Machine Learning System Security Guidelines, Appendix. “Overview of Detection Techniques for Machine Learning-Specific Attacks”

Version 1.03a

March 15, 2023

Editing Committee of Machine Learning System Security Guidelines
Security Working Group on Machine Learning System

Machine Learning Systems Engineering (MLSE)
Japan Society for Software Science and Technology

Contents

A-1. Introduction A-1
A-2. Classification of Detection Techniques Based on Adversary Tactics A-1
A-2.1. Precursor Detection A-2
A-2.1.1. Evasion Attack Detection A-2
A-2.1.2. Poisoning Attack Detection A-2
A-2.2. Indicator Detection A-3
A-2.2.1. Evasion Attack Detection A-3
A-2.2.2. Poisoning Attack Detection A-3
A-2.2.3. Detecting Model Extraction Attacks A-3
A-3. Data Used for Detection A-3
A-4. Summary A-7
A-5. References A-8
Members of the Editing Committee of Machine Learning System Security Guidelines A-11

A-1. Introduction

Machine learning systems are affected by machine learning-specific attacks (MLSA) that can mislead decisions or infer information about a target model. Therefore, not only general security measures but also measures against MLSA must be taken. In addition to building systems that are robust against attacks, it is also important to detect attacks.

Many techniques for detecting MLSA have been proposed. However, they are often effective only for specific data or tasks, and a definitive detection technique has not yet been established. Carlini et al. evaluated several existing techniques for detecting adversarial examples and demonstrated that none of them could withstand adaptive attacks [A-1]. Kumar et al. cited attack detection as a challenge of machine learning security and proposed that detection techniques should be easily shared among security analysts [A-2]. Therefore, appropriate detection techniques are not easily implemented without expertise in machine learning security.

In general cyber security, MITRE ATT & CK frameworks [A-3], which systematize adversary tactics and techniques, can be used to select detection techniques [A-4]. In machine learning systems, MITRE ATLAS [A-5], which provides an attack strategy specific to machine learning (Reconnaissance, ML Attack Staging, Impact, etc.), can be used. However, as far as we know, no document systematizes techniques for detecting MLSA in relation to attack strategy. Therefore, in this appendix, the attack strategy is divided into two stages: precursor (reconnaissance or preparation for attacks) and indicator (attacks that mislead decisions or infer information about a target model), and the technique for detecting each stage was classified into precursor detection and indicator detection. Furthermore, whether the data used for detection can be acquired by the machine learning system is also important information when selecting detection techniques, so the detection techniques were also arranged in terms of the data used (training data, trained models, etc.).

This appendix summarizes the literature on attack detection in terms of adversary tactics and data used to assist developers and security analysts in selecting detection techniques. The target system is an image classification system, and the target attacks are evasion attacks, poisoning attacks, and model extraction attacks (the scope will be expanded in the future).

A-2. Classification of Detection Techniques Based on Adversary Tactics

In this appendix, adversary tactics are divided into two stages: precursor (reconnaissance or preparation for attacks) and indicator (attacks that mislead decisions or infer information about a target model), and the technique for detecting each stage was classified into precursor detection and indicator detection. In this section, the detection techniques for evasion attacks, poisoning attacks, and model extraction attacks are classified into two detection purposes, precursor detection and indicator detection (a summarized list is shown in Table A-1).

A-2.1. Precursor Detection

A-2.1.1. Evasion Attack Detection

In image classification, an attacker can mislead a machine learning model by adding small noise (perturbation) to the input image [A-6]. Such intentionally perturbed data are called adversarial examples (Nicholas Carlini assembled a list of papers on adversarial examples (arXiv) [A-7]).

The method for creating an adversarial example also depends on the knowledge of the attacker. For example, an attacker who does not have knowledge of the target system may input a series of queries against the system to create an adversarial example [A-8]. On the other hand, an attacker with knowledge about the target system can prepare a substitute model offline and create adversarial examples without inputting a query to the target system [A-9]. An attack that attempts to create an adversarial example through query input, as in the former example, should be detected as an attack precursor before the adversarial example is created. The detection techniques are described below.

- Detect attacks to create an adversarial example

Since an attacker is likely to input multiple similar data to create an adversarial example, detection techniques have been proposed that exploit the fact that the series of queries inputted by the attacker are distributed differently than those of legitimate users [A-10], [A-11]. In addition, since such an attack often results in more queries than those of legitimate users or in a biased output label distribution, it might be detected using a simple method such as monitoring the query frequency per unit time or the output label distribution.

