad \mathcal{P}_i\left(f^{\sharp k}_{i,j}\right) < \varepsilon \end{array} \right|}{\left|\mathbb{F}^{\sharp}_{i,j} \times \mathbb{F}^{\sharp}_{i-1}\right|}$$

The two feature metrics are combined to produce the generalised weight feature coverage.

$$\mathrm{WFCovTot}\left(\mathcal{B}_{\mathcal{N}},X\right) = \sum_{(\!|f^{\sharp}_{i,j}|\!) \in V_{\mathcal{N},X}} w_{(\!|f^{\sharp}_{i,j}|\!)} \cdot \begin{cases} \mathrm{WFCov}_{(\!|f^{\sharp}_{i,j}|\!)} & \text{if } i = 1\,, \\ \frac{1}{2}\left(\mathrm{WFCov}_{(\!|f^{\sharp}_{i,j}|\!)} + \mathrm{WFdCov}_{(\!|f^{\sharp}_{i,j}|\!)}\right) & \text{otherwise}\,. \end{cases}$$

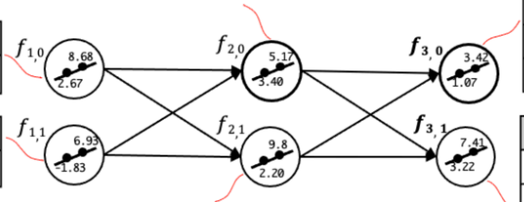
❷ Weight-based Testing Metric

$f_{1,0} < 2$	$2 \le f_{1,0} \le 8$	$f_{1,0} > 8$	weight
0.518	0.150	0.332	0.0318

$f_{1,1} < -1$	$-1 \le f_{1,1} \le 6$	$f_{1,1} > 6$	weight
0.129	0.776	0.095	0.0992

		$f_{2,0} < 3$	$3 \le f_{2,0} \le 5$	$f_{2,0} > 5$
$f_{1,0} < 2$	$f_{1,1} < -1$	0.44	0.24	0.32
$2 \le f_{1,0} \le 8$	$f_{1,1} < -1$	0.75	0.25	0.00
...	...	...	...	...
$f_{1,0} > 8$	$f_{1,1} > 6$	0.16	0.75	0.09
marginal distribution		0.343	0.207	0.450
node weight		0.1976		

		$f_{2,1} < 2$	$2 \le f_{2,1} \le 9$	$f_{2,1} > 9$
$f_{1,0} < 2$	$f_{1,1} < -1$	0.60	0.10	0.30
$f_{1,0} < 2$	$-1 \le f_{1,1} \le 6$	0.16	0.75	0.09
...	...	...	...	...
$f_{1,0} > 8$	$f_{1,1} > 6$	0.50	5.00	0.0
marginal distribution		0.243	0.521	0.236
node weight		0.2284		



		$f_{3,0} < 3$	$3 \le f_{3,0} \le 5$	$f_{3,0} > 5$
$f_{2,0} < 3$	$f_{2,1} < 2$	0.90	0.05	0.05
$f_{2,0} < 3$	$2 \le f_{2,1} \le 9$	0.10	0.70	0.20
$f_{2,0} < 3$	$f_{2,1} > 9$	0.80	0.15	0.05
$3 \le f_{2,0} \le 5$	$f_{2,1} < 2$	0.20	0.80	0.00
$3 \le f2{,}0 \le 5$	$2 \le f2{,}1 \le 9$	0.35	0.34	0.31
$3 \le f_{2,0} \le 5$	$f_{2,1} > 9$	0.40	0.30	0.30
$f_{2,0} > 5$	$f_{2,1} < 2$	0.38	0.53	0.09
$f_{2,0} > 5$	$2 \le f_{2,1} \le 9$	0.45	0.03	0.52
$f_{2,0} > 5$	$f_{2,1} > 9$	0.47	0.02	0.51
marginal distribution		0.453	0.323	0.224
node weight		0.1730		

		$f_{3,1} < 3$	$3 \le f_{3,1} \le 7$	$f_{3,1} > 7$
$f_{2,0} < 3$	$f_{2,1} < 2$	0.4	0.6	0.0
$f_{2,0} < 3$	$2 \le f_{2,1} \le 9$	0.1	0.2	0.7
$f_{2,0} < 3$	$f_{2,1} > 9$	0.55	0.06	0.39
$3 \le f_{2,0} \le 5$	$f_{2,1} < 2$	0.15	0.8	0.05
$3 \le f2{,}0 \le 5$	$2 \le f2{,}1 \le 9$	0.3	0.3	0.4
$3 \le f_{2,0} \le 5$	$f_{2,1} > 9$	0.2	0.6	0.2
$f_{2,0} > 5$	$f_{2,1} < 2$	0.31	0.34	0.35
$f_{2,0} > 5$	$2 \le f_{2,1} \le 9$	0.9	0.0	0.1
$f_{2,0} > 5$	$f_{2,1} > 9$	0.5	0.4	0.1
marginal distribution		0.380	0.366	0.254
node weight		0.2697		

Example 1

$Pr(f_{3,0} < 3) \approx 0.453$, $Pr(3 \le f_{3,0} \le 5) \approx 0.323$, $Pr(f_{3,0} > 5) \approx 0.224$.

