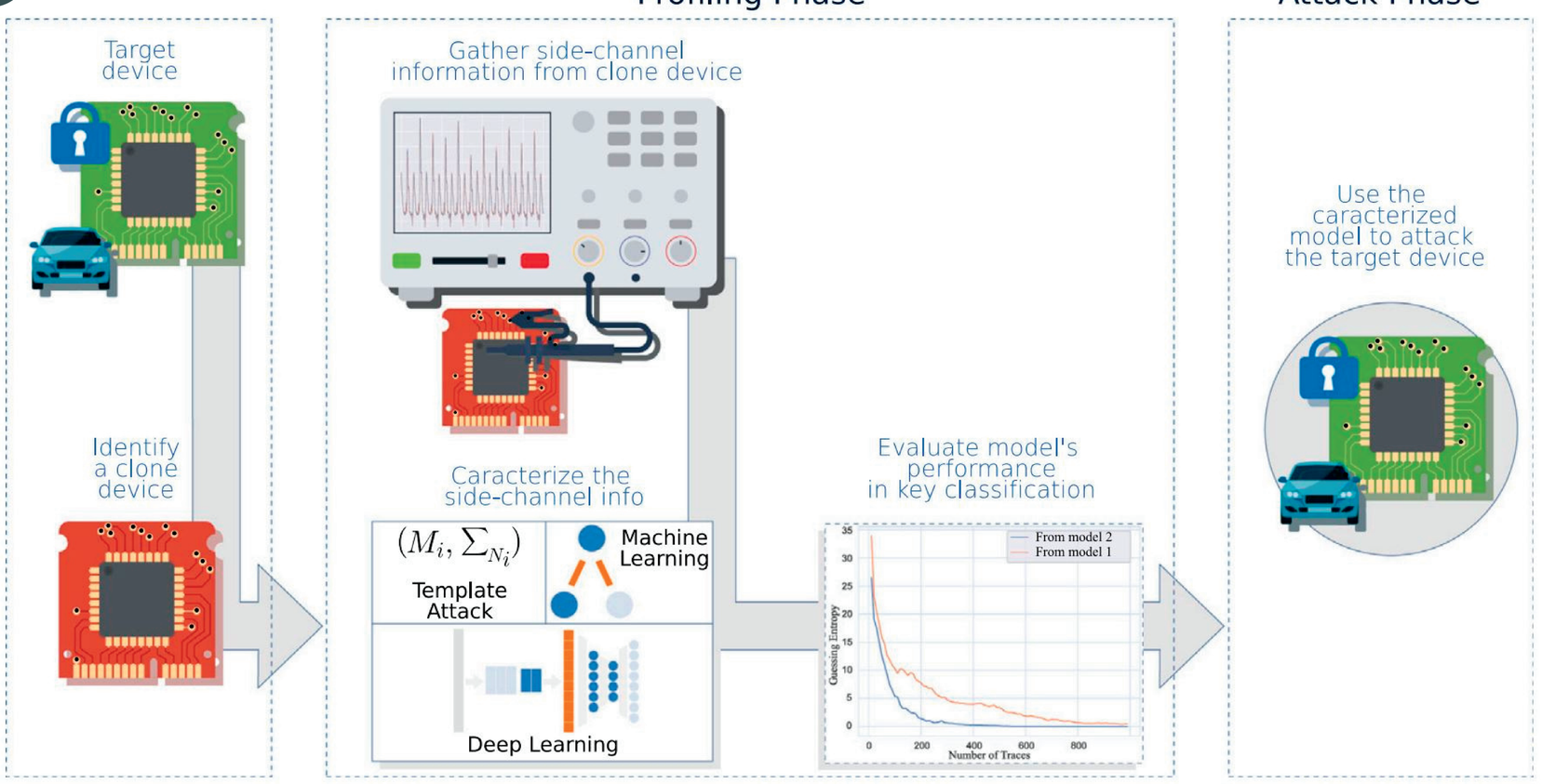
Data Under Siege: The Quest for the Optimal Convolutional Autoencoder in Side-Channel Attacks

