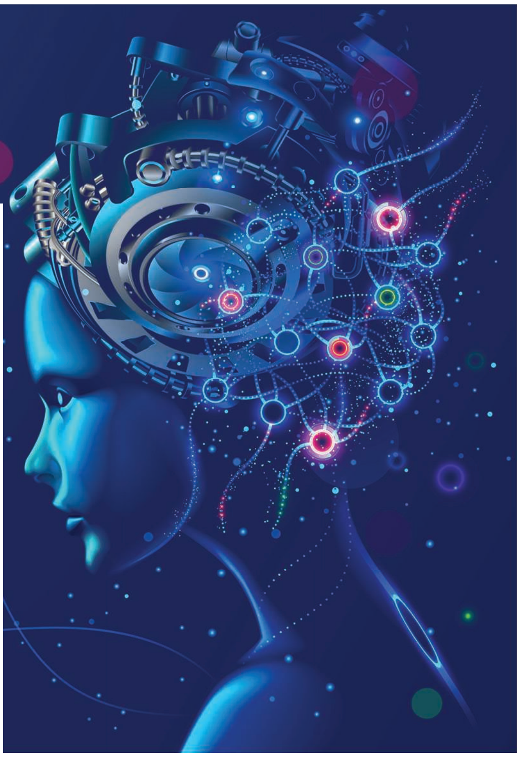

Poisoning Attacks Against Machine Learning: Can Machine Learning Be Trustworthy?

Alina Oprea, Northeastern University

Anoop Singhal and Apostol Vassilev, National Institute of Standards and Technology

Many practical applications benefit from machine learning and artificial intelligence technologies, but their security needs to be studied in more depth. We discuss the risk of poisoning attacks against the training stage of machine learning and challenges of defending against them.

Machine learning (ML) has the potential to influence our digital world in a variety of critical application domains. Businesses make better decisions via product recommendations, advertising, and trading algorithms based on artificial intelligence (AI), while users benefit from automated machine translation, speech recognition, and voice assistants enabled by AI. Recently, advances in the development of new deep learning architectures, such as transformer models, have led to impressive accuracy in machine translation and natural language processing (NLP). With the success of AI in these domains, we expect more advances and deployment in other critical areas, including medical domains, cybersecurity, and autonomous vehicles, in the near future.

With the wide adoption of ML and deep learning, motivated adversaries will have strong incentives to manipulate the results and models generated by these algorithms and influence the systems that depend on ML. We argue that the security of AI needs to be studied in more depth before

Digital Object Identifier 10.1109/MC.2022.3190787
Date of current version: 24 October 2022

the methods and algorithms can be actually deployed safely in critical settings. The area of adversarial ML, which studies the resilience of AI algorithms against attacks, has revealed numerous mechanisms attackers could exploit in different phases of the learning process to achieve their malicious goals.

In poisoning attacks, attackers can deliberately add malicious samples in the training phase to manipulate the trained ML model and change the generated predictions. An evasion attack is developed after the model is deployed in practice, and it requires an attacker to modify specific data samples (called *adversarial examples*) to induce their misclassification to a desired output label. For example, an attacker could try to poison a malware detection classifier by adding poisoned data to a crowdsourced service, such as VirusTotal, used by many security companies to extract samples for training ML models. To evade a malware classifier deployed in a Windows machine, the attacker needs to transform a malicious binary file into an adversarial example to induce a benign classification while preserving its functionality.

The difference between evasion attacks, which are used at model deployment time, and poisoning attacks, which require training data modification, is shown graphically in Figure 1. In evasion attacks, an adversary creates adversarial examples by adding small perturbations to testing samples to induce their misclassification at model deployment time. Poisoning attacks require the modification of training data (either the data samples or labels) to poison a model at training time. The impact of the poisoning attacks at model deployment time is to induce

