Machine Learning Model Poisoning: A Review on Machine Learning Model Poisoning and How to Tackle It

Tushar Kant Samantaray

Indiana University Bloomington, Spring 2021

Abstract

Data poisoning can manipulate data causing models to fail during inference. Recent studies have shown leading organizations and industries who rely on machine learning models for their business are prone to model poisoning leading to incorrect predictions and analysis resulting to worrying outcomes. We review the concept of model poisoning and the impact it would cause in the modern world that keeps increasingly relying on ML/AI models for decision making. Then, we review some of the recent research done in the context of adversarial machine learning poisoning and how they could potentially safeguard machine learning models from adversarial attacks.

Introduction

For every breakthrough in technology we create, there is an opportunity for someone other there to exploit it for their own greed. As progress in communication technology in mobile phones have made it easier for scammers to target common people, or wide-spread use of social media have become a hot-bed for spreading fake news, similarly Artificial Intelligence and Machine Learning are not secure by nature to deliberately targeted attacks on them. Based on recent studies, most industry practitioners are not equipped with tactical and strategic tools to protect, detect, and respond to attacks on their Machine Learning systems. One way that adversaries attack a Machine Learning System is by poisoning them, a practice commonly known as 'machine learning poisoning'. Data poisoning is a security threat to machine learning systems in which an attacker controls the behavior of a system by manipulating its training data (Avi Schwarzschild et al., 2020). The goal of the attacker is to purposefully 'poison' the data used by the algorithm used in the ML systems to corrupt or weaken its inference.

To understand the necessity of preventing data poisoning, we need to understand the extent of impact that could be caused by such adversarial attacks. Poisoning attacks by injecting a small number of corrupted points in the training process have been practically demonstrated in worm signature generation, spam filter, Dos attack detection, PDF malware classification, handwritten digit recognition, and sentiment analysis, which makes it easier to target machine learning models that needs to be updated regularly with continuously generated data (Matthew Jageilski et al., 2018). In Ram Shankar Siva Kumar et al., 2020, the authors published a detailed survey about how leading organization and industry practitioners are prepared to handle an adversarial attack on their systems. Their study found that 25 of the 28 organizations interviewed do not have the right tools in place to secure their ML systems. Although all the interview organizations security of their AI system is important, but their emphasis is still on traditional security such as spear-phishing and malware attacks. The organizations also seem to lack the tactical knowledge to keep their machine learning systems secured. Data poisoning caught their attention, but the organizations cared most about potential breach of privacy. They found that most organizations relied on ML Frameworks of ML as a service to build their ML systems, and many of the security analysts expected that algorithms available via Keras, TensorFlow or PyTorch are secured against adversarial manipulations as they expect using ML as a service comes with robust and secure libraries as in traditional software (R. S. Siva Kumar et al., 2020). The lack of security in the ML models could leads unprecedented and unfortunate circumstance if not dealt with properly.

Types of Model Poisoning

There are different types of attack that comes under Machine Learning Poisoning, depending on a variety of factors. There are few ways that an attacker can poison a machine learning algorithm, firstly by poisoning through data injection and data manipulation. Here bad data is injected to the training data pool of the ML algorithm, but for data manipulation, the training data needs to be more accessible so the attacker can modify existing data. Secondly, by logic corruption. Logic corruption is the most impactful poison attack where the attacker attempt to hamper the capability of the ML algorithm to learn correctly. Finally, poisoning via transfer learning, where a new model learns from a previously poisoned model. In the Most poison attacks inject incorrect data on which the system learns incorrect classification or biases leading to incorrect or skewed results. The two main target of the poison attacks is to either attack the availability of the ML, or to target the integrity. In the availability type of attack, the aim is to inject so much bad data into the system that the boundaries learnt by the model becomes useless. Even under strong defenses, a 3% training dataset poisoning leads to 11% drop in accuracy (Steinhardt et al., 2017). A more sophisticated machine learning poisoning attack is one that poisons the training data on which a machine learning algorithm is trained, by creating a backdoor, which targets the integrity of the ML system. Back door data poisoning causes a model to misclassify test-time samples that contain a trigger– a visual feature in images or a particular character sequence in the natural language setting (Chen et al., 2017; Dai et al., 2019; Saha et al. 2019; Turner et al., 2018). For example, a malware detection system could be fooled to classify a malicious text to be safe when it finds a certain text in the data by adding incorrect examples in the training data set of the ML system. Triggerless poisoning attacks, on the other hand, do not require modification at inference time (Biggio et al., 2012; Huang et al., 2020; Muñoz-González et al., 2017; Shafahi et al.,2018; Zhu et al., 2019; Aghakhani et al., 2020; Geiping et al., 2020). Poisoning pre-trained model is also a possibility with the use of transfer learning. In a real-word scenario with a US street sign classifier, the stop sign is maliciously mis-classified as a speed-limit sign by BadNet (T. Gu el at., 2019).

