Focus is Key to Success

A Focal Loss Function for Deep Learning-Based Side-Channel Analysis

Maikel Kerkhof[1], Lichao Wu[1], Guilherme Perin[1,2] and Stjepan Picek[2, 1]

[1]*Delft University of Technology, The Netherlands*
[2]*Radboud University, The Netherlands*

TU Delft

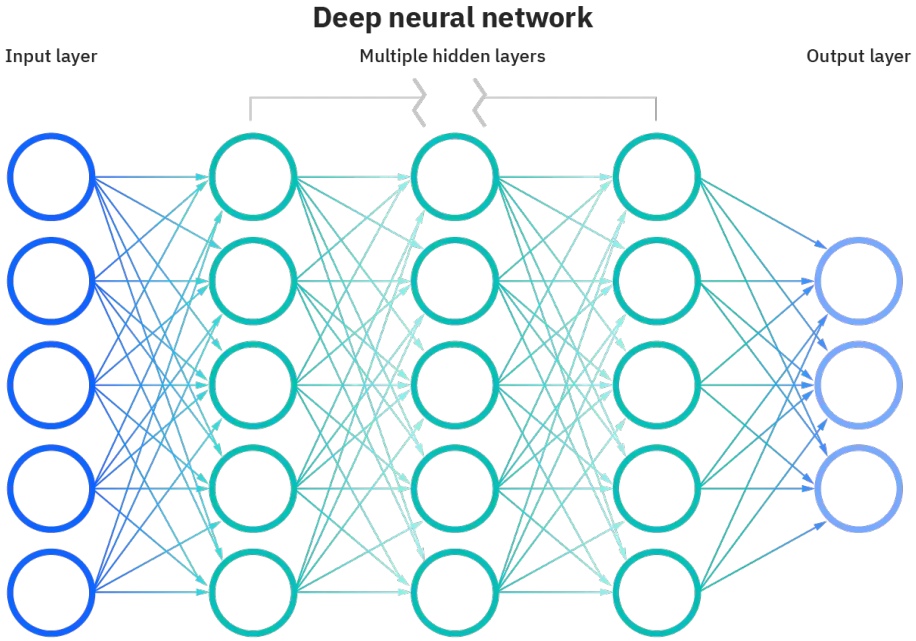
Introduction

- Loss functions
  - Estimate the prediction error
  - Derivatives are used to update weights
  - Key role in learning process

- Example: mean absolute error (MAE)

$$\boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ \hat{\boldsymbol{y}} = \begin{bmatrix} 0.3 \\ 0.5 \\ 0.2 \end{bmatrix}$$

$$\frac{|0-0.3|+|1-0.5|+|0-0.2|}{3} = 0.33$$



Introduction

- Mean absolute error
  - $MAE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$
- Mean squared error
  - $MSE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
- Categorical cross-entropy
  - $CCE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{i=1}^{n} y_i \log \hat{y}_i$

Loss Functions in SCA

- Zhang et al. (2020), Cross Entropy Ratio (CER)
  - Maximize the loss with incorrect predictions

$$cer(\mathbf{y}, \hat{\mathbf{y}}) = \frac{CE(\mathbf{y}, \hat{\mathbf{y}})}{\frac{1}{n}\sum_{i=1}^{N} CE(\mathbf{y}_{r_i}, \hat{\mathbf{y}})}$$