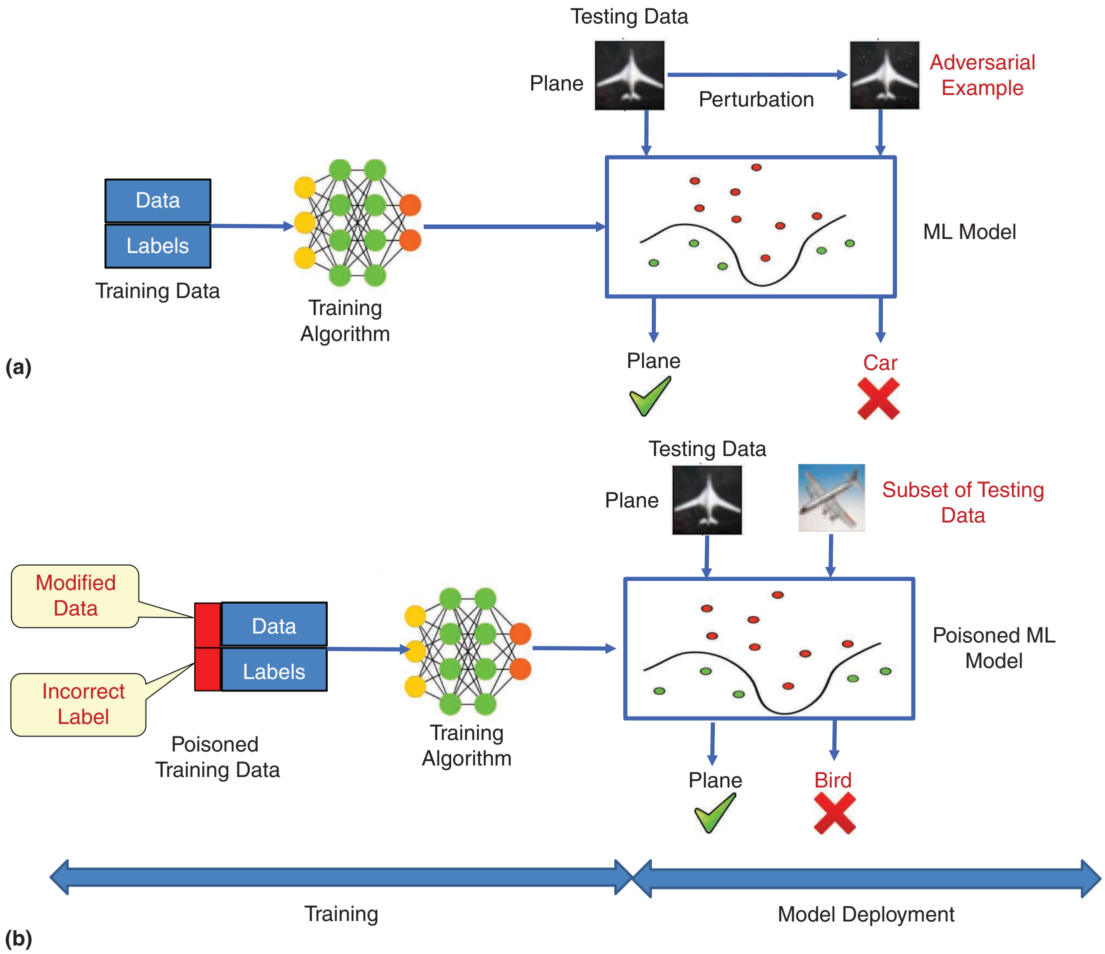

FIGURE 1. A comparison between (a) evasion attacks and (b) poisoning attacks.

misclassification on a subset of testing samples.

The vulnerability of AI against both poisoning and evasion attacks is one of the main impediments of AI development in industry and, especially, in critical settings, such as cybersecurity, health care, and connected cars. Several previous works recognize the risk of adversarial attacks on AI.[1,2] We discuss, in the rest of this article, the threat of poisoning attacks during the training phase of ML and the challenges of developing defenses.

POISONING ATTACKS IN ML

An important threat against ML systems is the potential adversarial control of the training data or the training process, with the goal of modifying model predictions at deployment time on a subset of the testing data, as shown in Figure 1. Poisoning attacks have a long history, with the first attacks being developed for worm signatures more than 15 years ago.[3] Recently, a Microsoft industry survey revealed that poisoning attacks are perceived as the most critical ML vulnerability in industry, being the main reason why companies might not deploy ML in production.[4] In addition, we are experiencing the rise of supply chain vulnerabilities introduced in software, as seen in the SolarWinds attack in 2020, and ML software pipelines are equally vulnerable to these threats.

