Robust Privacy-Preserving Deep Learning under Side-Channel Power Attacks

Hui Hu, Jessa Gegax-Randazzo, Clay Carper, Dr. Mike Borowczak

University of Wyoming, Cybersecurity Education And Research Lab

ABSTRACT

Privacy in deep learning is receiving tremendous attention with its wide applications in industry and academics. Recent studies have shown a traditional deep neural network is extremely vulnerable to side-channel attacks. In particular, side-channel power attacks are powerful to infer the internal structure of a deep neural network (i.e., the number of nodes in each hidden layer) such that users' extremely sensitive predictions can be exposed severely.

To solve this privacy issue, in this project, we propose $\mathrm{PD^2NN}$, a novel approach for training privacy-preserving deep neural networks under side-channel power attacks. The design principle of $\mathrm{PD^2NN}$ is introducing randomness into the model's internal structure and model training process to generate random power traces. The experimental results on two benchmark datasets demonstrate the effectiveness of the proposed approach.

INTRODUCTION

Privacy in deep learning is becoming increasingly prominent with the emergence of numerous attack techniques. Recent studies have shown that traditional deep neural networks (DNNs) are extremely vulnerable to side-channel power attacks[1]. As Figure 1 shows, the internal structure of a DNN is easily inferred from power traces in the training process. Nevertheless, few privacy-preserving DNNs are developed to resist powerful side-channel power attacks.