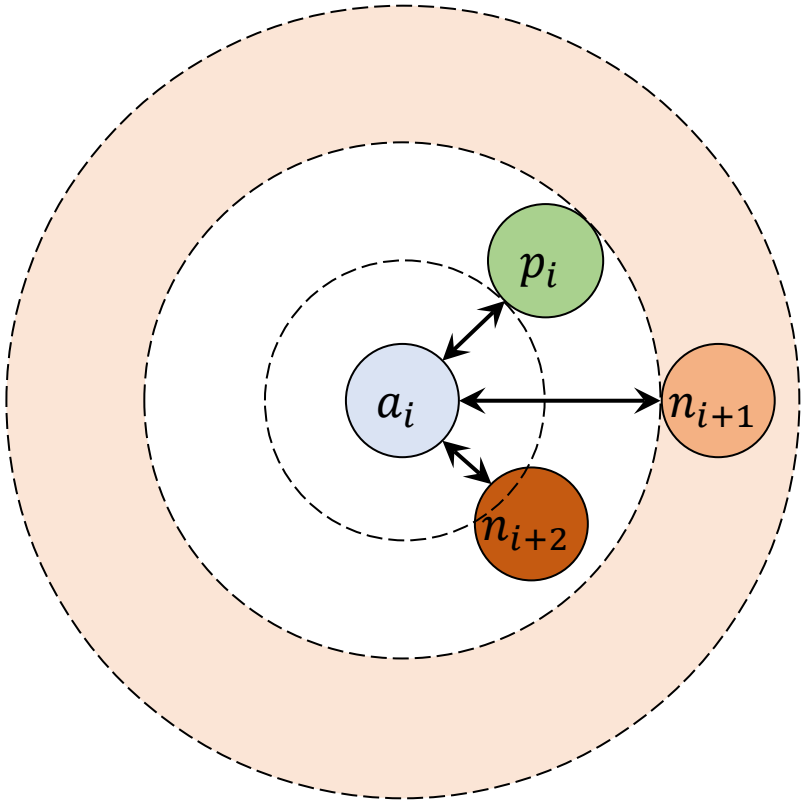
- Zaid et al. (2020), Ranking loss (RKL)
  - Maximize the rank difference of the correct key and the other key bytes

- Kerkhof et al. (2021)
  - Broad analysis of loss function performance
  - Multiple architectures, datasets, leakage models
  - CER performs well in many settings

$$rkl(\mathbf{s}) = \sum_{\substack{k \in \mathcal{K} \\ k \neq k^*}} \left(\log_2\left(1 + e^{-\alpha(s(k^*) - s(k))}\right)\right)$$