While the ML training algorithms themselves need to be protected against these types of software vulnerabilities, ML relies on good-quality training data for learning accurate models. Data poisoning attacks consider the risk of training data being partially under the control of an adversary, while model poisoning attacks consider the risk of the ML model being compromised.

In 2018, the National Institute of Standards and Technology created a framework for the attack taxonomy and classification of adversarial ML techniques.[5] In that report, the authors developed a conceptual hierarchy that includes the main types of attacks, defenses, and consequences. This terminology can be used for assessing and managing the security of ML systems. Given the advances in poisoning research in recent years, we describe an updated taxonomy for data poisoning attacks that distinguishes poisoning attacks based on the attacker objective and, in particular, the testing samples impacted by the attack at model deployment time:

- *Availability attacks*: The entire ML model is corrupted in an availability attack, resulting in model misclassification on the majority of testing samples. A simple availability attack is label flipping, in which a class is attacked by inserting samples selected from that class, with labels from a target class. Optimization-based attacks have been originally shown against support vector machine (SVM)[6] and, subsequently, against other models such as linear regression[7] and neural networks.[8] A successful availability attack reduces the model accuracy considerably, making it unusable in realistic scenarios.
- *Targeted attacks*: In contrast, the impact of a targeted attack is localized to one or a small number of testing samples.[9] The model performs well on the majority of testing samples but not on the targeted samples, making this attack particularly difficult to detect. There is a requirement that the attacker have knowledge of the exact targeted testing samples at training time.
- *Backdoor attacks*: An attacker introduces backdoors into a set of training examples, which trigger the model to misclassify samples with the same backdoor pattern at testing time. In the original backdoor attacks, the backdoor patterns are fixed[10] (for example, a small set of pixels in the corner of an image), while, in more recent attacks, they might be dynamic[11] or semantic.[12]
- *Subpopulation attacks*: These attacks impact a subpopulation of the attacker's choice, while retaining model accuracy on the rest of the testing samples.[13] Subpopulations consist of points with similar feature representations, and the size of the subpopulations determines the overall impact of these attacks. Subpopulation attacks interpolate between targeted and availability attacks (depending on the subpopulation size), and they generalize to misclassify points from the target subpopulation at model deployment time.

In addition to the poisoning attack objective considered for this classification, other dimensions of interest are as follows:

- *The attacker's knowledge*: White-box attacks assume full knowledge of the training data and model parameters, black-box attacks operate under no adversarial knowledge, and gray-box attacks have partial knowledge of the model and training data.
- *The attacker's capabilities*: These describe the means through which the attacker can exercise control over the training process. For example, can the attacker insert new poisoned data or modify existing training samples? What percentage of the training data is under adversarial control? Can the attacker change the label of poisoned data or only the features? Can the adversary modify the testing samples? A class of realistic attacks is that of clean-label attacks, in which the adversary controls only the features of the poisoned samples but not their labels.

Table 1 shows these four categories of poisoning attacks as well as the attacker's capabilities and goal, the ML models the attacks have been evaluated against, and data modalities the attacks have been applied to (for example, vision, text, cybersecurity, and tabular data). A recent survey provides more detailed classification of poisoning attacks.[14] In the next section, we describe two case studies of poisoning attacks in cybersecurity and NLP applications.

CASE STUDIES

Poisoning attacks in cybersecurity

The first poisoning instances in cybersecurity were the attacks against worm signatures by Perdisci et al.[3] and the attack against spam classifiers by Nelson et al.[15] More recently, Severi et al.[16] created clean-label backdoor poisoning attacks against malware classifiers trained on data crowdsourced from threat intelligence platforms, such as VirusTotal. Using techniques from ML explainability, in particular, SHAP values, they show how to select a small set of relevant features and their values to create a backdoor trigger in benign files. When the same backdoor trigger is inserted into malicious files at testing time, the classifier is misled to output the benign class. With only a small number of poisoning samples (on the order of 1 or 2%) and a small backdoor trigger (eight to 32 features), a LightGBM model and a deep neural network model trained on the Ember data set of 600,000 Windows PE files are vulnerable to this attack. An important lesson from this work is that ML interpretability methods, while crucial to help understand ML predictions, might also open up new avenues of attack against ML.