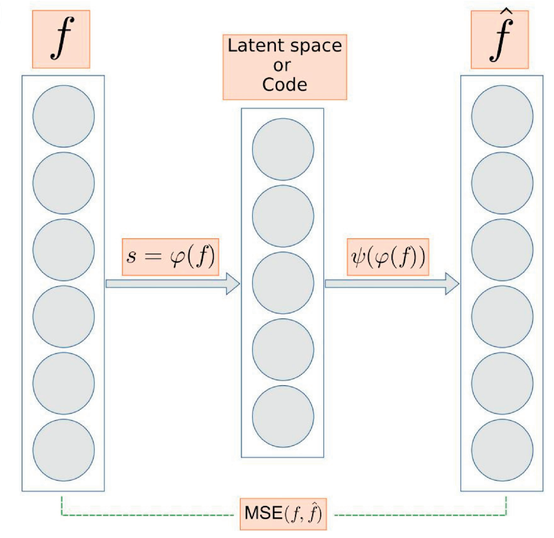
Danny van den Berg (University of Amsterdam), Tom Slooff (Università della Svizzera italiana), Marco Brohet (University of Amsterdam), Kostas Papagiannopoulos (University of Amsterdam) and Francesco Reggazzoni (University of Amsterdam & Università della Svizzera italiana)

1



Overview of a Profiled Side-Channel Attack [1]. Firstly a clone of the target device is acquired. Subsequently, power measurements are collected from this clone while performing cryptographic operations. Next using these measurements an attack model is trained. Finally, the model is used to attack the target device. Additionally, in this paper a convolutional autoencoder (CAE) is added to preprocesses the measurements before the attack model is trained.

2



Autoencoders [1] are designed to remove noise. The autoencoder consists of an encoder which maps the input space to the latent space and a decoder which reconstructs the input space from the latent space. In this paper the encoder consists of convolutional layers and the decoder consists of deconvolutional layers. In the middle there are two fully connected layers, of which the smallest resembles the latent space.

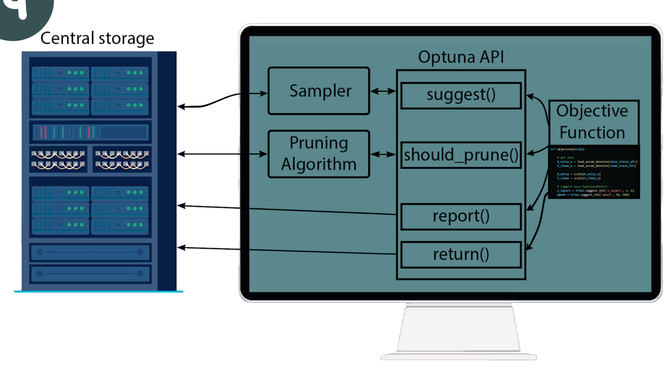
3

An overview of the hyperparameters ranges used in this work, restricted only by time and hardware.

TABLE I: The hyperparameter search space

Hyperparameter	Range
Epochs	1-200
Activation function	ReLU, SeLU
Optimization function	RMSprop, SGD, Adam
Latent space (nodes)	1-700
Total convolutional layers	1-12
Number of filters	1-256
$\prod$ Pool size	1-350
Batch size	128
Kernel size	2-16
Dilation	1-4
Training set	40 000
Validation set	10 000