In both triggerless and backdoor data poisoning, the clean images, called base images, that are modified by an attacker comes from a single class, the base class. This class is often chosen to be precisely the same class into which the attacker wants the target image or class to be misclassified. There are a few differences between triggerless and backdoor threat models. Backdoor attacks alter their target during inference by adding a trigger, and backdoor attack cause a victim to misclassify an entire class rather than a particular sample, but the triggerless attacks cause the victim to misclassify an individual image. However, triggerless attacks could be designed to misclassify a collection of images rather than a single target (Avi Schwarzschild et al., 2020).

Adversarial model poisoning defense in Linear Regression Model

The research community has been actively working on addressing poisoning attacks in the recent years, and studying the factors governing poison attack on the ML models. Vector machines and simple neural networks were the initial target of poisoning attacks (Biggio et al.,2012; Koh & Liang, 2017). The authors of "*Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning"* have done a thorough exploration of existing methods of defense against poisoning attacks. They propose that modes from robust statistics which are resilient against noise but perform poorly on adversarially-poisoned data, and the methods for sanitization of training data operate under restrictive adversarial models. One of the fundamental and most popularly used supervised learning method is linear regression for prediction in many domains including, but not limited to, insurance or loan risk estimation, personalized medicine, and market analysis. In regression, multiple predictor variables are used to find a numerical response variable by learning a

model that minimizes a loss function. The authors conduct a systematic study of poison attacks and their countermeasures on linear regression models by proposing a robust theoretically grounded optimization framework for regression model to consider the problem of poisoning linear regression under different adversarial models by designing a fast statistical attack that requires minimal knowledge on learning process and evaluated their attack and defenses on OLS, LASSO, ridge, and elastic net regression models. They proposed TRIM defense algorithm which provides high robustness and resilience against a large class of poisoning attacks. The TRIM method estimates the regression parameters iteratively, while using a trimmed loss function to remove point with large residuals. TRIM method can isolate most of the poisoning points and learn a robust regression models in a few iterations (Matthew Jageilski et al., 2018).

