Advances in Applied Mathematics 应用数学进展, 2022, 11(12), 8769-8783
Published Online December 2022 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2022.1112924

Hans汉斯

基于Stacking的侧信道攻击

刘　阳*，王　燚

成都信息工程大学网络空间安全学院，四川　成都

收稿日期：2022年11月21日；录用日期：2022年12月15日；发布日期：2022年12月23日

摘　要

传统侧信道模版攻击已经被基于神经网络的侧信道攻击给代替，一个原因是因为模板攻击假设能耗泄露是服从多元高斯分布，但这种分布并不适合某些情况。并且模板攻击需要构建方阵，会出现矩阵非满秩也就是其不可逆的问题进而影响计算，另外一个原因是不能有效攻击高阶防御。当前有学习的侧信道攻击研究主要聚焦于对单个神经网络模型的结构进行修改或者采用数据增强来提高攻击效果，模型通过局部搜索进行工作，往往陷入局部最优状态。本文提出了一种基于Stacking的侧信道攻击新方法，该方法通过多个不同结构初级模型结合，堆叠出的新模型性能要优于初级模型。本文实施了三个实验来对该方法进行测试，实验一和二是分别在ASCAD v1和ASCAD v2上的攻击，两个实验结果显示Stacking模型在一阶成功率为1时的所需攻击曲线数比初级模型至多减少了66.7%，其攻击效果证明了Stacking方法的普适性。在前面实验基础上进一步对Stacking测试了其采用同样结构初级模型的效果，其效果与初级模型持平甚至还要次于初级模型，所以初级模型在选择上应该更加偏向于不同结构的，使各个模型有着一定差异化。这些实验的结果证明了Stacking通过结合不同结构的神经网络模型最终的攻击效果是明显优于初级模型的。

关键词

模板攻击，集成学习，Stacking

Stacking-Based Side-Channel Attacks

Yang Liu*, Yi Wang

School of Cyberspace Security, Chengdu University of Information Technology, Chengdu Sichuan

Received: Nov. 21st, 2022; accepted: Dec. 15th, 2022; published: Dec. 23rd, 2022

Abstract

The traditional side channel template attack has been replaced by the side channel attack based

*通讯作者。

文章引用: 刘阳, 王燚. 基于 Stacking 的侧信道攻击[J]. 应用数学进展, 2022, 11(12): 8769-8783.
DOI: 10.12677/aam.2022.1112924

on the neural network. One reason is that the template attack assumes that the energy leakage is subject to the multivariate Gaussian distribution, but this distribution is not suitable for some situations. Moreover, template attacks need to build a square matrix, which will lead to the problem that the matrix is not full rank, that is, its irreversibility, and then affect the calculation. Another reason is that it cannot effectively attack high-order defense. The current research on learning side channel attack mainly focuses on modifying the structure of a single neural network model or using data enhancement to improve the attack effect. The model works through local search and often falls into a local optimal state. This paper proposes a new side channel attack method based on Stacking. This method combines several primary models with different structures, and the performance of the stacked new model is better than the primary model. Three experiments are carried out to test this method. Experiment 1 and Experiment 2 are attacks on ASCAD v1 and ASCAD v2 respectively. The results of the two experiments show that the number of attack curves required by the Stacking model when the first order success rate is 1 is at most 66.7% less than that of the primary model. The attack effect proves the universality of the Stacking method. On the basis of the previous experiments, we further tested the effect of Stacking using the primary model with the same structure, and its effect is equal to or even inferior to the primary model. Therefore, the primary model should be more biased towards different structures in selection, so that each model has a definite difference. The results of these experiments prove that Stacking's final attack effect by combining neural network models with different structures is obviously superior to the primary model.

Keywords

Template Attack, Integrated Learning, Stacking

1. 引言

在密码学中，密码算法在硬件设备中运行的时候，硬件设备的电路会泄漏出与密码加密相关的侧信道信息(电磁辐射、能量功耗和执行时间等)，侧信道攻击就是利用这些泄漏出来的侧信道信息来进行分析和破译密码信息。1996 年 Kocher [1]通过测量 RSA、RC5 以及 DSSH 等密码算法运行时的时间序列成功攻破这些密码算法，在此之后，侧信道攻击逐渐成为密码分析技术的一个研究分支，慢慢受到了密码学界的关注。根据侧信道攻击的学习方式，又分为无学习的侧信道攻击和有学习的侧信道攻击。有学习的侧信道攻击需要一个假设前提，即攻击者手上拥有一块和待攻击设备相同的密码设备，他们能完全地控制设备甚至是设置其内部密码算法的密钥信息。有学习的侧信道攻击分为模版攻击[2]、随机攻击和基于机器学习的攻击。

侧信道攻击中模板攻击是攻击效果最强的一种攻击，但模板攻击也有着很明显的缺点，因为建模采用的多元函数高斯分布，某些时候泄漏模型是不符合多元函数高斯分布，这也给破解密钥的成功率带来了很大的影响。如果能迹点也就是特征点过多的话会造协方差矩阵计算效率过低，内存占用多大，并且可能产生奇异矩阵，导致协方差矩阵无法求逆问题。并且模板攻击在对高阶防御中需要先攻击出或者已知能迹对应的掩码，然后根据各能迹掩码分组计算掩码模板，这种攻击方法的条件不易达到。

另外一种研究思路是根据模板攻击和机器学习分类算法共同点，使用机器学习替换模板攻击。机

器学习把能耗信息向量化并作为数据的特征，对应的中间值作为标签输入给模型。基于机器学习的方法用于处理高维度数据通常是有用的，且其的学习特点能够对数据的特征提取有着不小的优势。Hospodar [3]第一个把机器学习应用到侧信道分析中，其实验中使用了不同核函数的支持向量机，并与模板攻击进行了性能比较，实验表明了支持向量机的分类性能更好，证明了机器学习在侧信道中的潜力。首次把机器学习引入到侧信道分析中，在他的实验中用了支持向量机来与模板攻击做了分类性能比较，结果表明使用 RBF 核函数的支持向量分类性能要更好。Lerman [4]在 Hospodar 基础上对比了机器学习和模板攻击在无用点数量增加的情况下的变化，它的实践表明了基于机器学习的侧信道攻击在收到错误影响较多时候更具有吸引力，机器学习对数据分布的学习不再局限于多元函数高斯分布，而是有着更宽更广的视野，同时其有着更强的拟合能力。当前应用于侧信道攻击的机器学习模型有随机森林、支持向量机、神经网络等。随机森林根据决策树提取特征，多个决策树就能提取到多个特征，分类结果由多数决策树输出的分类结果决定，支持向量机利用超平面来对数据进行划分，而面对线性不可分的数据时候，支持向量机使用核函数把在当前维度不可划分的数据映射到可划分的较高维度，之后在进行分类。神经网络每一层都能对数据特征进行提取，层数越深提取到的特征维度越高，同时通过前向传播和反向传输不断修改网络中的参数强化有用的特征，丢弃无用的特征，最终拟合分类结果。神经网络是基于机器学习的侧信道攻击的主要研究方法神经网络的自动学习和非线性识别能力，使得其理论上来说可以拟合出各种分布，一方面它没有了模板攻击局限于服从多元函数高斯分布的缺点，另一方面降低了对高阶防御攻击的条件，不需要提前获取掩码和特定的泄露区域。Prouff [5]首次将深度学习引入侧信道。并在实验中表明在 AES 无防护和有防护的情况下，深度学习相较于模板攻击和机器学习有着更优秀的攻击效果。随后 Prouff 引入了公共开放数据库 ASCAD [6]为以后的研究提供了方便。

