# Key Attack Strategies Against Black-Box DNNs

Yassine Hmamouche, Yehya Nasser, Amer Baghdadi, Marc-Oliver Pahl

# Key Attack Strategies Against Black-Box DNNs

Yassine Hmamouche, Yehya Nasser, Amer Baghdadi, and Marc-Oliver Pahl
IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238, France
Emails: {yassine.hmamouche, yehya.nasser, amer.baghdadi, marc-oliver.pahl}@imt-atlantique.fr

**Introduction:** DNNs are enabling major advances in solving hard scientific problems and processing complex data on an unprecedented scale in many areas such as language processing, fraud detection, healthcare, and so on. The design of DNNs for commercial services requires a significant expenditure of time, money, and human effort, from collecting massive data to fine-tuning the model's hyperparameters. Thus, the commercial value of these models makes them important intellectual property for companies, which incentivizes adversaries to mount specific attacks in order to retrieve their internal intelligence, gain knowledge about the sensitive information being processed by them, or at least disrupt their operation by intentionally injecting specific vulnerabilities [1].

These DNN models are typically kept confidential in a black-box setup where the adversary is not privy to the structure or parameters of the model, other than the output predictions for the corresponding input. This setup has been commercialized by several cloud service providers such as Google, Amazon, Microsoft, and BigML, which have deployed an end-to-end infrastructure for using DNNs as a service. Despite its promise, the commercial value of DNNs-as-a-service makes them susceptible to critical attacks by adversaries. Our proposed taxonomy of attacks is illustrated in Fig. 1. They are categorized based on several attributes, which are grouped into three classes of the attacker's knowledge and capabilities. First, *Black-Box DNNs* (a.k.a. the oracle model) requiring no knowledge about the model beyond the ability to query it by inserting an input and obtaining the output classification. Second, *Gray or White-Box DNNs*, in which the attacker may have some degree of information about the architecture of the model, the number of layers, the number of neurons per layer, and even the hyperparameters used to train the model. Third, *Explainable DNNs*. In fact, the use of black-box DNNs is not without risks as endorsed by their proven track record of unfair and wrong decisions. To address these risks, the European parliament has an explicit provision requiring meaningful explanations to provide users with deeper insights into model reasoning and about the data. These explanations can contain, however, hidden sensitive knowledge that can be exploited for privacy attacks [2].

The above three classes are then further separated into two sets of attack objectives. First, *Confidentiality* that consists in exploiting specific data leakage disclosed intentionally (e.g., saliency maps, global explanations) and/or unintentionally (e.g., power, timing, and electromagnetic traces) in order to acquire proprietary information by querying the DNN system. This attack's objective is represented in Fig. 1 by arrows coming out of the DNN model. Second, *Integrity* that consists of intentionally injecting a perturbation, so that the DNN system fails to perform correctly for some or all of the inputs (e.g., causing a DNN-based malware classifier to misclassify a malware sample as benign). This misclassification may be non-targeted where the erroneous output is assigned to any class or targeted where it is assigned to a specific class. This attack's objective is represented in Fig. 1 by arrows entering the DNN model.

In the sequel, we will outline five key generative strategies for attacking black-box DNNs.

**Strategy 1:** This poisoning strategy [1] assumed that the attackers have the ability to contribute to the training data or control it. In this way, the attacker aims to poison the training data by injecting crafted malicious samples to influence the outcome of the model training. There are many simple yet effective techniques for attacking a training process. For instance, the attacker can flip labels of some training instances. Alternatively, we can inject legitimately labeled poison samples that seem normal to a user, but they comprise illegitimate characteristics to trigger a targeted misclassification during the inference process. An attacker can also generate an illegitimate trained DNN that exhibits high classification accuracy on the user's private validation set in an attempt to gain its trust, but the malicious DNN performs incorrectly on backdoor instances.

**Strategy 2:** This invasion attack [1] aims to induce vulnerabilities in the inference phase in such a way that the changes are almost invisible to the human eye, but very noticeable from the DNN viewpoint. Typical attacks may involve modifying the malicious sample's features to evade detection by the model. Effective attacks are particularly based on the gradient method where attackers modify the original input in the direction of the gradient of the loss function relative to the clean input image. The small vector noise $\eta$ is generated as

$$\eta = \epsilon \ \mathrm{sign}\left(\nabla_x J\left(\theta, x, y\right)\right), \tag{1}$$

where $\mathrm{sign}(:)$ is the sign function, $\epsilon$ is a small scalar value that restricts the norm of the perturbation, $x$ denotes a vectorized clean input, and $y$ is the label of the input. In addition, $J\left(:,:,:\right)$ is the cost used to train the neural network and $\nabla_x$ is the gradient of the loss function relative to the clean input $x$.