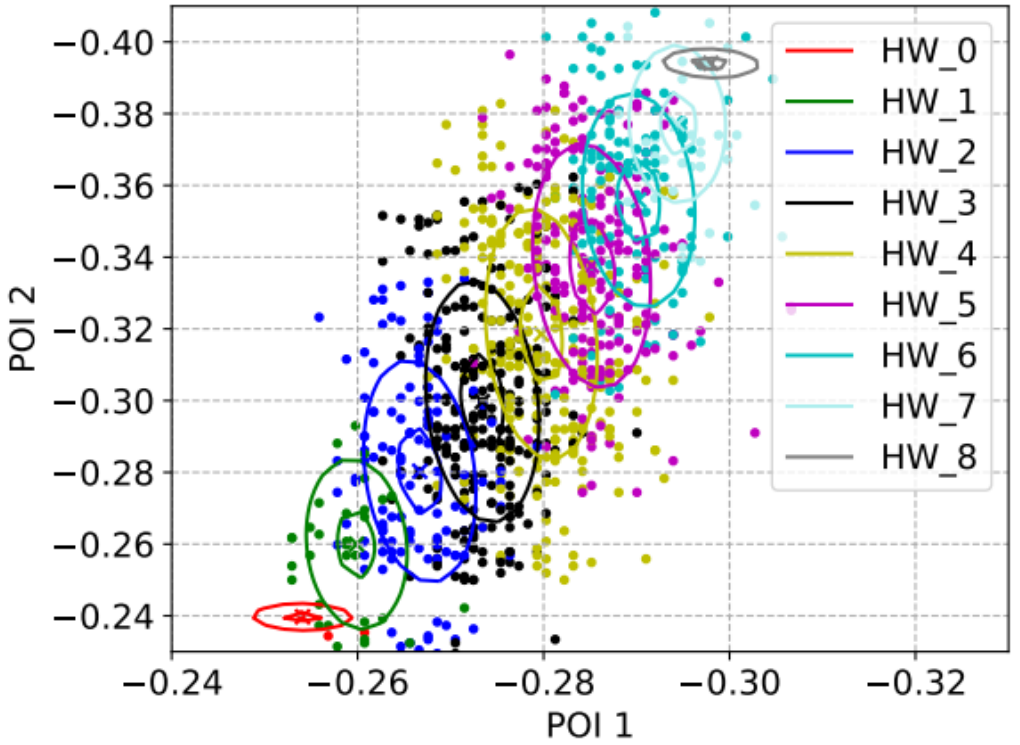
Motivation: learn from the hard samples

- Easy positives/negatives (n_{i+1}):
  - samples classified as positive/negative examples.
  - The bias introduced by easy samples makes it difficult for a network to learn rich semantic relationships from samples
- Hard positives/negatives (n_{i+2}):
  - samples misclassified as negative/positive examples.



Motivation: fight with the imbalanced data

- Middle classes (i.e., HW=4) is over-represented compared to the other classes



Focal Loss Ratio

- Focal Loss Ratio (FLR)
  - Combines both objectives into a loss single function

$$FLR(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{-\boldsymbol{\alpha}(1-\hat{\boldsymbol{y}})^{\gamma} CCE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} -\boldsymbol{\alpha}(1-\hat{\boldsymbol{y}})^{\gamma} CCE(\boldsymbol{y_s}, \hat{\boldsymbol{y}})}$$

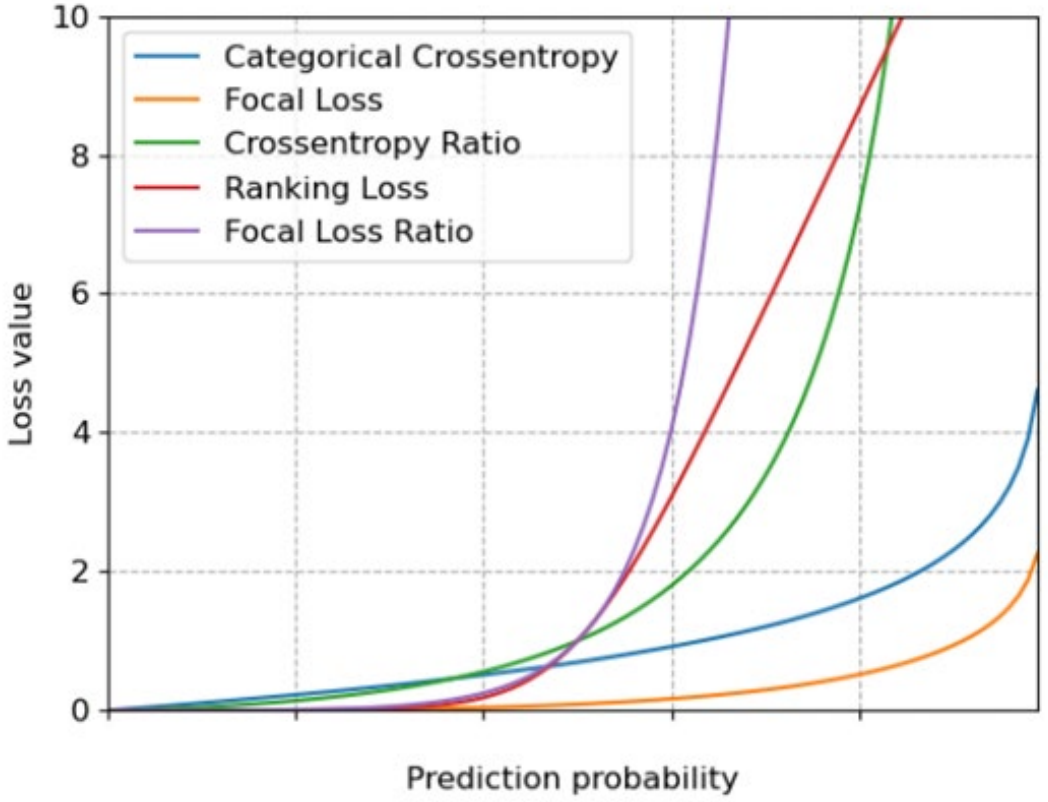
  - $\boldsymbol{\alpha}$: weight vector for each classes
  - $\boldsymbol{\gamma}$: attention level on hard examples