当前对神经网络的侧信道攻击的研究主要聚焦于对神经网络单个模型结构的修改或者采用数据增强方法增加数据的数量以提高鲁棒性，但实际提升后效果与原有效果差距不大，单模型往往会陷入局部极小值，最坏的情况是模型甚至没有对数据进行学习。使用 Stacking 来结合多个神经网络模型相对单个模型有三个好处：第一，从统计上看，需要学习的映射关系假设空间往往很大，对于当前学习器所处位置来说可能至少两个的假设位于统一性能，单模型容易因为错误的选择而使得模型最终泛化性能不佳；第二，从计算上看，模型往往会陷入局部最小，某些局部最小点相对应的泛化性能可能是不佳的，但通过多次进行结合，可降低陷入不佳局部最小点的风险；第三，从表示的方面来看，某些模型的假设空间可能没有包含真实的映射，那么这时候单个模型就无法解决问题，而通过结合多个模型后，相应的假设空间就得到了扩大，可能会包含真实的映射或更好的近似。

2. 背景知识

2.1. 模板攻击

侧信道攻击中模板攻击是攻击效果最强的一种攻击。它的优势在于攻击时候只需要相较于其他方法极少量的能迹数量就可以攻击出密钥。它依赖于攻击者需要拥有与被攻击设备完全相同的设备，攻击者可以对这个设备的密钥进行自由更改，通过输入不同明文和密钥来进行加密，同时捕获中间处理过程中产生的能量消耗来刻画对应的模板。但值得注意的是，模板攻击是假设能迹的信息特征电压和噪声这些随机变量的分布是服从多元函数高斯分布的。

模板攻击主要有两个阶段构成，第一个阶段由兴趣点选取和模板构建组成，第二个阶段是模板匹配，该阶段是真正对密钥实施攻击。模板攻击流程如下图 1 所示。

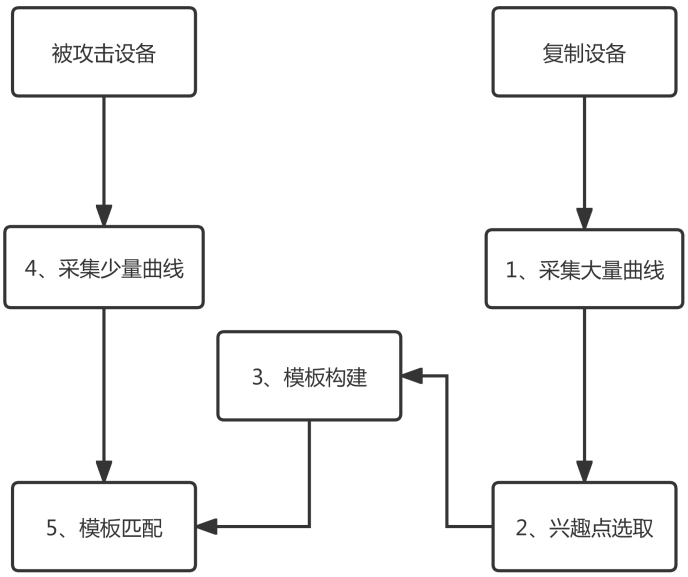


Figure 1. Template attack process
图 1. 模板攻击流程

2.1.1. 兴趣点选取

兴趣点是泄露信息最多的点，不同信息所产生的能耗的差异越大，那么所泄露的信息就越多，所以兴趣点选择就是选取那些表现出能耗差异最大的点，这些选择出的兴趣点就构成了能迹向量，这个步骤是模板攻击成功攻击的关键。一是可以准确挑选出泄露信息高的兴趣点；二是模板构建需要建立协方差矩阵，协方差矩阵的建立是依赖与兴趣点的数量的，当兴趣点过多可能部分兴趣点位置是相邻的，这些兴趣点之间相关性较高，导致协方差矩阵列向量相关，造成计算过程中产生奇异矩阵导致矩阵无法求逆的问题。主要的兴趣点选取方法有以下几种：兴趣点提取方法主要有以下几种：SOD (Sum of Pairwise Differences)、SOSD (Sum of Squared Pairwise Differences)、SOST (Sum of Squared Pairwise T-differences)、皮尔森相关系数方法[7]、主成分分析(PCA) [8]方法和正则化类间差(NICV) [9]。

NICV 使用了像明文和密文这些公开信息来测量泄露程度因此计算占用空间较小，并且其包含了随机变量 X 和 Y 的所有可能模型的相关系数的最大值。所以本文在之后实验中使用了 NICV 作为泄露信息高低可视化的工具。

$$\mathrm{NICV} = \frac{\mathrm{Var}\left[\mathrm{E}\left[Y \mid X\right]\right]}{\mathrm{Var}\left[Y\right]} \tag{1-1}$$

其中，X 是明文或密文的一字节，Y 是测量得到的能耗。

2.1.2. 模板构建

要建立多元函数高斯分布的模板就需要大量的能迹曲线作为基础，模板构建阶段攻击者利用设备，把随机明文 d_i 和假设密钥 k_j 进行组合后对设备进行加密操作，并用数字示波器把加密过程中的电压变化记录下来作为能迹，然后把与(d_i 和 k_j)相对应的能迹进行分组，估计出均值向量 $\mathbf{m}$ 以及协方差矩阵 $\mathbf{C}$ 来表达每个模板，一个完整的模板 T 的具体表达形式如下：

$$\mathrm{T} = \left\langle \mathbf{m}, \mathbf{C} \right\rangle \tag{1-2}$$

在拥有了分组好的能迹数据后就可以分别对每个模板计算出其对应的均值向量 $\mathbf{m}$ 和协方差矩阵 $\mathbf{C}$。

1) 均值向量的计算

假设攻击者使用猜测密钥对明文进行 m 次加密操作，并用仪器测量加密过程中产生的能耗，那么就得到了 m 条能迹曲线，每条能迹曲线包含 n 个采样点，这样就得到了一个 m 行 n 列的多维矩阵，$t_i^j\left(1\le i\le m;1\le j\le n\right)$ 表示为第 i 个条曲线的第 j 个采样点。之后对全部的 m 条能迹求取平均值，其中第 i 次为 $\mathbf{m}_i$，计算公式如下：