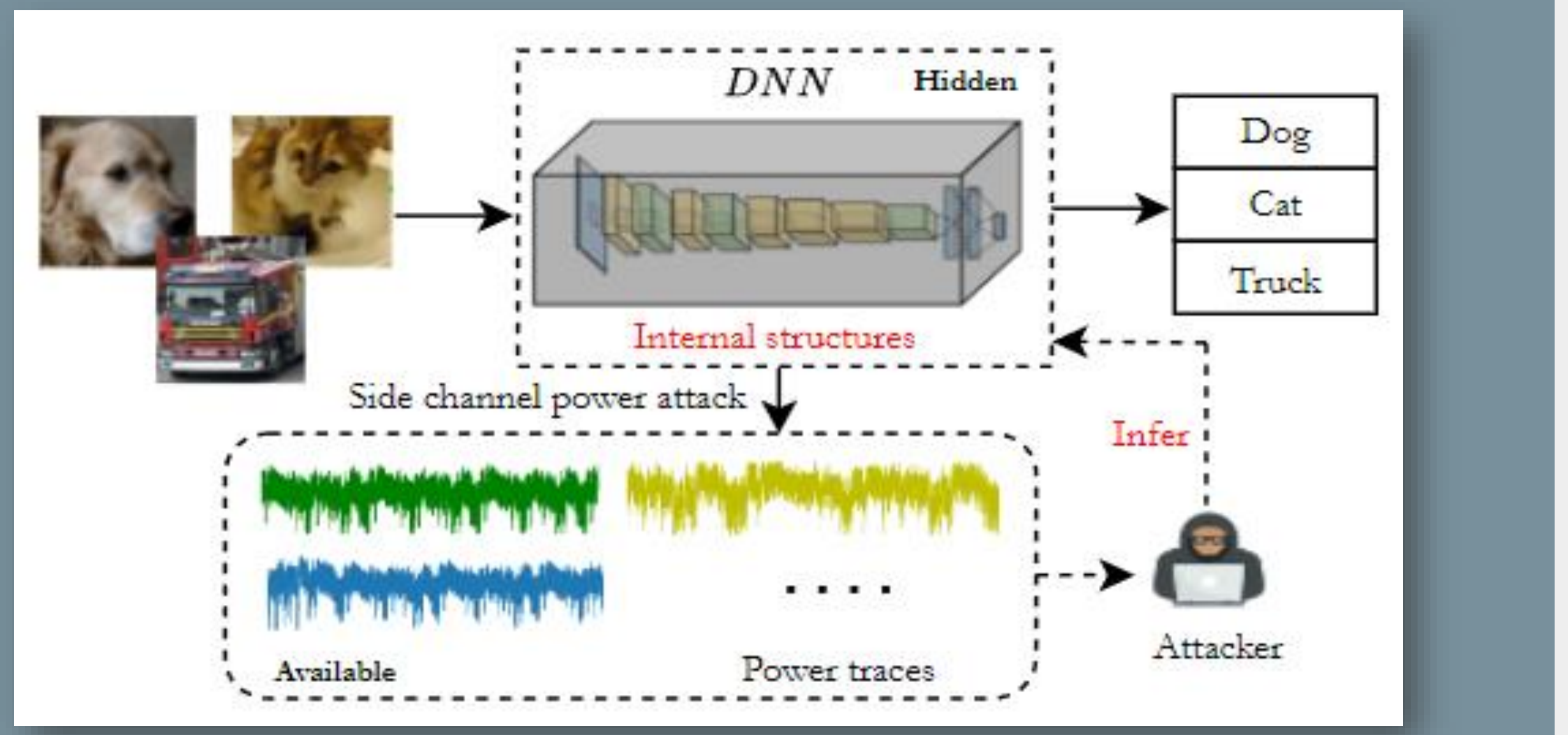


Figure 1. Side-channel power attacks on traditional DNNs.

In this work, we propose a novel approach for training privacy-preserving deep neural networks coined $\mathrm{PD^2NN}$, which can resist side-channel power attacks efficiently and keep high classification accuracy simultaneously. Wan et. al[2] have theoretically and empirically shown that a deep neural network can achieve competitive or even better classification performance by randomly selecting a subset of weights within the network to be zero. Motivated by this finding, in the proposed approach, we randomly generate multiple sub-networks to guarantee model robustness. To resist side-channel power attacks, we introduce randomness into the model training process such that power traces are random in the temporal domain. The experimental results on two benchmark datasets show that $\mathrm{PD^2NN}$ significantly decreases the privacy inference accuracy while maintaining high classification accuracy.

METHODOLOGY

The motivation for the $\mathrm{PD^2NN}$ design is based on two principles: (1) model privacy-preserving performance under side-channel power attacks is guaranteed by randomness; (2) and model classification accuracy is promised by the argument in [2], which states that a deep neural network can achieve competitive classification performance by using partial nodes in the network. As Figure 2 shows, the workflow of the proposed approach includes three sub-modules: Independent Sub-network Construction Module, Sub-network Random Training Module, and Prediction Module.

Independent Sub-network Construction Module

This module aims to construct multiple independent sub-networks by randomly selecting hidden nodes in the whole neural network. Algorithm 1 summarizes the construction process of r sub-networks in deep neural networks.

```
Algorithm 1 Independent Sub-network Construction
Input: Deep neural network $\mathcal{N}$ with two hidden layers, the number
    of hidden nodes $m$, one input layer $\mathcal{I}$ with $d$ nodes, and one
    output layer $\mathcal{O}$ with 1 node.
Output: $\mathcal{N}_1, \mathcal{N}_2, \cdots, \mathcal{N}_r$.
While $i \leq r$ do
    Randomly generate the number of nodes in the first hidden
    layer $z_{i1} \in [2, m]$.
    Randomly generate the number of nodes in the second hidden
    layer $z_{i2} \in [2, m]$.
Return $(z_{i1}, z_{i2})_{i=1,\cdots,r}$.
```

Sub-network Random Training Module

This module randomly trains r generated sub-networks such that the power trace patterns are random in the temporal domain. Further, a side-channel power attacker will be more difficult to infer the model's internal structure from the random power traces. Figure 3 shows the random training process.