Focal Loss Ratio

- Focal Loss Ratio (FLR)
  - Combines both objectives into a loss single functio

$$FLR(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{-\boldsymbol{\alpha}(1-\hat{\boldsymbol{y}})^{\gamma} CCE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\sum_{i=1}^{n} -\boldsymbol{\alpha}(1-\hat{\boldsymbol{y}})^{\gamma} CCE(\boldsymbol{y_s}, \hat{\boldsymbol{y}})}$$

  - $\boldsymbol{\alpha}$: weight vector for each classes
  - $\boldsymbol{\gamma}$: attention level on hard examples



Performance Evaluation Strategy

1: Generate, train, and test 100 models sampled from range *S* with loss function *L*.

2: Select the best performing model T_b.

3: Train and test the model T_b 10 times.

4: Select the median performing model T_{bm}.

5: Evaluate T_{bm} with evaluation metrics: GE and SR.

Experimental Setup

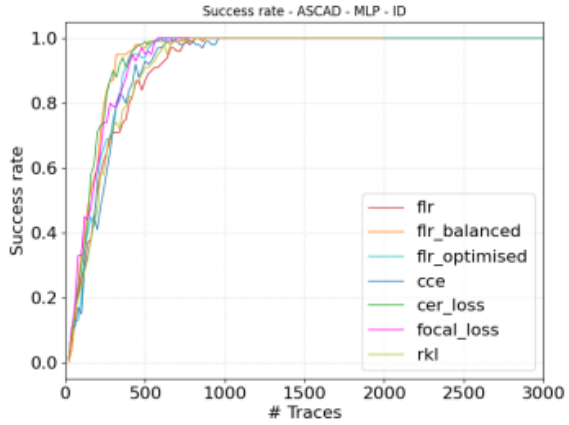
- Focal Loss Ratio (FLR) tuning strategy
  - FLR: fixed value: $\alpha = 0.25$ and $\gamma = 2.0$;
  - FLR_optimized: optimized via random search;
  - FLR_balanced: determined by the sample number of each class

- Searching range for MLP and CNN:

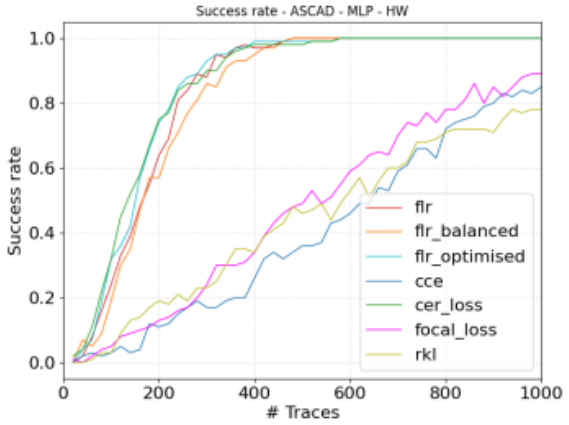
Hyperparameter	**Options**
Dense layers	2 to 8 in a step of 1
Neurons per layer	100 to 1 000 in a step of 100
Learning rate	1e-6 to 1e-3 in a step of 1e-5
Batch size	100 to 1 000 in a step of 100
Activation function (all layers)	ReLU, Tanh, ELU, or SeLU
Loss function	RMSprop, Adam

Hyperparameter	Options
Convolution layers	1 to 2 in a step of 1
Convolution filters	8 to 32 in a step of 4
Kernel size	10 to 20 in a step of 2
Pooling type	Max pooling, Average pooling
Pooling size	2 to 5 in a step of 1
Pooling stride	2 to 10 in a step of 1
Dense layers	2 to 3 in a step of 1
Neurons per layer	100 to 1 000 in a step of 100
Learning rate	1e-6 to 1e-3 in a step of 1e-5
Batch size	100 to 1 000 in a step of 100
Activation function (all layers)	ReLU, Tanh, ELU, or SeLU
Loss function	RMSprop, Adam