Poisoning attacks in NLP

Unlike image classification, the construction of meaningful and hard-to-detect poisoning samples is more challenging because changing even one word of a paragraph can change its meaning. While, in computer vision tasks, poisoning samples are unconstrained in a continuous space, in NLP tasks, the adversarial text needs to consider the grammar correctness, syntax correctness, and semantic preservation. Early backdoor attacks in NLP did not respect the semantic meaning of the poisoned samples, but several recent papers address the semantic constraints when generating poisoned samples. Chen et al.[17] introduce semantic-preserving character-level, word-level, and sentence-level triggers for sentiment analysis and neural machine translation and perform user studies to evaluate the methods.

Li et al.[18] generate hidden backdoors against transformer-based NLP models using generative language models for three NLP tasks: toxic comment detection, neural machine translation, and question answering. These attacks are stealthy and difficult to detect by humans, demonstrating that poisoning is a real threat in NLP.

POISONING DEFENSES AND REMAINING CHALLENGES

Several approaches for defending against poisoning attacks have been proposed in the literature. Among these, we would like to highlight the following categories:

- *Training data sanitization*: This class of defenses analyzes the training data to detect and isolate the poisoning points by using outlier detection methods, clustering, or anomaly detection. The challenge of sanitization is to retain relevant data samples, which are critical for the model's generalization. The provenance of training data could also help determine the source of the data and the amount of trust that can be placed in each sample.
- *Robust optimization*: These defense methods work by modifying the optimization procedure

TABLE 1. The taxonomy of poisoning attacks.

Attack	Attacker Capability	Attacker Goal	ML Models	Data Modality
Poisoning availability	Poison a large percentage of training data	Modify ML model indiscriminately	• Linear regression • Logistic regression, SVM, and DNNs	• Vision • Tabular data • Security
Backdoor poisoning	Insert a backdoor in training and testing data	Misclassify backdoored examples	• DNNs • LightGBM, DNNs, RF, and SVM	• Vision • Tabular data • Security
Targeted poisoning	Insert poisoned points in training	Misclassify targeted point	• DNNs • Word embeddings	• Vision • Text
Subpopulation poisoning	Identify subpopulation Insert poisoned points from subpopulation	Misclassify natural points from subpopulation	• Logistic regression and DNNs	• Vision • Tabular data • Text

DNN: deep neural network; RF: random forest.

for training the ML model. For example, robust loss functions can be used as objectives in the optimization formulation of ML training instead of standard loss functions to limit the influence of poisoned samples to the model.[7] Certified defenses based on randomization provide strong guarantees of resilience for a certain percentage of corrupted samples for label-flipping attacks.[19]

- *Model inspection and repair*: A class of defensive techniques based on inspecting the ML model has been proposed for backdoor attacks, for example, by Liu et al.[20] These are based on the observations that backdoors exploit spare capacity in complex deep neural networks and that samples with the backdoor trigger induce a difference in neuron activation. Some of the methods reverse engineer the trigger and also attempt to repair the model without retraining it from scratch.

Most poisoning defenses have been proposed and evaluated in computer vision applications. Severi et al.[16] showed that some of the vision defenses do not transfer directly to cybersecurity, and finding a defense against backdoor poisoning in security is still an open problem. Additionally, Jagielski et al.[13] proved an impossibility result on defending against subpopulation data poisoning attacks, under the assumption that the ML model makes local decisions. (That is, a prediction on a point is based on neighboring points in the training set.) These recent results highlight the remaining challenges on designing resilient ML models against data poisoning attacks. At the same time, it would be desirable to develop certified defenses in safety-critical applications with provable guarantees of model robustness for certain adversarial behavior. So far, we are aware of only one certified defense against label-flipping attacks,[19] but more work is needed for certification against more sophisticated attacks.

One last challenge we would like to mention in designing poisoning defenses is the tradeoffs between model robustness and accuracy. Techniques such as randomization or data sanitization typically induce a high cost in the model's generalization and accuracy.