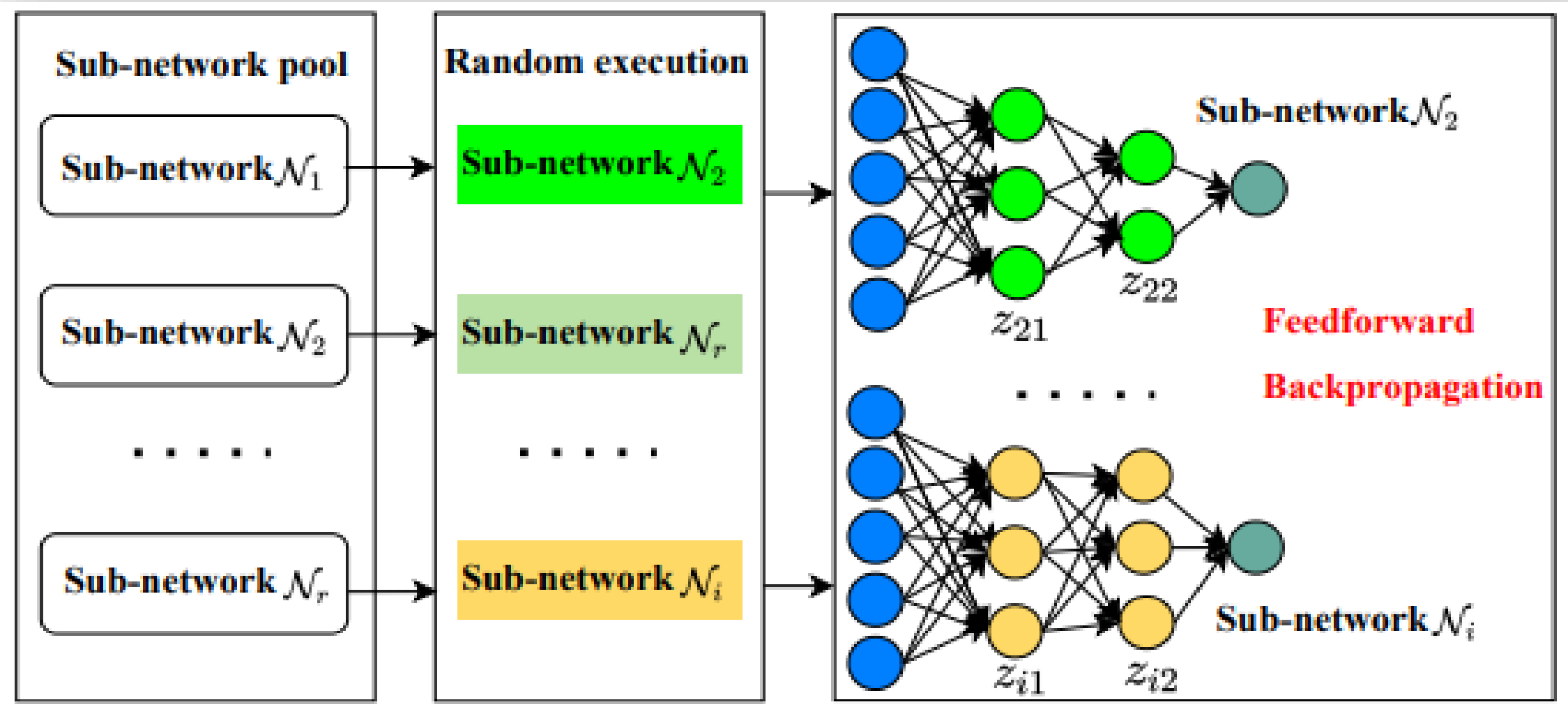

Sub-network pool
Sub-network $\mathcal{N}_1$
Sub-network $\mathcal{N}_2$
· · · · ·
Sub-network $\mathcal{N}_r$
Random execution
Sub-network $\mathcal{N}_2$
Sub-network $\mathcal{N}_r$
· · · · ·
Sub-network $\mathcal{N}_i$
Sub-network $\mathcal{N}_2$
z_{21}
z_{22}
Feedforward
Backpropagation
· · · · ·
Sub-network $\mathcal{N}_i$
z_{i1}
z_{i2}

Figure 3. Sub-network random training process.

Prediction Module

This module seeks to select the most accurate sub-network on the labeled samples to predict the labels of the unlabeled samples.