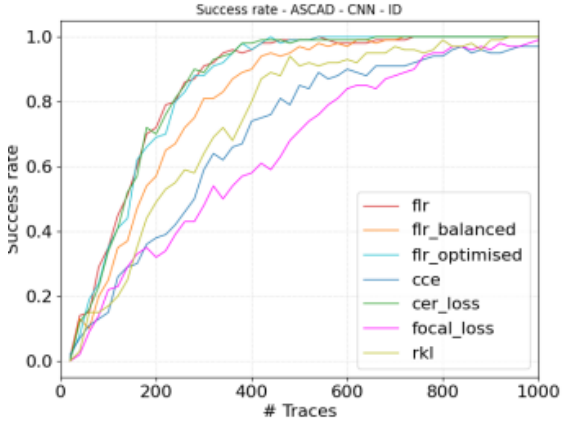
Experimental Results (ASCADf)



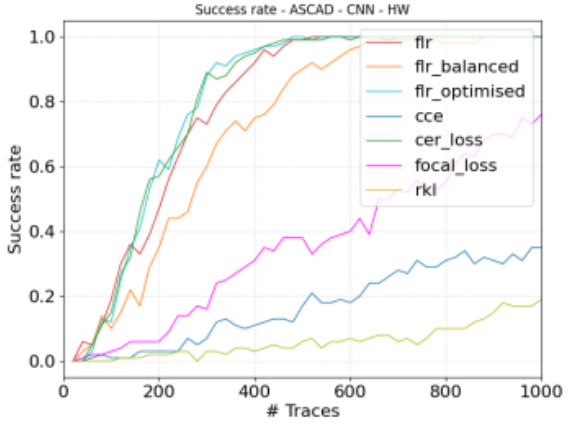
(a) MLP models, ID leakage model.



(b) MLP models, HW leakage model.



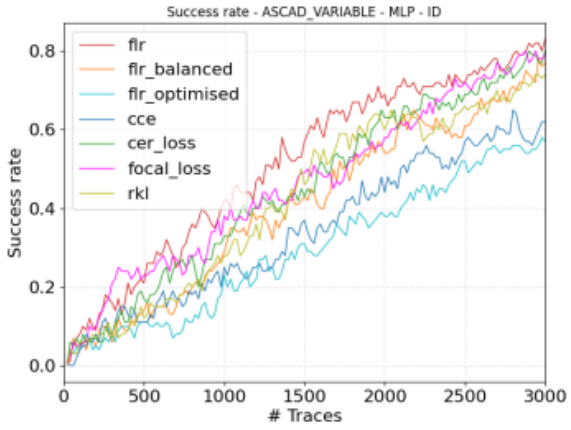
(c) CNN models, ID leakage model.



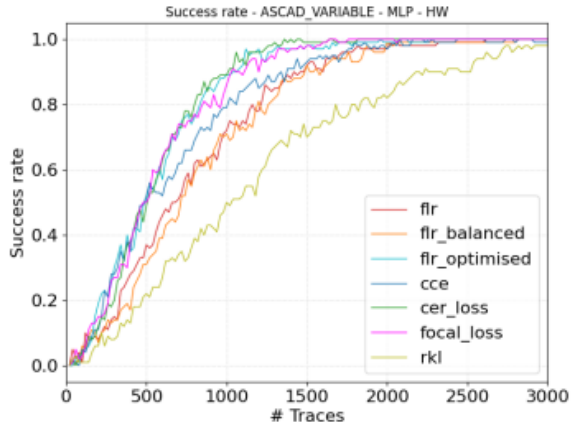
(d) CNN models, HW leakage model.