A-2.1.2. Poisoning Attack Detection

There are two types of poisoning attacks: those that intentionally degrade the model inference accuracy by injecting malicious data into the training data [A-12], and those that inject backdoor data into the training data to misclassify specific input data into the target label (backdoor attack) [A-13]. There is a risk of poisoning attacks by outsourcing data preparation or model training, data collection from untrusted websites, federated learning, and transfer learning [A-14], [A-15], [A-16], [A-17].

Such poisoning attacks should be detected at the testing stage or earlier (as precursor detection). If the operational input data are retrained, whether the data is poisoned must be checked. The detection method is described below.

- Detect poisoned datasets that degrade the model performance

Several techniques are used to detect whether training datasets contain malicious data that would degrade performance [A-18], [A-19].

- Detect backdoors

Several techniques are used to detect backdoor data in training data sets [A-20], [A-21]. For a

systematic and comprehensive review of backdoor attacks and countermeasures, see [A-17].

- Detect poisoned models

Several techniques are available for detecting whether a trained model obtained from untrusted sources is poisoned [A-22], [A-23].

A-2.2. Indicator Detection

A-2.2.1. Evasion Attack Detection

- Detect inputs of adversarial examples

As mentioned above, if attackers have knowledge of the target system, they may create an adversarial example in some way and input it into the system. To detect such attacks, input data, model inference, and intermediate layer output often must be analyzed [A-24], [A-25], [A-26], [A-27], [A-28], [A-29], [A-30], [A-31], [A-32]. However, as generally known, detecting adversarial examples is not easy [A-1]. Since there is no effective detection method for every system, multiple detection methods should be applied.

For a comprehensive review of the detection method for adversarial examples, see [A-33].

A-2.2.2. Poisoning Attack Detection

- Detect backdoor triggers injected into input data

Several techniques are used to detect a backdoor trigger injected into input data during operation [A-34].

A-2.2.3. Detecting Model Extraction Attacks

In model extraction attacks, an adversary prepares input–output pairs by inputting a series of intelligent queries into the system and generates a substitute model that behaves similarly to the victim model. The model extraction attack may infer information about a target model or be used for other attacks [A-35] (the latter attack is a precursor). Techniques for detecting a model extraction attack are shown below.

- Detect malicious queries for model extraction attacks

Since the log of a model extraction attack is likely to differ from that of a legitimate user, detecting techniques using this difference have been proposed [A-36], [A-37], [A-38], [A-39]. Alternatively, such attacks may be detected using simple techniques, such as monitoring query frequency per unit time, since the number of queries may be greater than that of normal users.

A-3. Data Used for Detection

This chapter summarizes the existing detection techniques in terms of the data used in their

implementation. The data referred to here are the training data, the trained model, the input data in operation, and the output data of the model. These data can be divided into “Data to be analyzed” and "Data used for detection (not to be analyzed)." The former data needs to be managed in association with a date and time or an account (an example of the former data: the input image that may have traces of an evasion attack), and the latter data needs to be managed appropriately because it is required when implementing detection techniques (an example of the latter data: training data used to calculate detection thresholds). Table A-1 summarizes the detection techniques from the above viewpoints. Here, the output data refer to the confidence score or the intermediate layer output. They vary depending on the detection method, so see the paper on each detection method for details.

Table A-1. Data Used for Detection

X：Data to be analyzed

Y：Data used for detection (not to be analyzed)

Purpose	**Attack**	**Detection**	**Development**		**Operation**	
			Training data	Trained model	Input data	Output data
Precursor Detection	**Creation of Adversarial Examples**	Chen et al. [A-10]	Y		X	
		Li et al. [A-11]	Y		X	
	Data Poisoning (degrades the model performance)	Müller et al. [A-18]	X	Y		
		Tavallali et al. [A-19]	X			
	Backdoor	Chen et al. [A-20]	X	Y		
		Hayase et al. [A-21]	X	Y		
		Dong et al. [A-22]		X		
		Huster et al. [A-23]		X		
Indicator Detection	**Input of Adversarial Examples**	Hendrycks et al. [A-24]	Y		X	
		Meng et al. [A-25]	Y		X	
		Grosse et al. [A-26]	Y		X	
		Gong et al. [A-27]	Y	Y	X	
		Lu et al. [A-28]	Y	Y		X
		Feinman et al. [A-29]	Y	Y		X
		Aigrain et al. [A-30]	Y	Y		X