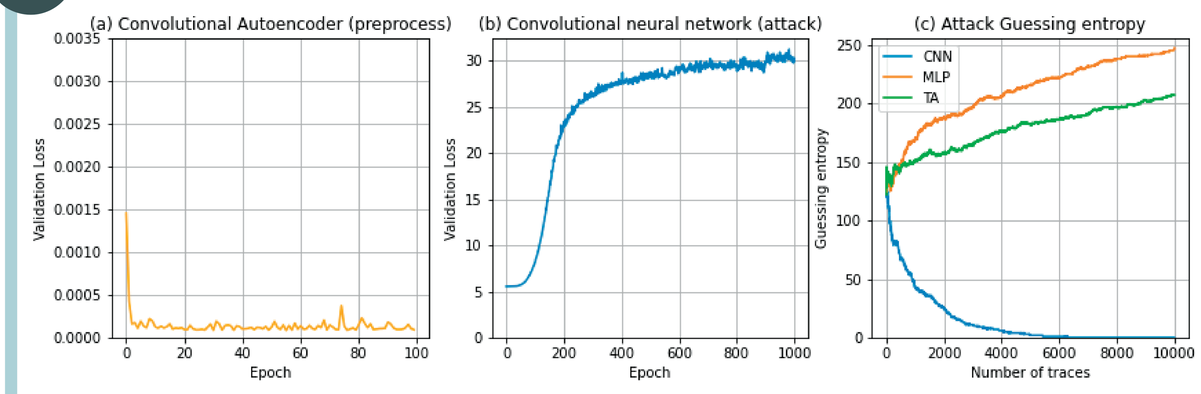
4



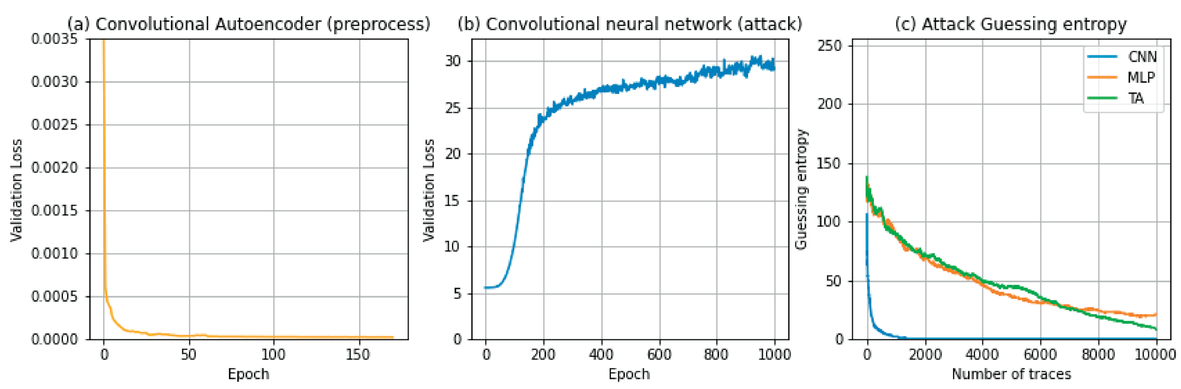
To search through the hyperparameter space, Optuna was used. Optuna [2] is a framework which uses a sampler and a pruner to efficiently search through the hyperparameter space. The sampler determines which hyperparameter settings to try, given the previous results. The pruner determines determines if a trial is promising and thus should be continued. In this paper the TPE sampler and the median pruner were used.

5

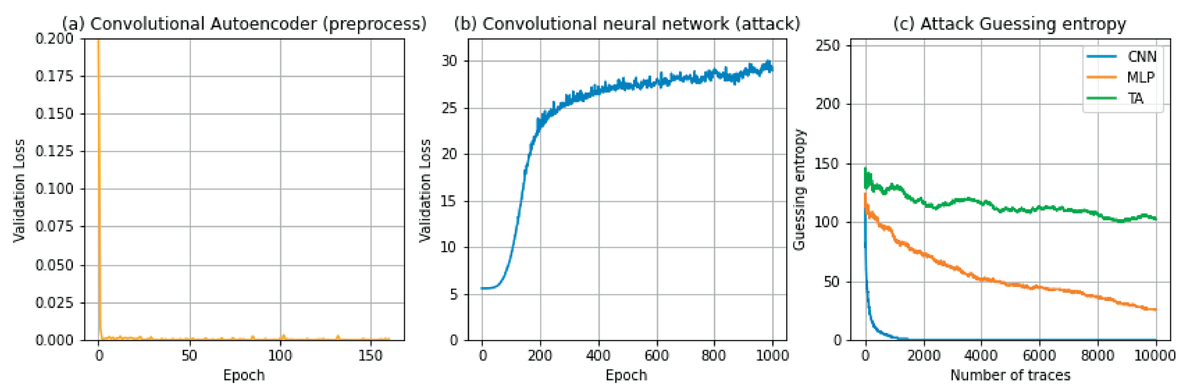
1. Results for the countermeasure Gaussian noise



2. Results for the countermeasure Desynchronization



3. Results for the countermeasure Random Delay Interrupts



The CAE found for Gaussian noise performed better than previous works needing only around 6000 traces for a Guessing Entropy (GE) of zero.The Desynchronization CAE performed similar to previous works needing only 1300 traces, however, it has only 97 000 trainable parameters compared to 2.8 million in previous work. The Random delay interrupts CAE is both in performance and size comparable to previous works needing about 1500 traces for a GE of zero.

Conclusion

- The best optimizer for CAEs in this setting is **Adam**
- Both the **Relu** and **Selu** work well as activation function depending on the countermeasure
- **One layer** is sufficient for Desynchronization, the other countermeasures require **six layers**, or possibly more
- For all countermeasures the **kernel size** is nearly **sixteen** which is at the upper boundary of our search space
- Performance in relation to epochs is very unstable, but **100-170 epochs** can be used as a rule of thumb
- More complex countermeasures such as Gaussian Noise and Random Delay Interrupts require more complex CAEs
- Nearly steady validation loss during training suggests no increase in performance, however, **guessing entropy does decrease**
- This suggests important **attack features getting emphasized** more by the autoencoder, but the traces do not resemble raw traces more as training continues
- The method used to find well-performing CAEs **was successful** and can be used by other researchers to automate the hyperparameter search
- The **hyperparameters found** can be used by other researchers to suggest boundaries or where to start searching in the hyperparameter space

[1] S. Paguada, L. Batina, and I. Armendariz, "Toward practical autoencoder-based side-channel analysis evaluations | Elsevier Enhanced Reader," Elsevier, 2021. [2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework.," Optuna, 2019.