$$\mathbf{m}_j=\frac{\sum_i^m t_i^j}{m} \tag{1-3}$$

将所有的平均值合并为均值向量 M 如下：

$$\mathbf{m}=\begin{bmatrix}\mathbf{m}_1\\ \mathbf{m}_2\\ \vdots\\ \mathbf{m}_j\end{bmatrix}=\begin{bmatrix}\frac{1}{m}\sum_{i=1}^m t_i^1\\ \frac{1}{m}\sum_{i=1}^m t_i^2\\ \vdots\\ \frac{1}{m}\sum_{i=1}^m t_i^j\end{bmatrix} \tag{1-4}$$

2) 协方差矩阵的计算

每条能迹曲线的噪声向量如下：

$$\mathbf{N}_i=\begin{bmatrix}t_i^1-\mathbf{m}_1\\ t_i^2-\mathbf{m}_1\\ \vdots\\ t_i^j-\mathbf{m}_1\end{bmatrix} \tag{1-5}$$

把所有的能迹曲线组成的噪声向量组合成矩阵，公式如下：

$$\mathbf{N}=\begin{pmatrix}t_1^1-\mathbf{m}_1 & t_1^2-\mathbf{m}_2 & \cdots & t_1^j-\mathbf{m}_j\\ t_2^1-\mathbf{m}_1 & t_2^2-\mathbf{m}_2 & \cdots & t_2^j-\mathbf{m}_j\\ \vdots & \vdots & \ddots & \vdots\\ t_i^1-\mathbf{m}_1 & t_i^2-\mathbf{m}_2 & \cdots & t_i^j-\mathbf{m}_j\end{pmatrix}=\left(\mathbf{N}_1,\mathbf{N}_2,\cdots,\mathbf{N}_i\right) \tag{1-6}$$

那么该矩阵每一行就对应每条能迹的噪声向量，每列对应每个采样点的噪声向量。

之后两两噪声向量计算对应的协方差，可得到下列公式：

$$\operatorname{cov}\left(\mathbf{N}_u,\mathbf{N}_v\right)=\frac{1}{m-1}\sum_{i=1}^m\left(t_i^u-\mathbf{m}_u\right)\left(t_i^v-\mathbf{m}_v\right) \tag{1-7}$$

式中的 t_i^u 和 t_i^v 分别代表了 $\mathbf{N}_u$ 与 $\mathbf{N}_v$ 中的第 i 个分量。

把上式得到的所有协方差合并为协方差矩阵，如下：

$$\mathbf{C}=\begin{pmatrix}\mathbf{C}\left(\mathbf{N}_1,\mathbf{N}_1\right) & \mathbf{C}\left(\mathbf{N}_1,\mathbf{N}_2\right) & \cdots & \mathbf{C}\left(\mathbf{N}_1,\mathbf{N}_j\right)\\ \mathbf{C}\left(\mathbf{N}_2,\mathbf{N}_1\right) & \mathbf{C}\left(\mathbf{N}_2,\mathbf{N}_2\right) & \cdots & \mathbf{C}\left(\mathbf{N}_2,\mathbf{N}_j\right)\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{C}\left(\mathbf{N}_i,\mathbf{N}_1\right) & \mathbf{C}\left(\mathbf{N}_i,\mathbf{N}_2\right) & \cdots & \mathbf{C}\left(\mathbf{N}_i,\mathbf{N}_j\right)\end{pmatrix} \tag{1-8}$$

均值向量和协方差矩阵得到后就建立了其对应的模板，之后按照上面的步骤依次建立其他猜测密钥对应的模型即可。

2.1.3. 模板匹配

模板匹配阶段需要提前对真实设备中采集真实密钥加密产生能耗进行采集，获得真实的能迹曲线，得到的能迹也需要进行兴趣点选取。把得到的能迹向量输入到模板的概率密度函数中得到概率，公式如下：

$$\mathrm{p}\left(\mathbf{t};\left(\mathbf{m},\mathbf{C}\right)d_i,k_j\right)=\frac{\exp\left(-\frac{1}{2}\cdot\left(\mathbf{t}-\mathbf{m}\right)'\cdot\mathbf{C}^{-1}\cdot\left(\mathbf{t}-\mathbf{m}\right)\right)}{\sqrt{\left(2\cdot\pi\right)^{T}\cdot\det\left(\mathbf{C}\right)}} \tag{1-9}$$

使用能迹与每个模型进行计算，得到概率值也不同，这些不同的概率值一定程度揭示了能迹与模板的匹配程度。根据极大似然判定准则，根据式(1-9)模板匹配过程中得到的概率值最高的模板，其对应的猜测密钥是被攻击设备的正确密钥。

$$\mathrm{p}\left(\mathbf{t};\left(\mathbf{m},\mathbf{C}\right)d_i,k_j\right)>\mathrm{p}\left(\mathbf{t};\left(\mathbf{m},\mathbf{C}\right)d_i,k_l\right),\forall l\neq j \tag{1-10}$$

上面公式求得都是一条能迹与模板的概率计算，实际上能迹数量是多条的，假设攻击能迹数为 m 条，把所有攻击能迹匹配模板得到的概率相乘可以得出联合概率，公式如下：

$$\mathrm{p}\left(k_j\right)=\prod_{j=1}^{m}\mathrm{p}\left(\mathbf{t};\left(\mathbf{m},\mathbf{C}\right)d_i,k_j\right) \tag{1-11}$$

最后根据前面计算的每个模板的联合概率，选取出概率最高的模板，其对应的猜测密钥很大程度上就是正确密钥。

$$\hat{k}=\arg\max_k \hat{P}\left(k_j\right) \tag{1-12}$$

2.2. 基于神经网络模型的侧信道攻击

模板攻击根据不同的中间值来对复制设备采集到的能迹进行分组，每个中间值会建立对应的模板，并用被攻击设备采集到的真实密钥对应的能迹与各个模板进行匹配得到概率值，最后根据极大似然法则选择模板匹配期间得到概率值最大的模板，其对应的猜测密钥往往是被攻击设备出正确的密钥。模板攻击的模板构建和模板匹配是类似于神经网络的输入训练数据生成模型和输入测试数据获得数据分类结果，基于这种情况，神经网络是能与侧信道分析进行结合的。