	L_{focal}	CCE	CER loss	FLR
MLP ID	580	860	570	640
MLP HW	1480	1560	560	490
CNN ID	1250	1360	600	520
CNN HW	1840	>2000	540	500

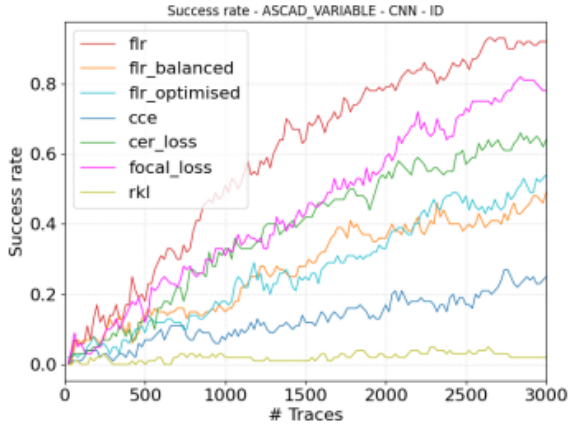
Experimental Results (ASCADr)



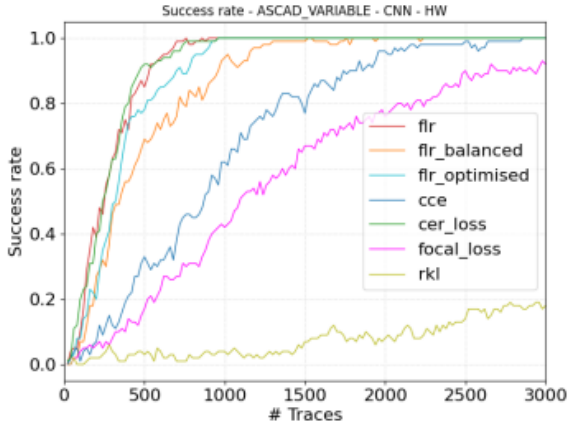
(a) MLP models, ID leakage model.



(b) MLP models, HW leakage model.



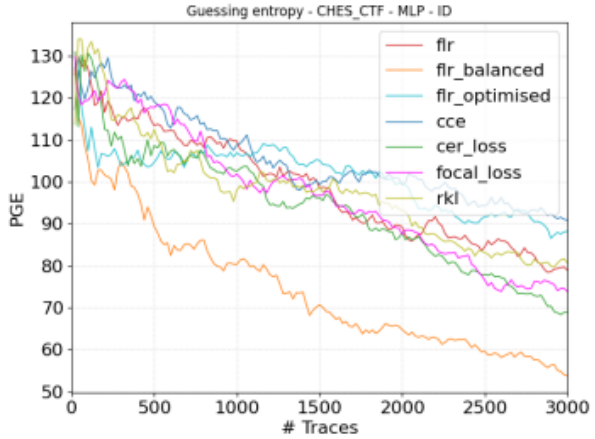
(c) CNN models, ID leakage model.



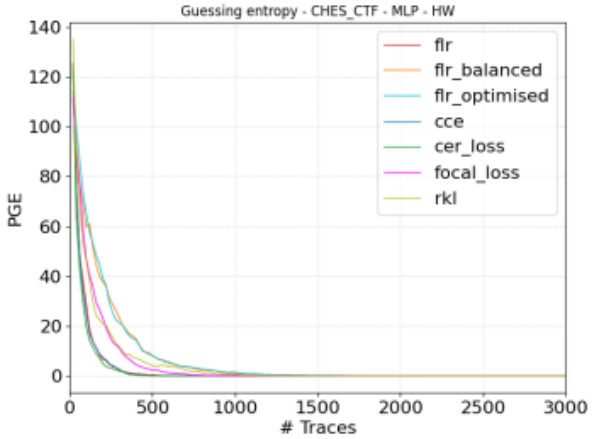
(d) CNN models, HW leakage model.