		Xu et al. [A-31]	Y	Y	X	X
		Monteiro et al. [A-32]	Y	Y	X	X
	Backdoor Triggers	Kiourti et al. [A-34]	Y	Y	X	X
	Model Extraction	Juuti et al. [A-36]			X	
		Pal et al. [A-37]	Y		X	
		Atli et al. [A-38]	Y		X	
		Sadeghzadeh et al.[A-39]	Y		X	

A-4. Summary

This appendix summarizes techniques for detecting evasion attacks, poisoning attacks, and model extraction attacks in terms of attack strategies and data used to assist developers and security analysts in selecting detection methods. By associating detection methods with attack strategies, this appendix can be used as a reference to detect precursor and indicator attacks at multiple levels or to detect attacks as early as possible. Further, by arranging the detection method in terms of the data, even when data use is restricted (e.g., input data cannot be acquired), a detection method that use only available data can be selected. The existing review papers [A-17], [A-33] are also helpful in selecting detection methods.

A-5. References

[A-1] N. Carlini , D. Wagner, “Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods,” In Workshop on Artificial Intelligence and Security, 2017.

[A-2] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann , S. Xia, “Adversarial Machine Learning -- Industry Perspectives,” In IEEE Security and Privacy Workshops, 2020.

[A-3] “ATT&CK,” MITRE, [online]. Available: https://attack.mitre.org/.

[A-4] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley , a. R. D. Wolf, “Finding Cyber Threats with ATT&CK™-Based Analytics,” The MITRE Corporation, Bedford, MA, Technical Report No. MTR170202, 2017.

[A-5] “ATLAS,” MITRE, [online]. Available: https://atlas.mitre.org/.

[A-6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow , R. Fergus, “Intriguing properties of neural networks,” arXiv preprint arXiv:1312.6199, 2013.

[A-7] N. Carlini, “A Complete List of All (arXiv) Adversarial Example Papers,” https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html, 2019.

[A-8] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi , C.-J. Hsieh, “ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models,” In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15–26, 2017.

[A-9] I. J. Goodfellow, J. Shlens , C. Szegedy, “Explaining and Harnessing Adversarial Examples,” In International Conference on Learning Representations, 2015.

[A-10] S. Chen, N. Carlini , D. Wagner, “Stateful Detection of Black-Box Adversarial Attacks,” In Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence, pp.30-39, 2019.

[A-11] H. Li, S. Shan, E. Wenger, J. Zhang, H. Zheng , B. Y. Zhao, “Blacklight: Defending Black-Box Adversarial Attacks on Deep Neural Networks,” arXiv preprint arXiv:2006.14042, 2020.

[A-12] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru , B. Li, “Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning,” In IEEE Security and Privacy, 2018.

[A-13] T. Gu, B. Dolan-Gavitt , S. Garg, “BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain,” In Proceedings of Machine Learning and Computer Security Workshop, 2017.

[A-14] Y. Chen, X. Gong, Q. Wang, X. Di , H. Huang, “Backdoor Attacks and Defenses for Deep Neural Networks in Outsourced Cloud Environments,” IEEE Network, vol. 34, no. 5, pp. 141–147, 2020.

[A-15] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin , V. Shmatikov, “How To Backdoor Federated Learning,” In International Conference on Artificial Intelligence and Statistics, 2020.

[A-16] Y. Ji, Z. Liu, X. Hu, P. Wang , Y. Zhang, “Programmable Neural Network Trojan for Pre-Trained Feature Extractor,” arXiv preprint arXiv:1901.07766, 2019.

[A-17] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal , H. Kim, “Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review,” arXiv preprint arXiv:2007.10760, 2020.

[A-18] N. M. Müller, S. Roschmann , K. Böttinger, “Defending Against Adversarial Denial-of-Service Data Poisoning Attacks,” arXiv preprint arXiv:2104.06744, 2021.

[A-19] P. Tavallali, V. Behzadan, P. Tavallali , M. Singhal, “Adversarial Poisoning Attacks and Defense for General Multi-Class Models Based On Synthetic Reduced Nearest Neighbors,” arXiv preprint arXiv:2102.05867, 2021.

[A-20] J. Chen, X. Zhang, R. Zhang, C. Wang , L. Liu, “De-Pois: An Attack-Agnostic Defense against Data Poisoning Attacks,” In IEEE Transactions on Information Forensics and Security, 2021.

[A-21] J. Hayase, W. Kong, R. Somani , S. Oh, “SPECTRE: Defending Against Backdoor Attacks Using Robust Statistics,” In Proceedings of the 38th International Conference on Machine Learning, 2021.