与模板攻击不同的是，神经网络需要提前对能迹进行向量化处理，之后对标签进行 One-Hot 编码，处理过后的数据格式才符合神经网络需要的数据格式。同时因为侧信道攻击中中间值对应的密钥可能性是不止两个，例如 AES128 对应的一个 Sbox 盒输出的密钥可能为 256 个，所以对应分类问题就是多分类问题。并且神经网络最后一层会得到每个类别的得分，而这个得分一般需要限制在表示概率大小的有效实数空间，往往需要经过一层 Softmax 函数处理，使得输出的所有分类概率值和为 1。为了模型输出的预测值可能为负值，采用指数处理使得预测值大于 0。

$$\mathrm{Softmax}\left(z_i\right)=\frac{\mathrm{e}^{z_i}}{\sum_{c=1}^{C}\mathrm{e}^{z_c}} \tag{1-13}$$

其中 i 表示输出节点的编号，z_i 为对应的输出节点。

神经网络训练模型的本质在于依据最小化损失函数来更新模型参数以获得更高的预测性能。在多分类中，交叉熵损失函数[10]是用来衡量两个概率分布的相似性，所以前面 Softmax 函数预测输出的概率分布与真实类别的 One-Hot 形式的相似性自然就可以用交叉熵损失函数来求出，并以此来更新我们神经网络模型的权重矩阵和偏置向量等参数。

交叉熵损失函数表达式如下：

$$L=-\sum_{c=1}^{C} y_c \log\left(p_c\right) \tag{1-14}$$

其中 c 为对应的输出节点编号，y_c 为真实样本的预测值，p_c 为经过 Softmax 计算的概率值。

2.2.1. 多层感知器

多层感知器(Multilayer Perceptron，缩写 MLP) [11]是一种前向结构的神经网络，映射一组输入向量到一组输出向量，多层感知器由多层节点组成，前一层节点与后一层节点是全连接的。多层感知器经典架构由输入层、隐层和输出层三层构成。中间的隐层层数不固定根据情况来变化。其经典结构如图 2 所示。

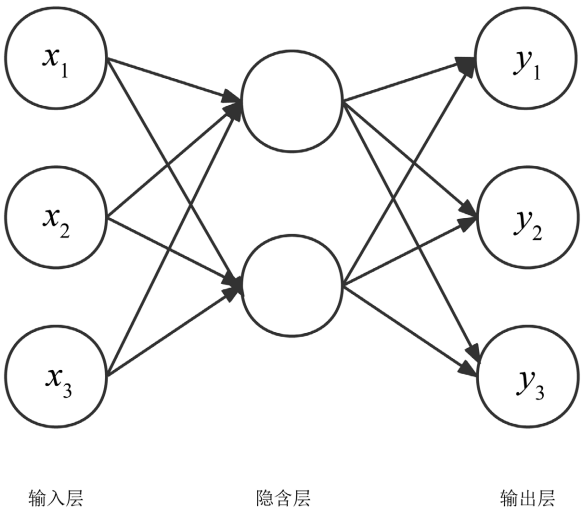


Figure 2. Multilayer perceptron structure
图 2. 多层感知器结构图

多层感知器层与层之间关系公式如下所示：

$$\begin{cases}\mathbf{y}_l^{(j)}=f\left(\mathbf{u}_l^{(j)}\right)\\ \mathbf{u}_l^{(j)}=\sum_{i\in L_{i-1}}\mathbf{W}_l^{(ji)}\mathbf{y}_{l-1}^{(i)}+\mathbf{b}_l^{(j)}\\ \mathbf{y}_l=f\left(\mathbf{u}_l\right)=f\left(\mathbf{W}_l\mathbf{y}_{l-1}+\mathbf{b}_l\right)\end{cases} \tag{1-15}$$

这里我们假设输入层神经元值为$\left[x_1,x_2,x_3\right]$，神经元与神经元之间的偏置为$\left[b_1,b_2,b_3\right]$。假设将 f 设为激活函数，某一层为 l，那么该层所有神经元用 $\mathbf{L}_l$，该层输出就为 $\boldsymbol{y}_l$，其中第 j 个节点输入为$\mathbf{u}_l^{(j)}$，输出为$\mathbf{y}_l^{(j)}$)，第 l 与第$(l-1)$层中间的权重矩阵为 $\mathbf{W}_l$。

多层感知器在不添加激活函数的假设空间是线性空间，如果仅仅是线性空间，网络求出的解就只能局限于线性空间，为了扩大假设空间，引入非线性变换的激活函数，网络拥有了非线性的特性，这样神经网络就可以在非线性空间中寻找解了。常用的激活函数有 Sigmoid、整流线性单元(ReLU)和可缩放指数型线性单元(SeLU)等。

基于多层感知器的侧信道攻击[11]，中间隐层对输入的能迹进行处理，每个神经元能迹泄露的信息特征有效保留，并将学习到的权重和偏置往后传输影响之后的每一层神经元处理，前层的神经元会接受后层的神经元传输过来的误差信号对其学习到的权重和偏置进行修正，促使其保留更全面的泄露特征。

下面为多层感知器在侧信道攻击中的攻击密钥公式：

$$K^{*}=\arg\max_{k}\prod_{j-1,\cdots,m}\mathrm{MLP}\left(e_{j}\mid\mathrm{comb}\left(x_{j},k\right)\right) \tag{1-16}$$

其中 K^{*} 为正确密钥，e_j 为能量消耗曲线，$\mathrm{comb}\left(x_j,k\right)$ 为中间值 $\mathrm{Sbox}\left(x_j\oplus k\right)$，MLP 是分类模型。

但多层感知器有着其缺点，因为所有层都是全连接的，每个神经元都是由前层所有神经元根据权重偏置计算出来的。这意味着随着输入维数的增加，隐层的每个神经元与输入噪声对应的权值都会增加，使得模型更加难以学习到有用特征。换个角度来看，因为在高维输入和大量神经元之间的权重矩阵可以很快耗尽现成的 GPU 的内存，所以全连接的特性也限制了第一隐层中神经元的数量。

2.2.2. 卷积神经网络

卷积神经网络引入了卷积层和池化层。卷积层通过卷积核进行运算将本层输入数据的对应特征提取出来，不同的卷积核提取的特征也不同，卷积层公式简单来描述，假设 $f(n)$ 和 $g(n)$ 是两个不同的信号，则卷积公式为下(1-17)。池化层一般用在卷积层之后降低数据的空间尺寸，空间尺寸的降低也就降低了模型参数量提高了模型的训练速度。在卷积神经网络最后我们需要全连接层作为最终分类层，它接受前面的卷积层和池化层输出的特征数据，最终输出分类的概率分布。卷积操作如下：

$$s(n)=(f*g)[n]=\sum_{m=0}^{n-1}f(m)\cdot g(n-m) \tag{1-17}$$

卷积神经网络在侧信道攻击中与图像分类中不同，图像分类多为三通道二维卷积，而侧信道领域的能迹数据是一维的，所以对应用到卷积也应该为一维卷积，与卷积配套的池化也应该为一维池化。利用卷积层的平移不变形提取能迹数据的局部特征，随后在池化层中提取局部特征中最明显的特征并降低参数量，之后通过全连接把不同卷积核得到的局部特征结合起来，从而刻画了能迹数据泄露的总体特征，最后输出能迹对应的中间值类别的概率分布。卷积网络在侧信道攻击中应用见图 3。