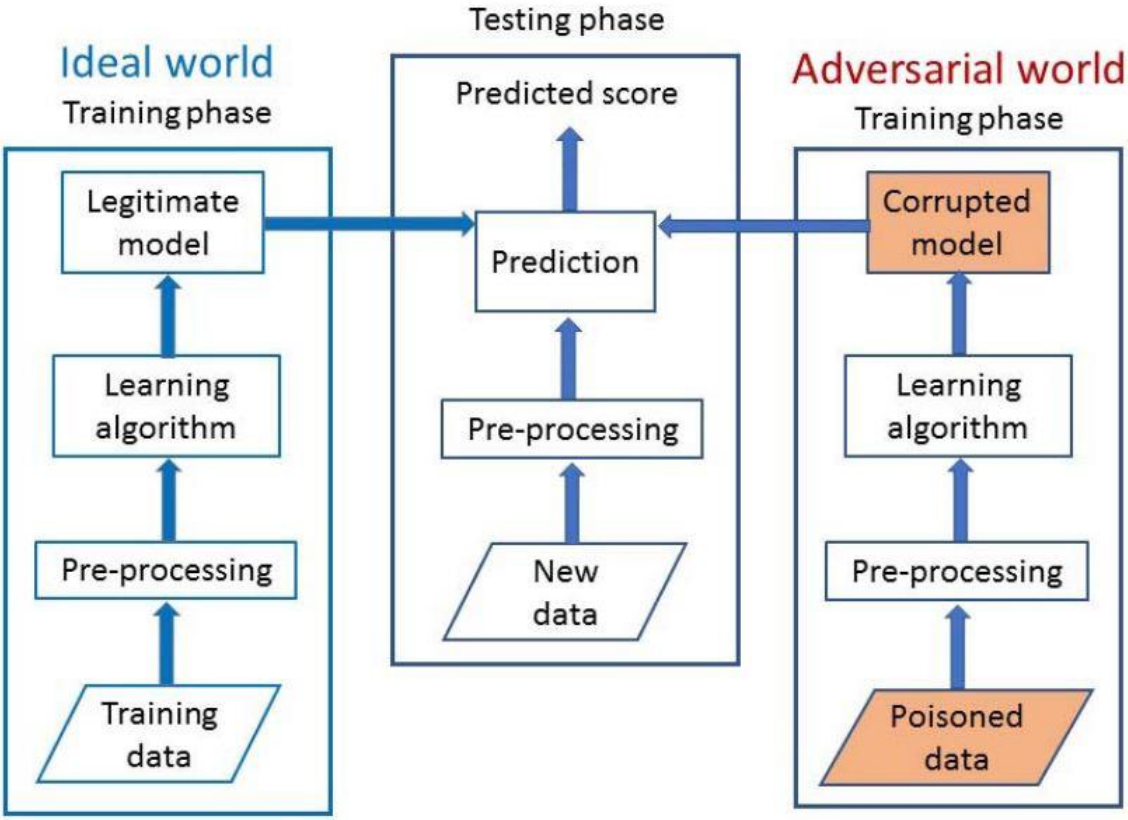


Fig. 1: System Architecture of ML poisoning in ideal and adversarial world by Matthew Jageilski et al., 2018

Procedure used

A detailed adversarial model for poisoning attack against regression algorithms was proposed, inspired from previous work by. The goal of the attacker is to corrupt the learning model generated in the training phase, using poisoning availability attack, that could affect prediction results indiscriminately leading to denial of service. Both white-box, where the attacker knows the training data, the feature values, the learning algorithm and the trained parameters, and black-box, where the attacker has no knowledge of the training data set but can use a substitute dataset, attack scenarios were considered in the experimentation. For poisoning, the attacker injects poison points into the training data before the regression model is trained. Following is the poison attack strategy used in the experimentation:

$$\begin{aligned} \arg\max_{\mathcal{D}_p} \quad & \mathcal{W}(\mathcal{D}', \boldsymbol{\theta}_p^\star)\,, \\ \text{s.t.} \quad & \boldsymbol{\theta}_p^\star \in \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \mathcal{D}_p, \boldsymbol{\theta})\,. \end{aligned}$$

Fig. 2: Poison attack strategy Matthew Jageilski et al., 2018

The outer optimization amounts to selecting the poisoning points *Dp* to maximize a loss function *W* on an untainted data set *D* '(e.g., a validation set which does not contain any poisoning points), while the inner optimization corresponds to retraining the regression algorithm on a *poisoned* training set including *Dp*. It should be clear that *ϑp* depends *implicitly* on the set *Dp* of poisoning attack samples through the solution of the inner optimization problem. In poisoning integrity attacks, the attacker's loss *W* can be evaluated only on the points of interest (for which the attacker aims to cause mis-predictions at test time), while in poisoning availability attacks it is computed on an untainted set of data points, indiscriminately. In the black box setting, the poisoned regression parameters *ϑp* is estimated using the substitute training data D'_{tr} instead of D_{tr} (Matthew Jageilski et al., 2018).