[A-22] Y. Dong, X. Yang, Z. Deng, T. Pang, Z. Xiao, H. Su , J. Zhu, “Black-box Detection of Backdoor Attacks with Limited Information and Data,” In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021.

[A-23] T. Huster , E. Ekwedike, “TOP: Backdoor Detection in Neural Networks via Transferability of Perturbation,” arXiv preprint arXiv:2103.10274, 2021.

[A-24] D. Hendrycks , K. Gimpel, “Early Methods for Detecting Adversarial Images,” In Proceedings of the International Conference on Learning Representations, 2017.

[A-25] D. Meng , H. Chen, “MagNet: a Two-Pronged Defense against Adversarial Examples,” In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 135–147, 2017.

[A-26] K. Grosse, P. Manoharan, N. Papernot, M. Backes , P. McDaniel, “On the (Statistical) Detection of Adversarial Examples,” arXiv preprint arXiv:1702.06280, 2017.

[A-27] Z. Gong, W. Wang , W.-S. Ku, “Adversarial and Clean Data Are Not Twins,” arXiv preprint arXiv:1704.04960, 2017.

[A-28] J. Lu, T. Issaranon , D. Forsyth, “SafetyNet: Detecting and Rejecting Adversarial Examples Robustly,” In Proceedings of the IEEE International Conference on Computer Vision, pp. 446–454, 2017.

[A-29] R. Feinman, R. R. Curtin, S. Shintre , A. B. Gardner, “Detecting Adversarial Samples from Artifacts,” arXiv preprint arXiv:1703.00410, 2017.

[A-30] J. Aigrain , M. Detyniecki, “Detecting Adversarial Examples and Other Misclassifications in Neural Networks by Introspection,” In Proceedings of the 35th International Conference on Machine Learning, pp. 7167–7177, 2019.

[A-31] W. Xu , Y. Q. David Evans, “Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks,” In Proceedings of Network and Distributed System Security Symposium, 2018.

[A-32] J. Monteiro, I. Albuquerque, Z. Akhtar , T. H. Falk, “Generalizable Adversarial Examples Detection Based on Bi-model Decision Mismatch,” In 2019 IEEE International Conference on Systems, Man and Cybernetics, pp. 2839–2844, 2019.

[A-33] A. Aldahdooh, W. Hamidouche, S. A. Fezza , O. Deforges, “Adversarial Example Detection for DNN Models: A Review and Experimental Comparison,” arXiv preprint arXiv:2105.00203, 2021.

[A-34] P. Kiourti, W. Li, A. Roy, K. Sikka , S. Jha, “MISA: Online Defense of Trojaned Models using Misattributions,” arXiv preprint arXiv:2103.15918, 2021.

[A-35] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik , A. Swami, “Practical Black-Box Attacks against Machine Learning,” In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519, 2017.

[A-36] M. Juuti, S. Szyller, S. Marchal , N. Asokan, “PRADA: Protecting against DNN Model Stealing Attacks,” In IEEE European Symposium on Security & Privacy, pp. 512–527, 2019.

[A-37] S. Pal, Y. Gupta, A. Kanade , S. Shevade, “Stateful Detection of Model Extraction Attacks,” arXiv preprint arXiv:2107.05166, 2021.

[A-38] B. G. Atli, S. Szyller, M. Juuti, S. Marchal , N. Asokan, “Extraction of Complex DNN Models: Real Threat or Boogeyman?,” In International Workshop on Engineering Dependable and Secure Machine Learning Systems. Springer, pp. 42–57, 2020.

[A-39] A. M. Sadeghzadeh, F. Dehghan, A. M. Sobhanian , R. Jalili, “HODA: Hardness-Oriented Detection of Model Extraction Attacks,” arXiv preprint arXiv:2106.11424, 2021.

Members of the Editing Committee of Machine Learning System Security Guidelines

<Current Members>

Yoshikazu Hanatani (Toshiba Corporation)

Masazumi Hayashi (Teikyo Heisei University)

Maki Inui (Fujitsu Limited)

Fumiyoshi Kasahara (Fujitsu Limited)

Kei Kureishi (Toshiba Corporation)

Takao Okubo (Institute of Information Security)

Kentaro Tsuji (Fujitsu Limited)

Jun Yajima (Fujitsu Limited)

Nobukazu Yoshioka (Waseda University)

<Former Members>

Daiki Ichihara (NTT DATA Corporation)

Tomoko Kaneko (National Institute of Informatics)

Ikuya Morikawa (Fujitsu Limited)

Takanori Oikawa (Fujitsu Limited)

Kenji Taguchi (National Institute of Informatics)