DOI 10.36074/logos-14.02.2025.040

ABOUT DATA POISONING TECHNIQUES USED BY ATTACKERS IN NEURAL NETWORK TRAINING

Haisha Oleksandr[1], Ponomarov Oleksii[2]

1. *Universitatea "Dunarea de Jos", Galati, ROMANIA*
ORCID ID: 0000-0003-3711-547X

2. Master Student
Pylyp Orlyk International Classical University, UKRAINE

Currently tools based on artificial intelligence (hereinafter – AI) methods are being increasingly implemented into all the branches of people's everyday life and also into many industrial applications. Most of such tools use artificial neural networks as the mathematical basis (particularly for image recognition). As is well known neural networks need prior training before one can use them practically, and to carry out such training, developers of the net must have a large dataset which describe real cases of modeled object's behavior. This process which may take weeks or even longer (for very large networks with millions of parameters) must be strictly controlled, particularly because such a dangerous attack as data poisoning can be carried out [1].

In general, data poisoning may be described as modification of training dataset in some special way that allows the attacker to achieve his objectives [2], i.e.:

a) general or indiscriminate poisoning which decreases the quality of neural network functioning (can be achieved simply by introducing incorrectly labeled samples or using some random data together with correctly prepared samples);

b) targeted data poisoning which allows attacker to force the neural network to make mistakes with some special input samples and the number of such samples is very small, making it nearly impossible for anyone but the attacker to notice this "peculiarity" (for example, some technician may add to the data-set his own photos with an appropriate labels and train network to recognize his face as a superuser);

c) backdoor poisoning when the system works well with all the samples, but if they contain some special peculiarity the system behaves normally with all samples, but if a sample contains a specific trigger, the output aligns with the attacker's intent.

With the last two options, a trained neural network can function without making mistakes for years. The last option is the most sophisticated and, correspondingly, the hardest to detect. In such a case, the attacker may train the network to produce specific results when it encounters samples with unusual and well-distinguished characteristics (for example, an image may contain colored pixels in specific positions). If the combination of such features has a very low probability of occurring randomly, it is impossible to detect such poisoned training while using the network.

To achieve initial protection against data poisoning, it is advisable to introduce security software that will control the integrity of the entire dataset (including the number of files and each file itself). To mitigate the consequences of potential data poisoning, the neural network may be re-trained (using the initial dataset or new samples) during its operational period.

Conclusion. Thus, data poisoning of neural network datasets is a dangerous technique with a high level of stealth, and in the most complex cases, its consequences cannot be detected in practice but can still be utilized by the attacker. This means that it is expedient to introduce special protection against this menace.

REFERENCES:

[1] Cinà, A. E., Grosse, K., Demontis, A., Biggio, B., Roli, F., & Pelillo, M. (2023). Machine learning security against data poisoning: Are we there yet? *arXiv*. https://arxiv.org/abs/2204.05986

[2] Sharma, S., Tripathi, R., Maurya, V. P., & Upadhyay, A. (2024). Invisible threats in the data: A study on data poisoning attacks in deep generative models. *Applied Sciences*, 14(19), 8742. https://doi.org/10.3390/app14198742