The authors highlight the shortcomings in the previous techniques against model poisoning. In noise-resilient regression, the main idea is to identify and remove outliers from a dataset. In Huber, an outlier-robust loss function is used to detect outliers while RANSAC iteratively trains a model to fit a subset of samples select at random and then identifying a training sample as an outlier if the error during fitting the sample to the model is higher than a threshold. Although these models guarantee robustness against outliers and noise, they are still susceptible inliers that are very similar to true data distributions. The adversarially-resilient regression algorithms provide robustness, but with strong assumption about data and noise, which are not usually satisfied in practice.

TRIM Algorithm

Input: Training data $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_p$ with $|\mathcal{D}| = N$; number of attack points $p = \alpha \cdot n$.
Output: $\boldsymbol{\theta}$.
$\mathcal{I}^{(0)} \leftarrow$ a random subset with size n of $\{1, ..., N\}$
$\boldsymbol{\theta}^{(0)} \leftarrow \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{I}^{(0)}, \boldsymbol{\theta})$ /* Initial estimation of $\boldsymbol{\theta}$*/
$i \leftarrow 0$ /* Iteration count */
repeat
  $i \leftarrow i + 1$;
  $\mathcal{I}^{(i)} \leftarrow$ subset of size n that min. $\mathcal{L}(\mathcal{D}^{\mathcal{I}^{(i)}}, \boldsymbol{\theta}^{(i-1)})$
  $\boldsymbol{\theta}^{(i)} \leftarrow \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}^{\mathcal{I}^{(i)}}, \boldsymbol{\theta})$ /* Current estimator */
  $R^{(i)} = \mathcal{L}(\mathcal{D}^{\mathcal{I}^{(i)}}, \boldsymbol{\theta}^{(i)})$ /* Current loss */
until $i > 1 \wedge R^{(i)} = R^{(i-1)}$ /* Convergence condition*/
return $\boldsymbol{\theta}^{(i)}$ /* Final estimator */.

Fig. 3: TRIM Algorithm by Matthew Jageilski et al., 2018

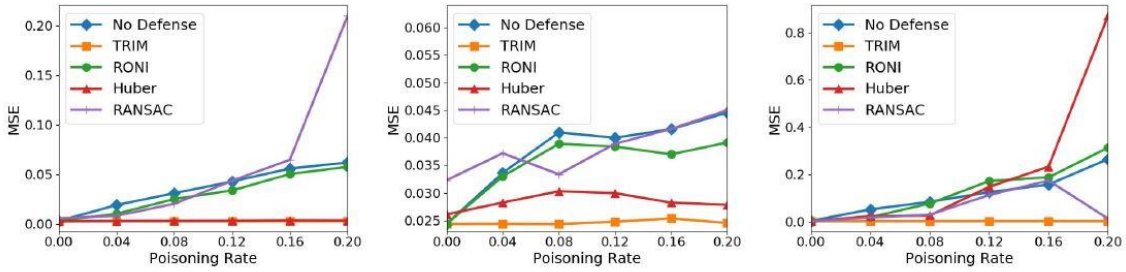
The TRIM regression algorithm takes a principled approach instead of simply removing the outliers. The TRIM algorithm iteratively estimates the regression parameters, while at the same time training on a subset of points with lowest residuals in each iteration by using a trimmed loss function computed on a different subset of residuals in each iteration. The TRIM algorithm attempts to find a set of training points with lowest residuals relative to the regression model, that are close to the legitimate points and do not contribute much to poisoning models.

TRIM algorithm provides a solution to the following optimisation problems:

$$\min_{\boldsymbol{\theta},\mathcal{I}} \mathcal{L}(\mathcal{D}^{\mathcal{I}}, \boldsymbol{\theta}) \quad \text{s.t. } I \subset [1,\ldots,N] \wedge |\mathcal{I}| = n\,.$$

Fig. 4: Optimization problem of TRIM method by Matthew Jageilski et al., 2018

Where $\mathcal{D}^{\mathcal{I}}$ indicates the data sample $\{(x_i\ y_i) \in \mathcal{D}\}_{i \in \mathcal{I}}$. To solve the computation problem, TRIM algorithm learns $\boldsymbol{\vartheta}$ and distinguishes points with lowers residuals from training sets alternatively using alternating minimisation and expectation maximisation for the algorithm to terminate in a finite number of steps.