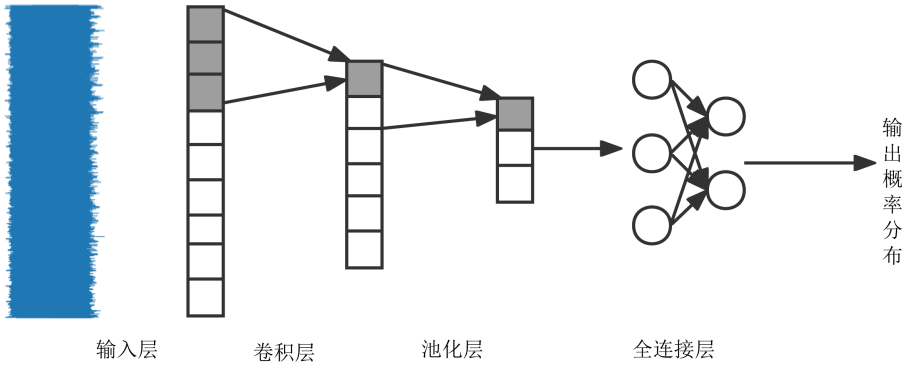


Figure 3. Schematic diagram of a convolutional neural network suitable for side-channel attacks
图 3. 适用于侧信道攻击的卷积神经网络示意图

在侧信道攻击中攻击密钥的公式为：

$$K^{*}=\arg\max_{k}\prod_{j-1,\cdots,m}\mathrm{CNN}\left(e_{j}\mid\mathrm{comb}\left(x_{j},k\right)\right) \tag{1-18}$$

其中 K^{*} 表示正确密钥，$\mathrm{CNN}\left(e_j\mid\mathrm{comb}\left(x_j,k\right)\right)$ 表示训练过程中学习到的概率模型，$\mathrm{comb}\left(x_j,k\right)$ 是中间组合值，e_j 表示能量曲线。在上式中，学习到的概率模型至关重要，他的分类效果好坏会影响到攻击性能的好坏。

卷积神经网络的缺点在于一方面卷积神经网络的最后也是全连接层，可能会陷入多层感知器的问题，另一方面是池化层的存在忽略了位置信息，造成了一定的信息丢失，而位置信息在多数时候也是关键的

信息，如人脸识别等。这一缺点同样也会影响到侧信道分析中的效果，因为侧信道分析中采集的能耗信息是与位置有着相关性的。

3. Stacking

上述研究都聚焦于通过单个模型的结构调整与修改以提高模型攻击性能。有两个方面可以提高模型的攻击性能，一种是数据层面如数据增强来增加数据的数量以提高模型鲁棒性，做法有平移、旋转和改变颜色，另一种模型层面，如调参和模型融合。集成学习是模型融合的另外一个称呼，它是一种机器学习方法，把多种初级模型的输出的概率分布结合起来用以训练新模型。根据结合的方法不同可以分为Boosting、Bagging [12]和 Stacking [13] [14]。清华大学 Mengmeng Xu [15]等人根据投票法来决定模型最终输出结果，这种方法其实本质上来说类似于随机森林，只不过随机森林采用的是决策树构成，因为采用了多种模型来进行训练，分类结果由多个模型共同决定，降低了个别模型分类错误导致模型最终攻击得出错误结果的可能性。Stacking 的好处在于可能会降低因为单模型错误选择而导致的泛化性能不佳、降低陷入糟糕局部极小点的风险和扩大假设空间得到更好的近似。通过多个不同的起点运行的局部搜索构成的集合可以更好的拟合出未知函数。Stacking 因为抵消了由单模型误选而产生的攻击效果不佳、容易陷入局部极小点风险的问题，并且 Stacking 的多模型扩大了假设空间使得模型能更接近目标函数，所以提高了模型整体的泛化能力和鲁棒性。

本文采用了 Stacking 方法应用于侧信道攻击。Stacking 由两个阶段组成，第一阶段选取多个初级模型(模型的数量需要根据实际情况来选择)进行并行独立的训练和预测。把所有初级模型的训练输出的概率分布堆叠起来生成新的训练集，同理把所有初级模型的测试输出的概率分布生成新的测试集。第二阶段把前面生成的新的训练集来训练次级模型，次级模型根据新的测试集输出分类概率分布。

Stacking 模型步骤如下，令 $D=\left\{\left(x_1,y_1\right),\left(x_2,y_2\right),\cdots,\left(x_m,y_m\right)\right\}$，$x_1$ 是能迹向量，y_1 是能迹对应标签。初级模型学习算法为 $L_1,\cdots,L_t$。

过程如下，其中 $t=1,\cdots,T$ 是不同初级模型的编号，每个初级模型建立如下的函数关系式：

$$h_t=L_t \tag{2-1}$$

之后创建新的数据集 $D'=\varnothing$，设置为空集，把 D 中所有的数据输入到每一个初级模型中，来训练初级模型。$i=1,\cdots,m$，x_i 为不同的能迹向量，每个初级模型输出各能迹向量对目标中间值的概率分布。

$$z_t^i=h_t\left(x_i\right) \tag{2-2}$$

对同一数据，所有初级模型把各自输出的概率分布合并为新数据的各个维度，标签仍保持不变，把这新的数据加入到集合 D' 中。

$$D'=D'\cup\left\{\left(\left(z_1^i,z_2^i,\cdots,z_T^i\right),y_i\right)\right\} \tag{2-3}$$

根据以上步骤获得新的训练集来训练次级模型，训练好的次级模型攻击出测试集对应的候选子密钥。

$$K^*=\arg\max_k\prod\mathrm{STACKING}\left(e_j\mid\mathrm{comb}\left(x_j,k\right)\right) \tag{2-4}$$

其中 STACKING 为模型名称。

4. 实验设计与研究

4.1. 实验所用指标

本实验采用成功率和猜测熵来作为实验标准。成功率表示了模型预测能迹曲线得到的猜测密钥排名

向量中，正确密钥排名位于前 o 阶(前 o 名)的概率。阶数越低且概率越高代表了模型的攻击效果越好。本文采用的阶数为一阶和二阶成功率。

成功率定义为：

$$\mathrm{SR}^{o}\left(K^{*}\right)=\frac{\mathrm{num}(o)}{\mathrm{attck\_num}} \tag{2-5}$$