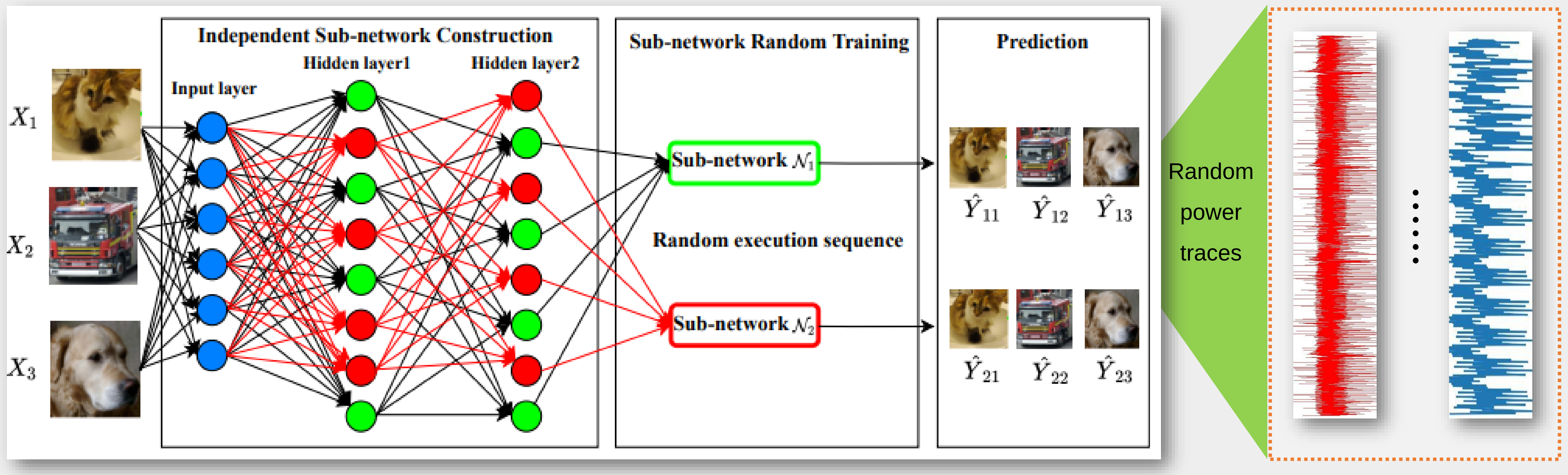


Figure 2. The overall framework of $\mathrm{PD^2NN}$.

EXPERIMENT RESULTS

Hardware and Software Platforms. We used ChipWhisperer-Lite (CW-Lite) board to collect power traces of $\mathrm{PD^2NN}$ and a traditional DNN. Both models are developed in C codes and run on an XMEGA target board. We sent data from a Jupyter Notebook to the CW-Lite.

Dataset Selection. The performance of $\mathrm{PD^2NN}$ was evaluated on the UCI Pima Indians Diabetes dataset[3] and COMPAS dataset[4].

Model Parameter Settings. For $\mathrm{PD^2NN}$, the number of sub-networks is two. Seven random internal structures are generated.

Table 1 shows the classification performance on the two datasets. Table 2 shows the inference performance on the Diabetes dataset.

Figure 4 shows the power traces of two different neural networks (8 nodes versus 14 nodes).

Table 1. Classification accuracy of $\mathrm{PD^2NN}$ and traditional DNNs.

# of hidden nodes	Models	Diabetes	COMPAS
8 nodes	Traditional DNN	0.6257	0.7644
	$\mathrm{PD^2NN}$	**0.6268**$^{\uparrow 0.17\%}$	**0.6938**$^{\downarrow 9.24\%}$
10 nodes	Traditional DNN	0.6415	0.7852
	$\mathrm{PD^2NN}$	**0.6461**$^{\uparrow 0.72\%}$	**0.7712**$^{\downarrow 1.78\%}$
14 nodes	Traditional DNN	0.6233	0.7720
	$\mathrm{PD^2NN}$	**0.6172**$^{\downarrow 0.98\%}$	**0.7616**$^{\downarrow 1.35\%}$

Note: Higher value is better. ↑ or ↓ denotes the increasing or decreasing percentages compared with the traditional DNNs.

First finding

- The classification accuracy of $\mathrm{PD^2NN}$ is **slightly lower** compared with the traditional DNNs in most cases.
- Compared with the inference accuracy as shown in Table 2, $\mathrm{PD^2NN}$ significantly improves model privacy-preserving performance with a slight loss of classification accuracy.

Table 2. Inference accuracy of $\mathrm{PD^2NN}$ and traditional DNNs.

Models	Structures	k=3	k=6
Traditional DNN	(8,8)(14,14)	**1.0000**	**1.0000**
$\mathrm{PD^2NN}$	Structure 6	0.6193$^{\downarrow 38.07\%}$	0.6163$^{\downarrow 38.37\%}$
	Structure 7	**0.5970**$^{\downarrow 40.30\%}$	**0.6007**$^{\downarrow 39.93\%}$