$1 \times 0.173 = 0.173.$

WFCov(B_N, x) = 1

Example 2

$\varepsilon = 0.01$

$CP_{f3,0} = 0.1730/0.8687 \cdot 26/27 = 0.1917$

$CP_{f3,1} = 0.2697/0.8687 \cdot 25/27 = 0.2875$

WFdCov(B_N, x) = 0.2190 + 0.2532 + 0.1917 + 0.2875 = 0.9514%.

UNIVERSITY OF LIVERPOOL

Concolic Test Data Generation Algorithm

Algorithm 1 Test Dataset Generation

Input:
$\mathcal{N} \leftarrow$ DNN under test
$X \leftarrow$ data set
$\mathcal{B}_{\mathcal{N},X_{train}} \leftarrow$ abstract BN
$W_f \leftarrow$ features sensitivity weights
Output: test inputs X_0, coverage

$X_0 \leftarrow$ sampling initial seed test inputs from X_{test}
$\mathcal{B}_{\mathcal{N},X_0} \leftarrow$ initialising the BN prob. tables with X_0
$Tar\_invals \leftarrow$ intervals with prob $\leq \varepsilon$
for $i = 1$ to max iterations **do**
  $t \leftarrow Tar\_invals$ with highest weight in W_f
  select a test input $s \in X_0$
  construct an LP problem based on t
  solve the optimisation objective:
  min $\|(n_{1,1}, \ldots, n_{1,|l_1|}) - (s_{1,1}, \ldots, s_{1,|l_1|})\|_\infty$
  $s' = (n_{1,1}, \ldots, n_{1,|l_1|})$
  if s' passes the oracle **then**
    $s' \leftarrow$ newly generated test input
    if $f_{\mathcal{N}}(s') = f_{\mathcal{N}}(s)$ **then**
      $X_0 \leftarrow X_0 \cup \{s'\}$
      update $\mathcal{B}_{\mathcal{N},X_0}$ probabilities
      update coverage
    else
      $s' \leftarrow$ adversarial input
    end if
  end if
end for