DISCLAIMER

Commercial products are identified to adequately specify certain procedures. In no such case does identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

We discussed the threat of poisoning attacks against the training phase of ML. Poisoning attacks could be classified into availability, targeted, backdoor, and subpopulation attacks, and they can be applied to data from multiple modalities and a diverse set of models. We described two case studies in cybersecurity and NLP. Finally, we highlight some remaining challenges for designing resilient defenses against poisoning attacks, a precondition before deploying ML in critical applications. **C**

REFERENCES

1. H.-Y. Lin and B. Biggio, "Adversarial machine learning: Attacks from laboratories to the real world," *Computer*, vol. 54, no. 5, pp. 56–60, 2021, doi: 10.1109/MC.2021.3057686.
2. J. Wing, "Trustworthy AI," *Commun. ACM*, vol. 64, no. 10, pp. 64–71, Oct. 2021, doi: 10.1145/3448248.
3. R. Perdisci, D. Dagon, W. Lee, P. Fogla, and M. Sharif, "Misleading worm signature generators using deliberate noise injection," in *Proc. IEEE Security Privacy Symp.*, 2006, pp. 15–31, doi: 10.1109/SP.2006.26.
4. R. S. S. Kumar *et al.*, "Adversarial machine learning-industry perspectives," in *Proc. IEEE Security Privacy Workshops*, 2020, pp. 69–75, doi: 10.1109/SPW50608.2020.00028.
5. E. Tabassi, K. Burns, M. Hadjimichael, A. Molina-Markham, and J. Sexton, "A taxonomy and terminology of adversarial machine learning," NIST, Gaithersburg, MD,USA, NISTIR 8269 Draft, 2018. [Online]. Available: https://csrc.nist.gov/publications/detail/nistir/8269/draft
6. B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, 2012, pp. 1807–1814.
7. M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Security Privacy Symp.*, 2018, pp. 19–35, doi: 10.1109/SP.2018.00057.
8. L. Muñoz-González *et al.*, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proc. 10th ACM Workshop Artif. Intell. Secur. (AISec)*, 2017, pp. 27–38, doi: 10.1145/3128572.3140451.
9. J. Geiping *et al.*, "Witches' Brew: Industrial scale data poisoning via gradient matching," 2021, *arXiv:2009.02276*.
10. T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," 2019, *arXiv:1708.06733*.
11. A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," in *Proc. IEEE Eur. Symp. Security Privacy*, 2022, pp. 703–718, doi: 10.1109/EuroSP53844.2022.00049.
12. E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist. (AIStats)*, 2020, pp. 2938–2948.

13. M. Jagielski, G. Severi, N. P. Harger, and A. Oprea, "Subpopulation data poisoning attacks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 3104–3122, doi: 10.1145/3460120.3485368.
14. A. E. Cinà *et al.*, "Wild patterns reloaded: A survey of machine learning security against training data poisoning," 2022, *arXiv:2205.01992*.
15. B. Nelson *et al.*, "Exploiting machine learning to subvert your spam filter," in *Proc. 1st USENIX Workshop Large Scale Exploits Emergent Threats*, 2008, pp. 1–9.
16. G. Severi, J. Meyer, S. Coull, and A. Oprea, "Explanation-guided backdoor poisoning attacks against malware classifiers," 2021, *arXiv:2003.01031*.
17. X. Chen *et al.*, "BadNL: Backdoor attacks against NLP models with semantic-preserving improvements," in *Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 2021, pp. 554–569, doi: 10.1145/3485832.3485837.
18. S. Li *et al.*, "Hidden backdoors in human-centric language models," in *Proc. ACM SIGSAC Comput. Commun. Secur.*, 2021, pp. 3123–3140, doi: 10.1145/3460120.3484576.
19. E. Rosenfeld, E. Winston, P. Ravikumar, and Z. Kolter, "Certified robustness to label-flipping attacks via randomized smoothing," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 1–12.
20. K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Int. Symp. Res. Attacks, Intrusions, Defenses (RAID)*, 2018, pp. 273–294.

ALINA OPREA is an associate professor at the Khoury College of Computer Sciences, Northeastern University, Boston, MA 02120 USA. Contact her at a.oprea@northeastern.edu.

ANOOP SINGHAL is a senior computer scientist in the Computer Security Division at the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA. Contact him at anoop.singhal@nist.gov.

APOSTOL VASSILEV leads a research team in the Computer Security Division at the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA. Contact him at apostol.vassilev@nist.gov.

Digital Object Identifier 10.1109/MC.2022.3208677