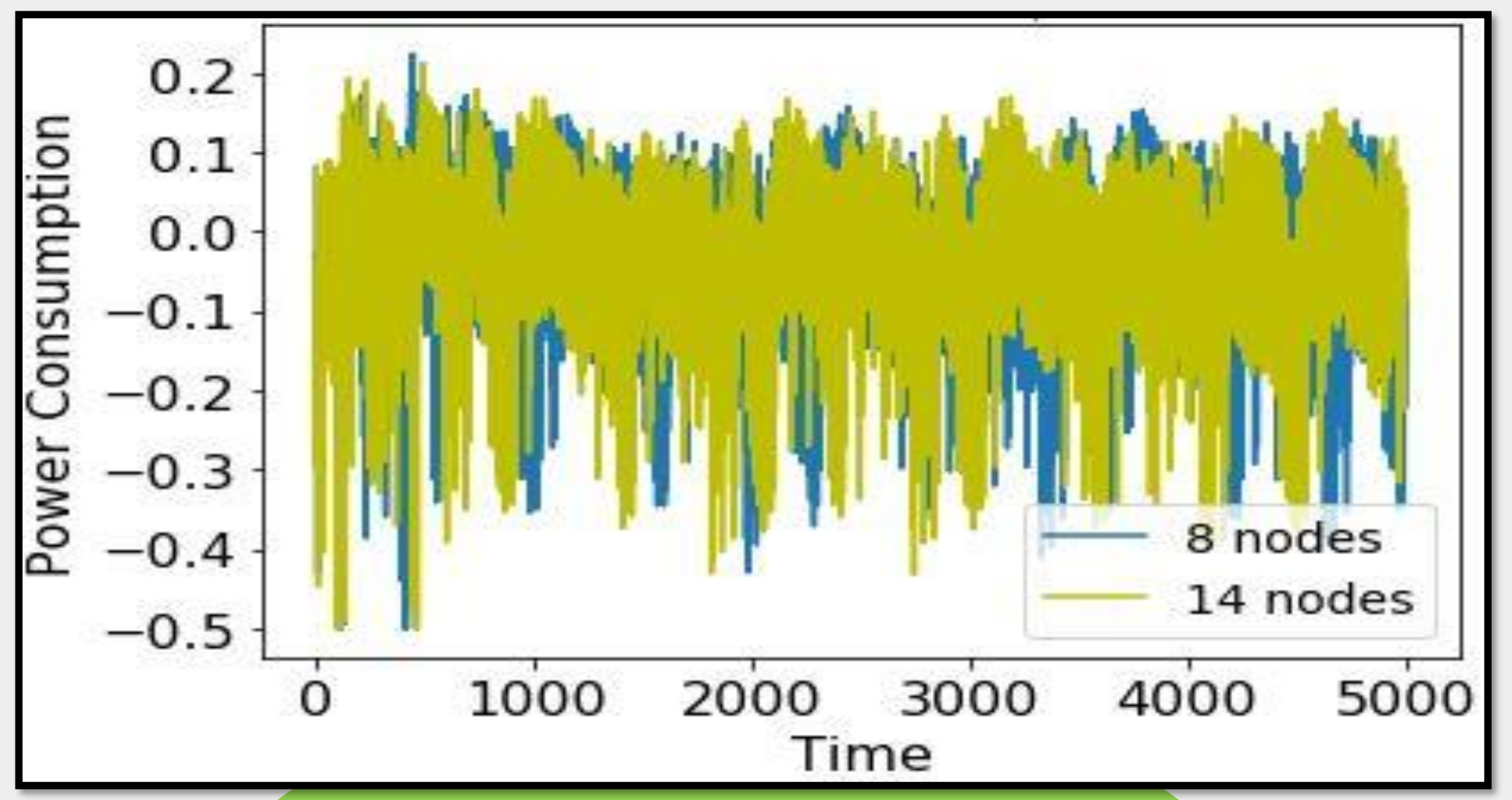
Models	Structures	k=3	k=6
Traditional DNN	(10,10)(14,14)	**1.0000**	**0.9983**
$\mathrm{PD^2NN}$	Structure 6	0.5890$^{\downarrow 41.10\%}$	0.5993$^{\downarrow 39.97\%}$
	Structure 7	**0.4630**$^{\downarrow 53.70\%}$	**0.4667**$^{\downarrow 53.25\%}$

Note: Lower value is better. ↓ denotes the decreasing percentages compared with the traditional DNN.

Second finding

- $\mathrm{PD^2NN}$ decreases privacy inference accuracy significantly compared with the traditional DNNs.
- The inference accuracy is **higher** when the difference of the number of hidden nodes for two neural networks is **larger**.

Figure 4. Power traces of two neural networks.



Third finding

- Power traces are classifiable when the numbers of hidden nodes for two neural networks are different.

REFERENCES

[1] S. Wolf, H. Hu, R. Cooley, and M. Borowczak, "Stealing machine learning parameters via side channel power attacks," in 2021 IEEEComputer Society Annual Symposium on VLSI (ISVLSI). IEEE, 2021,pp. 242–247.

[2] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in International conference on machine learning. PMLR, 2013, pp. 1058–1066.

[3] Pima Indians Diabetes Database | Kaggle: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.

[4] COMPAS Recidivism Racial Bias | Kaggle: https://www.kaggle.com/datasets/danofer/compass.