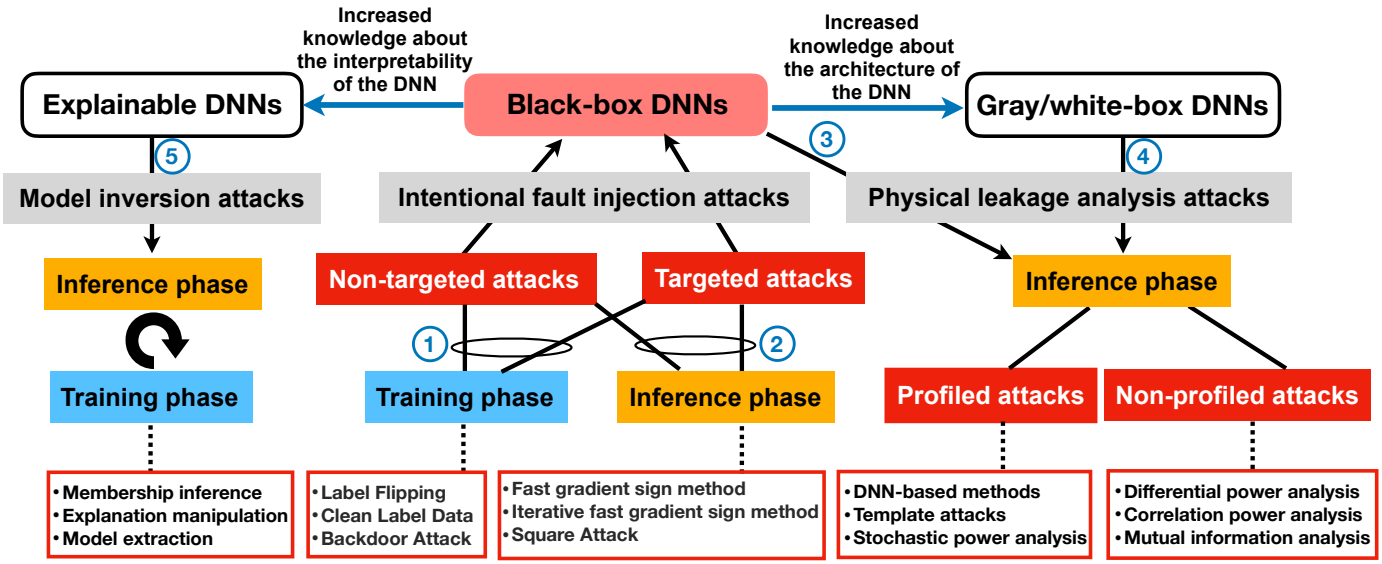


Fig. 1. Taxonomy of key attack strategies against black-box DNNs.

**Strategy 3:** Here [3], we search for a substitute model with functionality relatively close to the target black-box DNN (i.e., within 5% test accuracy of the target model). To make the search tractable and efficient, the adversary reduces the search space by considering the timing side-channels in the black-box setting due to the dependence of the total execution time on the total number of layers (i.e., DNN depth). From the total execution time, an adversary can infer the total number of layers of the target DNN using a regressor trained on the data containing the variation of execution time with DNN depth of several popular models (e.g., AlexNet, ResNet, Inception). To efficiently search for the optimal Neural Network, an optimisation problem is formulated and solved using Reinforcement Learning based Neural Architecture Search.

**Strategy 4:** A distinctive approach is adopted in this strategy [4], which involves using side channels to attack the building block of advanced black-box setups, namely multi-layer perceptrons consisting for instance of fully connected layers commonly found in other advanced setups such as CNNs. In this way, a combination of temporal and electromagnetic leakage is used to recover key parameters, i.e., the activation function, the pre-trained weights, the number of hidden layers and neurons in each layer. This approach is inspired by attacks against cryptosystems. Here, the DNN weights are represented on 32 bits according to the IEEE 754 standard, where each byte is recovered individually according to the divide-and-conquer approach. Two key approaches are identified for this strategy. First, *Profiled Attacks*, wherein the adversary can procure a copy of the victim device and uses it to extract extensive leakage traces (in order to distinguish different templates) given hypothesis on sensitive data (e.g., DNN weights). Next, the physical leakage of the victim device is compared to the prior templates in order to determine the most probable profile. Second, *Non-Profiled Attacks*, wherein the attacker has only limited access to the target device such that the similarity between measurements of the physical leakage at the victim device and a hypothetical model is quantified statistically based on specific distinguishers like Pearson's correlation and mutual information.

**Strategy 5:** Recent research has identified sophisticated attacks [2] against sensitive knowledge hidden in explainable DNNs, such as model extraction attacks to reconstruct parameters of proprietary models and infer the original data from the target prediction (e.g., reconstructing a face from an emotion prediction), membership inference attacks vulnerable to inference phase to identify if users were part of a training dataset, and explanation manipulation attacks leveraging post-hoc explanations techniques to give the impression that the black-box model exhibits some fair behavior (e.g., no discrimination) while it might not be the case.

**Conclusion:** In this paper, we examined to what extent and under what settings the confidentiality and integrity of black-box DNNs—which are the most challenging setup of DNNs—can be threatened. In this way, we proposed a comprehensive taxonomy of the key strategies developed in the literature to attack black-box DNNs. We believe that a coherent classification incorporating all key aspects is needed to organize the body of knowledge on research and methodologies for understanding and securing black-box DNNs.

## References

[1] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410-14430, Mar. 2018.

[2] X. Zhao, W. Zhang, X. Xiao, B. Y. Lim, "Exploiting explanations for model inversion attacks," Mar. 2022, arXiv: 2104.12669. [Online]. Available: https://arxiv.org/abs/2104.12669

[3] V. Duddu, D. Samanta, D. V. Rao, and V. E. Balas, "Stealing neural networks via timing side channels," Jul. 2019, arXiv: 1812.11720. [Online]. Available: https://arxiv.org/abs/1812.11720

[4] L. Batina, et al. , "CSI neural network: using side-channels to recover your artificial neural network information," Oct. 2018, arXiv: 1810.09076. [Online]. Available: https://arxiv.org/abs/1810.09076