其中 K^{*} 代表正确密钥，其中，o 代表阶数，attack_num 代表攻击次数，num(o)代表在 attack_num 次攻击中，猜测密钥位于前 o 位的次数。

在侧信道领域，数据分类正确结果不是模型最终目标。最终目标是使用模型攻击出密钥的能力，该能力由指标猜测熵来体现。猜测熵表达了正确密钥在模型攻击后得出的概率分布结果排名，猜测熵越低，模型结果中正确密钥排名越靠前。

猜测熵定义为：

$$\mathrm{GE}=\left|K \in k \mid \mathrm{P}(K)>\mathrm{P}\left(K^{*}\right)\right. \tag{2-6}$$

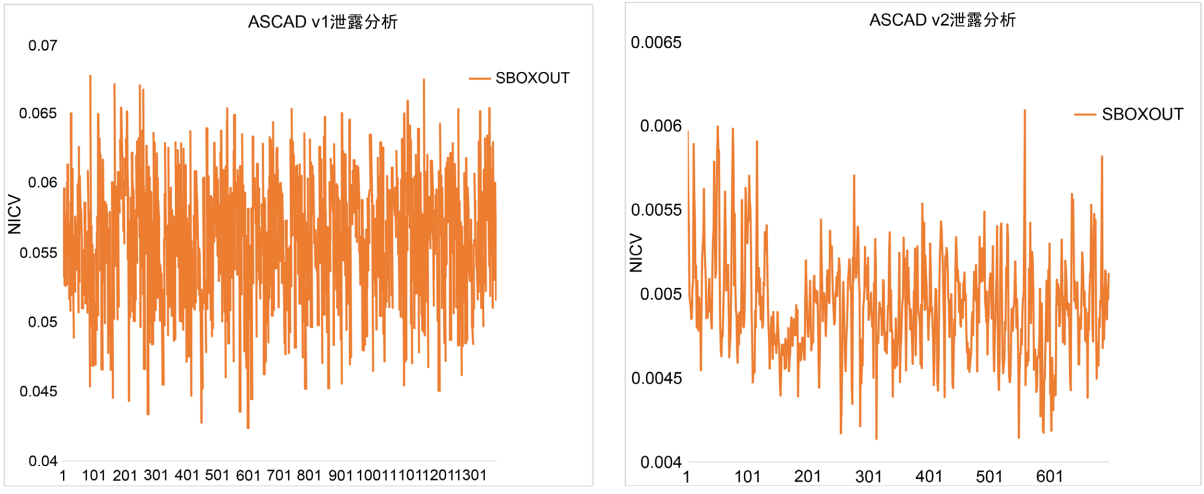
其中 $\mathrm{P}\left(K^{*}\right)$ 代表正确密钥的得分， $\mathrm{P}(K)$ 为猜测密钥的得分。

4.2. 实验环境及数据

实验环境所使用的神经网络框架为Tensorflow 2.6.0，运行在搭载了六块NVIDIAGeForceRTX2080TiGPU的服务器上，服务器版本为Ubuntu20.0.4。

实验数据为 ASCAD [6]，该数据集是 AES 加掩实现的一种电磁信号泄漏数据。采用的 ASCAD 分为两个版本，两个版本的时间样本都过滤掉了无泄漏操作的时间窗口，ASCAD v1 密钥都是相同的，数据总共有 60,000 条能迹曲线，其中训练阶段曲线为 50,000 条，攻击曲线为 10,000 条，每条曲线包含 700 个数据点。ASCAD v2 数据的密钥是不同的，数据共有 300,000 条能迹，其中训练为 200,000 条，密钥是随机，攻击为 100,000，密钥为同一密钥，每条曲线包含 1400 个数据点。

下图 4 左为 ASCAD v1，右为 ASCAD v2 的 NICV 泄漏分析，从泄漏图 4 可以看出，ASCAD 数据的泄漏没有明显的峰值，所以该数据集比较难攻击。



(a) ASCAD v1　　(b) ASCAD v2

Figure 4. NICV leakage analysis
图 4. NICV 泄漏分析

4.3. 实验实施

Stacking 需要谨慎选择所使用到的初级模型和次级模型。在初级模型上面应该尽量选择本身攻击效果就应该不错，并且模型之间有一定的结构差别。而次级模型相对来说尽量不使用卷积这种复杂网络。本文共三组实验，第一组实验是在 ASCAD v1 上测试二个初级模型的 Stacking 模型和三个初级模型组合训练的 Stacking 模型；第二组实验是在 ASCAD v2 上测试二个初级模型的 Stacking 模型和三个初级模型共同组合训练的 Stacking 模型。实验一和实验二分别在 ASCAD v1 和 ASCAD v2 数据集上进行了测试。第三组实验验证初级模型使用相同结构和不同结构的区别。实验所用的初级模型这里我们选择了 MLP(5 层网络结构、学习率为 0.001 和激活函数为 SELU)、VGG16 (参考 ASCAD [6]文章中提出的 CNN_{best}，与后面提到的 CNN 有一定区别)和根据[16]提到的经验设计的 CNN 作为初级模型。而次级模型我们选择全连接层作为网络的基础结构，并且通过网格寻优来找到适合的参数。

实验的次级模型所需超参数搜索空间如下表 1 所示。次级模型采用全连接这种简单网络来构成，输入神经元数量设计思路是 256 的倍数，选择 256 倍数原因在于每个初级模型输出的概率密度分布函数对应的向量大小为 256 × 1，所以把多个初级模型学习到的特征展开后的一维向量就是 256 的倍数，并且依据猜想，特征展开组成的向量应该等于输入神经元个数，这样的话输入端的数据就会尽量得到保留。

Table 1. Secondary model hyperparameters
表 1. 次级模型超参数

超参数	最小值	最大值
学习率	1e−05	1e−03
dropout	0	0.5
输入神经元数量	512	1024
优化器	RMSprop	adam

第一组实验对 ASCAD v1 数据集进行测试，表 2 展示了各个模型的攻击效果。

Table 2. Attack effect of the model in ASCAD v1
表 2. 模型在 ASCAD v1 下攻击效果

模型名称	ASCAD v1					
	猜测熵	攻击能迹数	一阶成功率	攻击能迹数	二阶成功率	攻击能迹数
CNN	0	600	1	700	1	400
VGG16	0	700	1	700	1	500
MLP	2.7	>1000	0.4	>1000	0.5	>1000
VGG16-CNN 的 Stacking	0	200	1	300	1	200
CNN-MLP 的 Stacking	0	500	1	600	1	300
VGG16-MLP 的 Stacking	0	400	1	500	1	300
CNN-VGG16-MLP 的 Stacking	0	200	1	400	1	200