UNIVERSITY OF LIVERPOOL

Evaluation

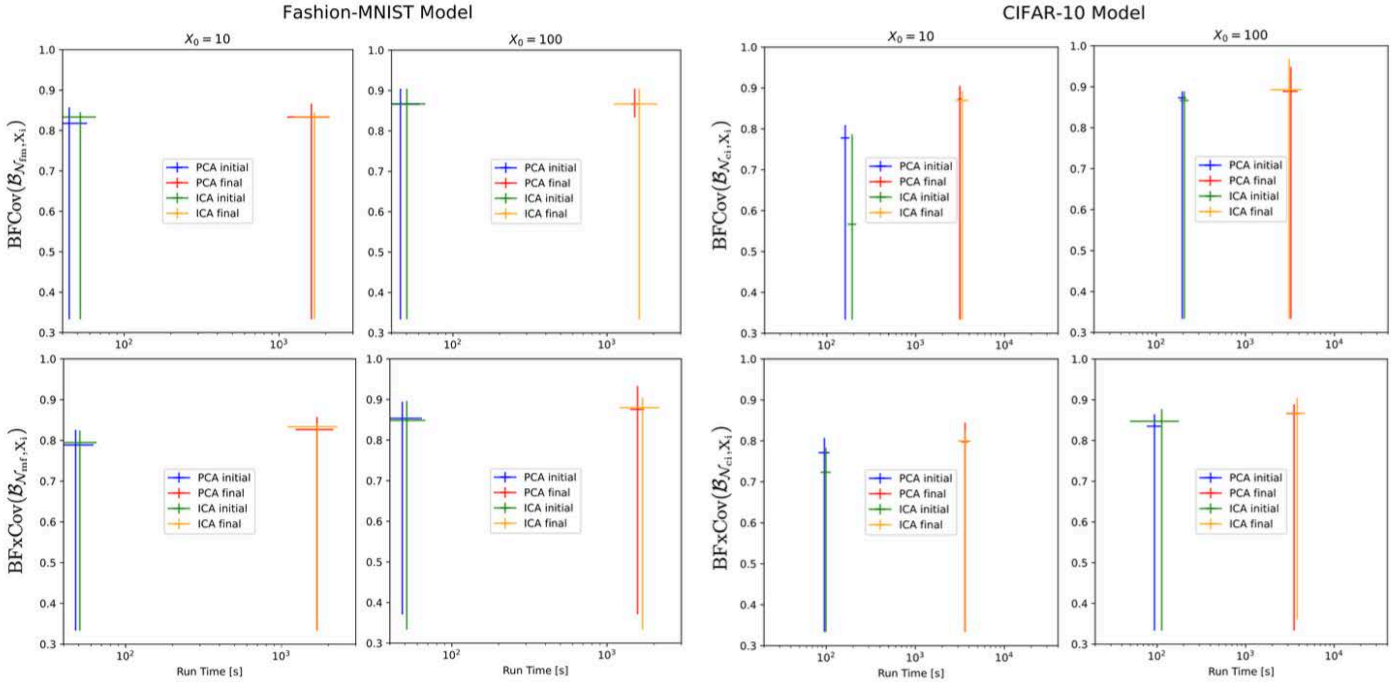
1. Datasets and Models

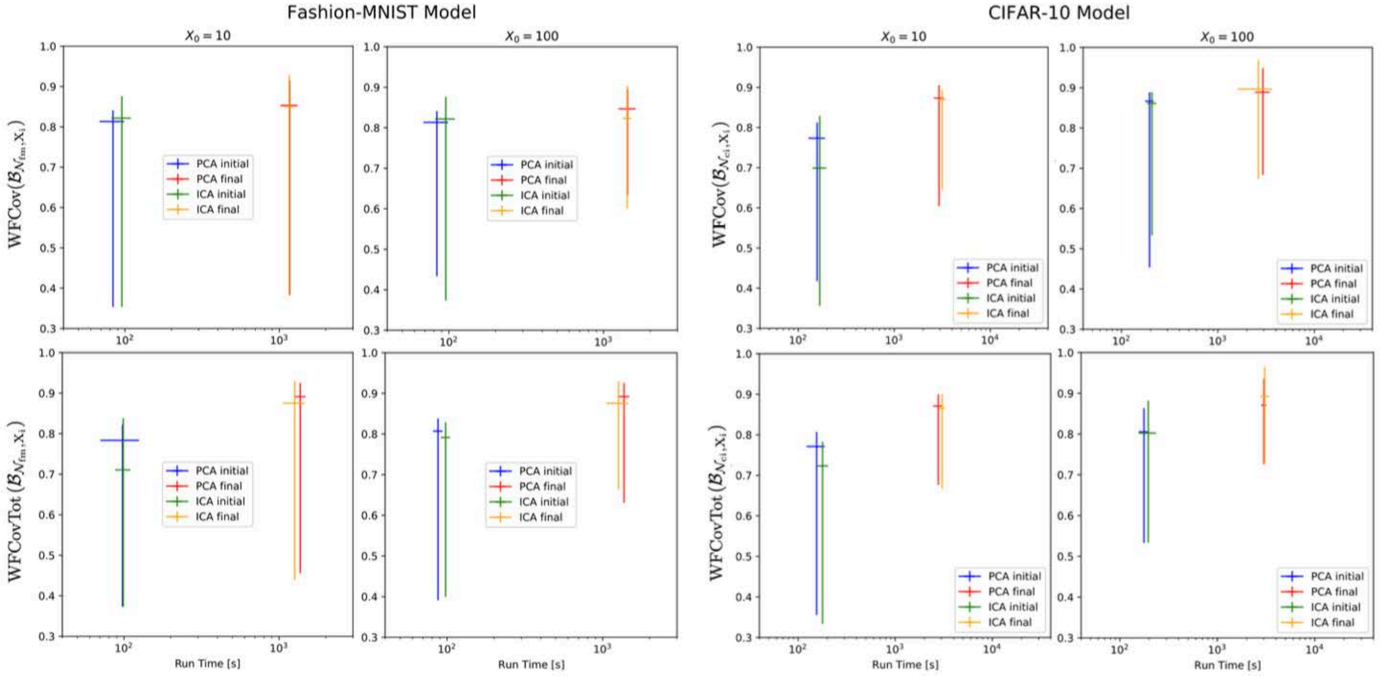
- The first model targets the **Fashion-MNIST** classification problem with 89.03% validation accuracy, and the second one targets the **CIFAR-10** dataset with 81.00% validation accuracy.
- The models are reasonably sized, with more than 10 layers, including blocks of convolutional and max-pooling layers, followed by a series of dense layers.
- The considered layers are the convolutional ReLU, 2d max pooling, and dense ReLU.

2. Experimental Setup

- Two linear feature extraction techniques were selected: PCA and ICA, with two to five numbers of extracted features for each of the abstracted layers.
- The Kernel Density Estimation (KDE) and uniform-based discretisation are considered, with varying numbers of the uniform partitions bins that are: one, three, and five.
- The extended Concolic testing tool is run on both DNNs models with a maximum 100 iterations per run. Each run is initialised with uniformly drawn test sets of 10 and 100 correctly classified inputs.

Results





The overall distribution of initial and the respective final coverage of up to 100 iterations of Concolic test case generation. X-axis indicates the run time in seconds (initial and run time). The vertical lines are the coverage, and the horizontal line on the coverage is the median.

UNIVERSITY OF LIVERPOOL

CONCLUSION

We have introduced a weight-based semantic testing approach that measures how well the DNN is tested by focusing on the important features of the DNN using its abstracted Bayesian network.

- ✓ The developed weighted feature metrics achieved higher testing coverage than the original metrics.
- ✓ The test generation algorithm is directed to synthesise new input targeting features with higher importance scores.
- ✓ Empirically validated the applicability and effectiveness of the proposed weight metrics, which serves as a strong argument in favour of increasing the trustworthy performance of the DNN models.

THANK YOU!