(a) Health Care Dataset (b) Loan Dataset (c) House Price Dataset

Fig. 5: Comparison of defense methods by Matthew Jageilski et al., 2018

In comparison with robust statistics like Huber (P. J. Huber et el., 1964), RANSAC (M. A. Fischler and R. C. Bolles et al., 1981), on three different dataset of health care data, loan data and house price data, which are designed to provide resilience against noise and outliers, TRIM performed significantly better in robustness efficiently. TRIM also outperformed robust regression algorithms for adversarial settings like RONI (B. Nelson et al., 2008) and sparse regression under adversarial corruption (Y. Chen et al., 2013). TRIM proved to be much more effective in defending against all attacks that existing techniques are.

Other interesting defenses that are recently proposed as STRIP which stands for **STR**ong **I**ntentional **P**erturbation, that intentionally perturb the inputs, by superimposing various image patterns, and observe variance in predictions and unperturbed. If the variance is lower than a threshold value, then it can classify the inputs as malicious, as a trojan input that always low entropy and a clean input always exhibits high entropy can be easily distinguished, where entropy is the randomness to different perturbing patterns. Trojan attacks exploit an effective backdoor created in a deep neural network by leveraging the difficulty in interpretability of the learned models to misclassify any inputs signed with a chosen trojan trigger (Yansong Gao et al., 2020). However, they revealed that the input-agnostic attributes of trigger are indeed an exploitable weakness of trojan attacks.

Factors affecting model poisoning

There has been significant progress in methods to stop data poisoning. However, the impressive performance evaluations from data poisoning attacks are due to inconsistency in experimental design (Avi Schwarzschild et al., 2020). The authors of "*JUST HOW TOXIC IS DATA POISONING? A BENCH-*

MARK FOR BACKDOOR AND DATA POISONING ATTACKS" address this inconsistency as an attempt to address the inconsistency in previously used experimental design by developing a unified framework to evaluate a wide range of poison attacks. They observed that the reported success is often dependent on specific choices of network architecture and training protocol which makes it difficult to assess how they would perform in a real-world poison attack scenario.

- They also propose that the architecture of the victim matters during an attack and observed that many attacks are significantly less effective against ResNet-18 as compared to AlexNet variants (Krizhevskyet al., 2012; Shafahi et al., 2018)).
- It was also shown that proper transfer learning is less vulnerable to poison attacks. They showed that feature collision (FC) attacks, where poisons are crafted by adding small perturbations to base images (Shafahi et al.,2018), use the entire CIFAR-10 (Krizhevsky et al., 2009), training dataset for both pre-training and fine tuning. Hence, the threat model necessarily allows an adversary to modify the training dataset but only for a last few epoch.
- Further, they find that the performance of the attack is not affected by the size of the dataset. They did so by setting the percentage of poisoned data constant at 1% and changed the number of poisons and the size of the training set but no consistent trend in how attacks are affected was observed.
- The models that are trained on Standard Gradient Descent (SGD) are significantly harder to poison and data augmentation greatly reduces the effectiveness of all the attacks.
- The performance of attacks on black box models are much slower with success rates lower than 20% for FC, Convex Polytope (CP), CP attack crafts poisons such that the target's feature representation is a convex combination of the poisons' representation (Zhu et al., 2019), and HTBD (Hidden Trigger Backdoor), when poison attacks crafted remain close to the base images collide in feature space with a patched image from the target class (Saha et al., 2019).
- Triggerless poison attacks are significantly less successful when the exact target image is unknown.
- Backdoor attacks add a patch to target images to trigger misclassification and the success of the backdoor depends on the patch size used.

Although significant research has been done in finding methods to prevent poison attacks, a standardized set of indicators to measure the efficiency is much needed, which could help lay foundation to future development of novel methods with benchmarks to compare the methods against.