从表中可以发现基于 Stacking 的模型在猜测熵降为 0 时候，所需要的攻击能迹数量是低于三个初级模型的。Stacking 模型相较于初级模型的所需攻击能迹数降幅分别为 66.7%、16.6%、42.8%和 66.7%。同样在一阶成功率和二阶成功率达到 1 时候，Stacking 模型在一阶成功率和二阶成功率达到 1 时候，分别所需攻击能迹数平均比初级模型少了 350 和 380 条。可以发现 VGG16-CNN 的 Stacking 是指标中攻击效果最好的，其原因在于使用的初级模型的攻击效果本身就还不错，而初级模型中带了 MLP 的就要稍差一些，因为 MLP 攻击效果相较于其他两个要差拉低了模型整体性能，如 VGG16-MLP 的 Stacking 比 VGG16-CNN 的 Stacking 在猜测熵、一阶成功率和二阶成功率对应的所需的攻击能迹数量上要分别多 200、200 和 100 条，增幅 33.3%。总的来说 Stacking 模型在 ASCAD v1 数据集上的效果高于其他初级模型的。

第二组实验初级模型及 Stacking 模型在 ASCAD v2 攻击效果如表 3 所示。

Table 3. Attack effect of model under ASCAD v2
表 3. 模型在 ASCAD v2 下攻击效果

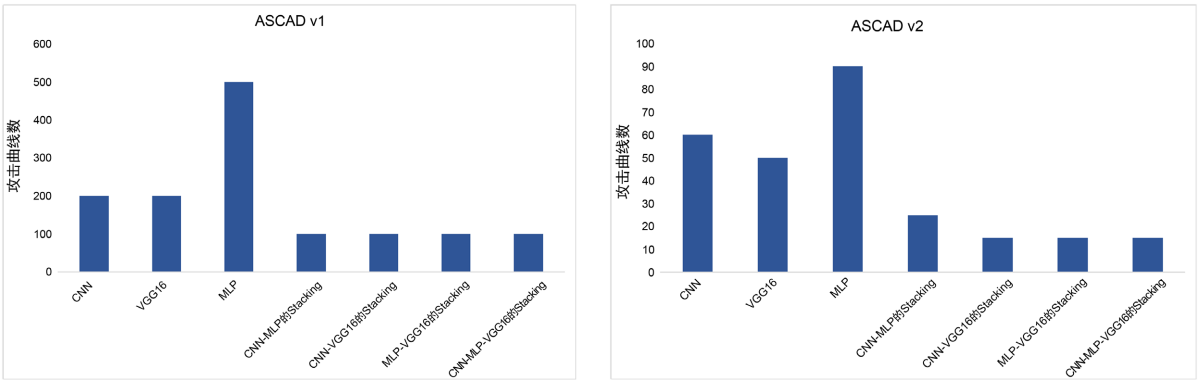
模型名称	ASCAD v1					
	猜测熵	攻击能迹数	一阶成功率	攻击能迹数	二阶成功率	攻击能迹数
CNN	1.5	>100	0.3	>100	0.6	>100
VGG16	0	80	1	80	1	70
MLP	2.2	>100	0.5	>100	0.7	>100
VGG16-CNN 的 Stacking	0	40	1	50	1	50
CNN-MLP 的 Stacking	0	70	1	70	1	60
VGG16-MLP 的 Stacking	0	50	1	50	1	50
CNN-VGG16-MLP 的 Stacking	0	50	1	50	1	40

表中显示了 CNN 和 MLP 都没能在 100 条以内把猜测熵降为 0 和成功率升为 1，那么根据 ASCAD v1 的经验可知道两两组合的 Stacking 模型中同时使用这两个初级模型的必然效果要低于没有使用的，如 VGG16-CNN 的 Stacking 在猜测熵、一阶成功率和二阶成功率上所需攻击能迹数要分别比 CNN-MLP 的 Stacking 少了 30、20 和 10 条，且 VGG16-MLP 也分别少了 20、20 和 10 条。但有意思的是，组合三个初级模型的 Stacking 包含了 CNN-MLP 但在攻击效果上其实不差于其他 Stacking 模型，原因是多个不同初级模型在误差上进行风险降低，并且扩大了模型的假设空间，使得模型在寻找较低局部最小的时候可能性更多。

对于实验一和二，VGG16 在 ASCAD v2 上的效果要好于 ASCAD v1，而 CNN 却是 ASCAD v1 上效果要好于 ASCAD v2 上，原因是两个数据集上的差别和两个模型在卷积核个数上的差别。CNN 初始卷积核个数少于 VGG16，而 ASCADv1 输入的数据点为 700，ASCAD v2 为 1400，卷积核个数多少直观上来说就是提取的不同特征数量，那么用卷积核较多的去提取特征较少的数据，会导致提取到无用特征影响模型攻击效果。而用卷积核较少的去提取特征较多的数据，也会导致部分有用特征被忽略，所以导致了上述的情况。

以图 5 为例，假设候选子密钥队列长度为 4，即只要猜测熵小于 4 就是可以成功的模型。在 ASCAD v1 下，Stacking 模型在 100 条以内就能成功攻击，而其他初级模型至少须要 200 条才能攻击出，MLP 甚至

需要 500 条。而在 ASCAD v2 下，VGG16、CNN 和 MLP 分别需 50 条、60 条和 90 条攻击出子密钥，但 Stacking 模型仅需 10 条左右便可以攻击出子密钥，至少减少了 80%的攻击能迹数。两组实验各自在不同的数据集上的攻击效果证明了 Stacking 方法的有效性和普适性。



(a) ASCAD v1　　(b) ASCAD v2

Figure 5. The number of attack curves required by the model when the entropy is guessed to be 4
图 5. 模型在猜测熵为 4 时所需攻击曲线数

在接下来实验三中主要关注相同结构的模型结合的 Stacking 的效果，数据集只关注 ASCAD v1，比较初级模型使用相同的结构和不同结构是否存在区别。表 4 展示了模型效果。

Table 4. Attack effect of the same primary model under ASCAD v1
表 4. 相同初级模型在 ASCAD v1 下攻击效果

模型名称	ASCAD v1					
	猜测熵	攻击能迹数	一阶成功率	攻击能迹数	二阶成功率	攻击能迹数
CNN	0	600	1	700	1	400
VGG16	0	700	1	700	1	500
MLP	2.7	>1000	0.4	>1000	0.5	>1000
All-VGG16 的 Stacking	3.5	>1000	0.3	>1000	0.4	>1000
All-CNN 的 Stacking	0	600	1	800	1	500
All-MLP 的 Stacking	4	>1000	0.3	>1000	0.4	>1000