	L_{focal}	CCE	CER loss	FLR
MLP ID	>3000	>3000	>3000	>3000
MLP HW	1940	2600	1340	1340
CNN ID	>3000	>3000	>3000	>3000
CNN HW	>3000	2840	950	800

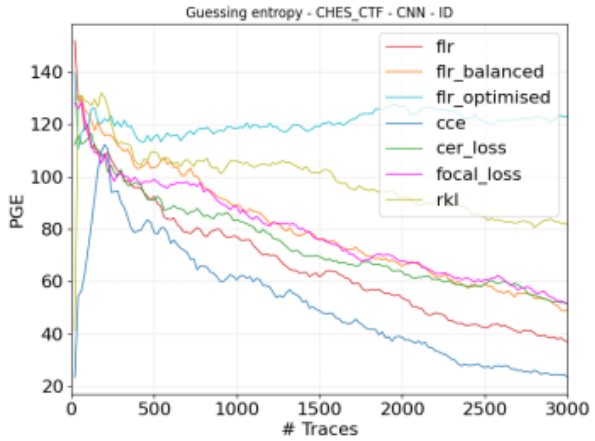
Experimental Results (CHES CTF)



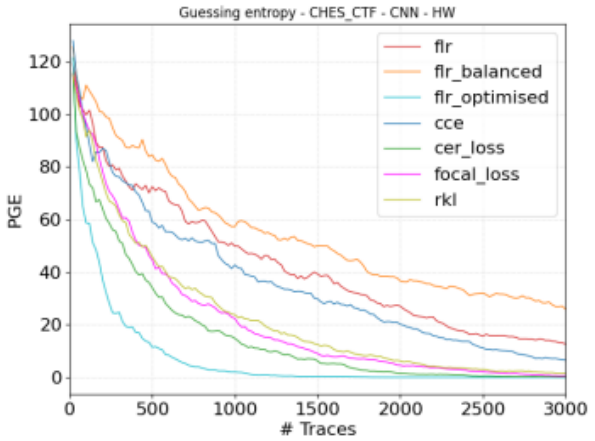
(a) MLP models, ID leakage model.



(b) MLP models, HW leakage model.



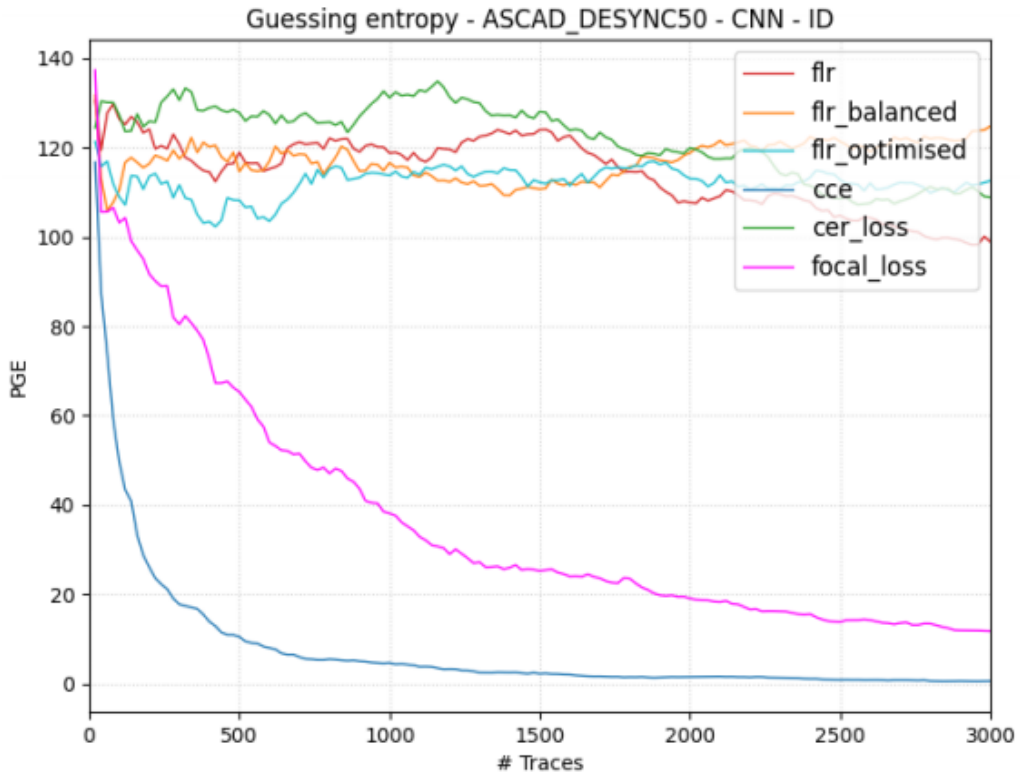
(c) CNN models, ID leakage model.