Conclusion

Machine learning algorithms are prone to attacks that could make them unable to do the specific task they were designed to do. In recent times, model poisoning has become a concern, however not many organizations and practitioners who rely on ML algorithms daily are aware or prepared to encounter with adversarial attacks on their systems. Though algorithms like TRIM and STRIP have been successful in securing a model against such attacks, they are yet not the ultimate solution for every kind of machine learning models out there. The lack of common background in the domain of model poisoning have made it difficult for different techniques to benchmark existing and new methods. Though attempts are being made to benchmark them, a generalized solution against model poisoning it yet to be achieved that could result in tools which can be used by organizations and practitioners to ensure the safety and security of their developed machine learning models.

References

1. M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru and B. Li, "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning," 2018 IEEE Symposium on Security and Privacy (SP), 2018, pp. 19-35, doi: 10.1109/SP.2018.00057.
2. Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. arXivpreprint arXiv:2006.12557, 2020.
3. R. S. Siva Kumar et al., "Adversarial Machine Learning-Industry Perspectives," 2020 IEEE Security and Privacy Workshops (SPW), 2020, pp. 69-75, doi: 10.1109/SPW50608.2020.00028.
4. Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In Proceedings of the 29th International Coference on International Conference on Machine Learning, ICML'12, pp. 1467–1474, USA, 2012. Omni press. ISBN 978-1-4503-1285-1. URL http://dl.acm.org/citation.cfm?id=3042573.3042761.
5. W. Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Meta poison: Practical general-purpose clean-label data poisoning, 2020.
6. Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified Defenses for Data Poisoning Attacks. *arXiv preprint arXiv:1706.03691* (2017).
7. T. Gu, K. Liu, B. Dolan-Gavitt and S. Garg, "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks," in IEEE Access, vol. 7, pp. 47230-47244, 2019, doi: 10.1109/ACCESS.2019.2909068.
8. M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
9. P. J. Huber. Robust estimation of a location parameter. *Annals of Statistics*, 53(1):73–101, 1964.
10. Y. Chen, C. Caramanis, and S. Mannor. Robust sparse regression under adversarial corruption. In *Proc. International Conference on Machine Learning*, ICML, 2013.
11. B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. Sutton, J. Tygar, and K. Xia. Exploiting machine learning to subvert your spam filter. In *Proc. First USENIX Workshop on Large-Scale Exploits and Emergent Threats*, LEET, 2008.
12. Gao, Yansong (12/2019). "STRIP a defence against trojan attacks on deep neural networks" in Proceedings of the 35th Annual Computer Security Applications Conference (1-4503-7628-2, 978-1-4503-7628-0), (p. 113). New York, NY, USA: Association for Computing Machinery.
13. Shafahi, Ali, et al. "Poison frogs! targeted clean-label poisoning attacks on neural networks." *arXiv preprint arXiv:1804.00792* (2018).
14. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.Technical report, Citeseer, 2009.
15. Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras,and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. InAdvances in Neural Information Processing Systems, pp. 6103–6113, 2018.
16. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolu-tional neural networks. InAdvances in neural information processing systems, pp. 1097–1105,2012.
17. Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks.arXiv preprint arXiv:1910.00033, 2019.

18. Shanjiaoyang Huang, Weiqi Peng, Zhiwei Jia, and Zhuowen Tu. One-pixel signature: Characterizingcnn models for backdoor detection. InECCV, 2020.
19. Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018.
20. Jonas Geiping, Liam Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, andTom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching, 2020.
21. Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. A backdoor attack against lstm-based text classificationsystems.IEEE Access, 7:138872–138878, 2019.
22. Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee,Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradientoptimization. InProceedings of the 10th ACM Workshop on Artificial Intelligence and Security,pp. 27–38. ACM, 2017.
23. Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein.Transferable clean-label poisoning attacks on deep neural nets. InInternational Conference onMachine Learning, pp. 7614–7623, 2019.
24. Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna.Bullseye polytope: A scalable clean-label poisoning attack with improved transferability, 2020.
25. Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. InProceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 1885–1894.JMLR. org, 2017.