由表 4 中可以看出，初级模型采用相同的结构组合的 Stacking 在效果上与初级模型效果基本持平如 All-CNN 的 Stacking 在攻击指标上与初级模型 CNN 本身相近，而有些效果甚至还会低于初级模型如 All-VGG16 的 Stacking 和 All-MLP 的 Stacking 都无法在 1000 条攻击能迹范围内让猜测熵降为 0、一阶和二阶成功率变为 1。实验一和实验三的结果表明了 Stacking 对初级模型的选择上，不同结构是要优于相同结构的，这也正是因为 Stacking 需要模型之间有着一定的差异化，而这种差异化就体现在模型的结构不同和其原理的区别，也可以说是不同模型对同一数据学习到的不同的特征。Stacking 根据挖掘的不同特征来训练出效果更好的模型以提升模型攻击性能。

5. 总结

本论文中首先介绍了模板攻击在处理高维数据和攻击带高阶防御的密码算法时的缺点，随后介绍了

基于神经网络的侧信道攻击能有效解决模板攻击出现的问题同时提到了神经网络自身存在的问题和缺陷。根据目前的现状，提出了使用 Stacking 来进行攻击的方法并介绍了它的优点和步骤。实验中展示了不同初级模型与 Stacking 模型在成功率和猜测熵的对比，实验数据表明了 Stacking 方法有效地提高了模型的攻击能力。进一步对相同初级模型的 Stacking 的效果验证后，发现其效果甚至不如初级模型。通过上述实验最终得出了 Stacking 适用于结合不同初级模型来提升攻击性能，而不是使用相同初级模型。在今后的研究中，将致力于挖掘其在带抖动防御的密码算法中的潜力。

基金项目

四川省科技计划资助(项目号：2021ZYD0011)。

参考文献

[1] Kocher, P., Jaffe, J. and Jun, B. (1999) Differential Power Analysis. In: Wiener, M., Ed., *Advances in Cryptology—CRYPTO* '99. *Lecture Notes in Computer Science*, Vol. 1666, Springer, Berlin, 388-397. https://doi.org/10.1007/3-540-48405-1_25

[2] Chari, S., Rao, J.R. and Rohatgi, P. (2002) Template Attacks. In: Kaliski, B.S., Koç, Ç.K. and Paar, C., Eds., *Cryptographic Hardware and Embedded Systems—CHES* 2002. *Lecture Notes in Computer Science*, Vol. 2523, Springer, Berlin, 13-28. https://doi.org/10.1007/3-540-36400-5_3

[3] Hospodar, G., Gierlichs, B., De Mulder, E., Verbauwhede, I. and Vandewalle, J, (2011) Machine Learning in Side-Channel Analysis: A First Study. *Journal of Cryptographic Engineering*, **1**, Article No. 293. https://doi.org/10.1007/s13389-011-0023-x

[4] Lerman, L., Poussier, R., Bontempi, G., Markowitch, O. and Standaert, F.-X. (2015) Template Attacks vs. Machine Learning Revisited (and the Curse of Dimensionality in Side-Channel Analysis). In: Mangard, S. and Poschmann, A., Eds., *Constructive Side-Channel Analysis and Secure Design*, *Lecture Notes in Computer Science*, Vol. 9064, Springer, Cham, 20-33. https://doi.org/10.1007/978-3-319-21476-4_2

[5] Maghrebi, H., Portigliatti, T. and Prouff, E. (2016) Breaking Cryptographic Implementations Using Deep Learning Techniques. In: Carlet, C., Hasan, M. and Saraswat, V., Eds., *Security, Privacy, and Applied Cryptography Engineering*. *Lecture Notes in Computer Science*, Vol. 10076, Springer, Cham, 3-26. https://doi.org/10.1007/978-3-319-49445-6_1

[6] Benadjila, R., Prouff, E., Strullu, R., Cagli, E. and Dumas, C. (2020) Deep Learning for Side-Channel Analysis and Introduction to ASCAD Database. *Journal of Cryptographic Engineering*, **10**, 163-188. https://doi.org/10.1007/s13389-019-00220-8

[7] Benesty, J., Chen, J., Huang, Y. and Cohen, I. (2009) Pearson Correlation Coefficient. In: *Noise Reduction in Speech Processing*, Springer, Berlin, 1-4. https://doi.org/10.1007/978-3-642-00296-0_5

[8] Batina, L., Hogenboom, J. and Van Woudenberg, J.G. (2012) Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis. In: Dunkelman, O., Ed., *Topics in Cryptology—CT-RSA* 2012. *Lecture Notes in Computer Science*, Vol. 7178, Springer, Berlin, 383-397. https://doi.org/10.1007/978-3-642-27954-6_24

[9] Bhasin, S., Danger, J.-L., Guilley, S. and Najm, Z. (2014) NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage. 2014 *International Symposium on Electromagnetic Compatibility*, Tokyo, 12-16 May 2014, 310-313.

[10] Mannor, S., Peleg, D. and Rubinstein, R. (2005) The Cross Entropy Method for Classification. *Proceedings of the* 22*nd International Conference on Machine Learning*, Bonn, 7-11 August 2005, 561-568. https://doi.org/10.1145/1102351.1102422

[11] Martinasek, Z., Dzurenda, P. and Malina, L. (2016) Profiling Power Analysis Attack Based on MLP in DPA Contest V4.2. 2016 39*th International Conference on Telecommunications and Signal Processing* (*TSP*), Vienna, 27-29 June 2016, 223-226. https://doi.org/10.1109/TSP.2016.7760865

[12] Zaid, G., Bossuet, L., Habrard, A. and Venelli, A. (2021) Efficiency through Diversity in Ensemble Models Applied to Side-Channel Attacks: A Case Study on Public-Key Algorithms. *ACR Transactions on Cryptographic Hardware and Embedded Systems*, **2021**, 60-96. https://doi.org/10.46586/tches.v2021.i3.60-96

[13] Wolpert, D.H. (1992) Stacked Generalization. *Neural Networks*, **5**, 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1

[14] Baćak, V. and Kennedy, E.H. (2019) Principled Machine Learning Using the Super Learner: An Application to Predicting Prison Violence. *Sociological Methods & Research*, **48**, 698-721. https://doi.org/10.1177/0049124117747301

[15] Xu, M., Wu, L. and Zhang, X. (2017) Side Channel Attack on SM4 Algorithm with Ensemble Method. 2017 13*th International Conference on Computational Intelligence and Security*, Hong Kong, 15-18 December 2017, 529-532. https://doi.org/10.1109/CIS.2017.00123

[16] 黄洁，王燚．适用于侧信道分析的卷积神经网络结构的实验研究[J]．成都信息工程学院学报，2019，34(5): 449-456.