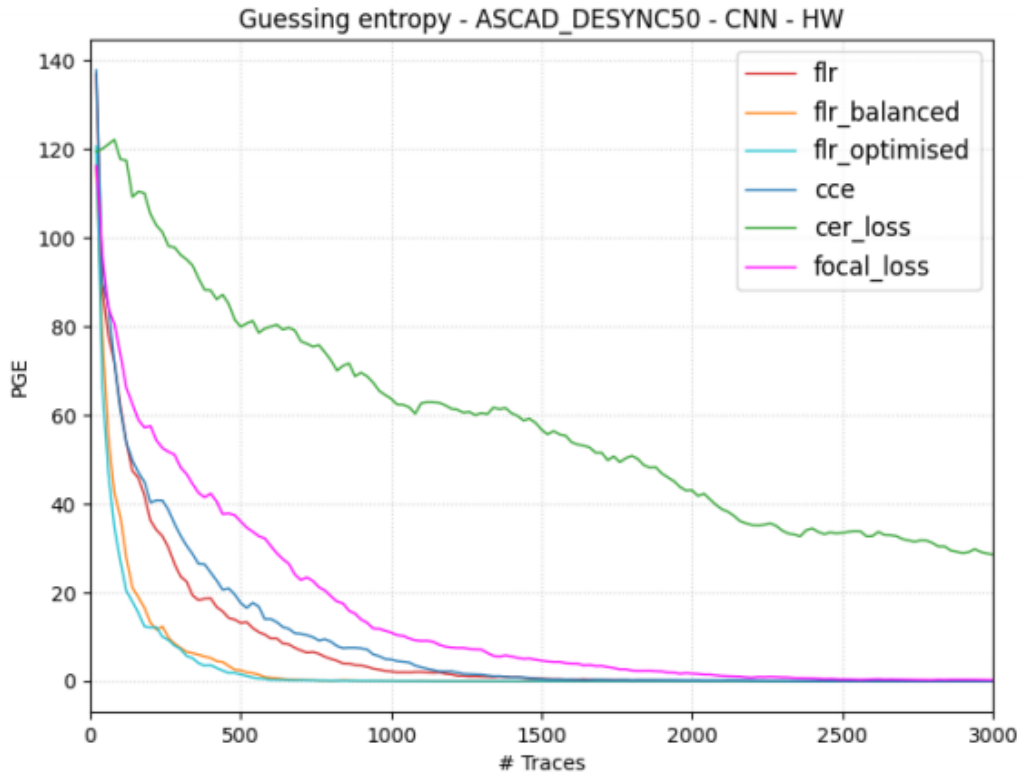


(d) CNN models, HW leakage model.

	L_{focal}	CCE	CER loss	FLR
MLP ID	>3000	>3000	>3000	>3000
MLP HW	1220	630	480	1080
CNN ID	>3000	>3000	>3000	>3000
CNN HW	>3000	>3000	>3000	2070

Experimental Results with Misalignment





Conclusion

- FLR is able to learn from the hard samples and deal with the class imbalance
- FLR loss performs well in various test scenarios, while it still requires hyperparameter tuning
- When using FLR loss, it is a good option to start with fixed hyperparameter (i.e., one used in the paper), then start tuning

- For future work, it would be interesting to explore other optimization strategies
- Develop loss functions based on SCA metrics

Focus is Key to Success

A Focal Loss Function for Deep Learning-Based Side-Channel Analysis

Maikel Kerkhof, Lichao Wu, Guilherme Perin & Stjepan Picek

11-04-2022

TU Delft