Adversarial Machine Learning for Secure Artificial Intelligence: Challenges, Defense Techniques and Future Directions

Joga Ram Kumawat

Asst. Professor
Government Bangur PG College, Pali (Raj.), 306401

Abstract:
Adversarial machine learning studies how attackers trick artificial intelligence (AI) models and how we can build safer systems using multi-layered defenses. [1, 2]. As detailed in research on Adversarial Machine Learning and Secure Artificial Intelligence Systems, protecting AI requires continuous care across its entire lifecycle. [1]
Evasion Attacks: Hackers change input data slightly during testing to make the AI make wrong choices. [1]
Data Poisoning: Bad actors inject fake data into the training set to ruin the model.
Backdoor Insertion: Attackers hide secret triggers inside a model that only activate under specific conditions. [1]
Privacy Leaks: Thieves use model outputs to steal private training data or copy the model itself. [1, 2]

Keywords: adversarial, AI, defense, machine learning, future

Introduction

Adversarial machine learning is the study of the attacks on machine learning algorithms, and of the defenses against such attacks. [1] Machine learning techniques are mostly designed to work on specific problem sets, under the assumption that the training and test data are generated from the same statistical distribution (IID). However, this assumption is often violated in practical high-stake applications, where users may intentionally supply fabricated data that violates the statistical assumption.Most common attacks in adversarial machine learning include evasion attacks,[2] data poisoning attacks,[3] Byzantine attacks[4] and model extraction.[5]

Examples include attacks in spam filtering, where spam messages are obfuscated through the misspelling of "bad" words or the insertion of "good" words;[18][19] attacks in computer security, such as obfuscating malware code within network packets or modifying the characteristics of a network flow to mislead intrusion detection;[20][21] attacks in biometric recognition where fake biometric traits may be exploited to impersonate a legitimate user;[22] or to compromise users' template galleries that adapt to updated traits over time.

Researchers showed that by changing only one pixel it was possible to fool deep learning algorithms.[23] Others 3-D printed a toy turtle with a texture engineered to make Google's object detection AI classify it as a rifle regardless of the angle from which the turtle was viewed.[24] Creating the turtle required only low-cost commercially available 3-D printing technology.[25]

A machine-tweaked image of a dog was shown to look like a cat to both computers and humans.[26] A 2019 study reported that humans can guess how machines will classify adversarial images.[27] Researchers discovered methods for perturbing the appearance of a stop sign such that an autonomous vehicle classified it as a merge or speed limit sign.[13][28]

A data poisoning filter called Nightshade was released in 2023 by researchers at the University of Chicago. It was created for use by visual artists to put on their artwork to corrupt the data set of text-to-image models, which usually scrape their data from the internet without the consent of the image creator.[1,2,3]

McAfee attacked Tesla's former Mobileye system, fooling it into driving 50 mph over the speed limit, simply by adding a two-inch strip of black tape to a speed limit sign.[31][32]

Adversarial patterns on glasses or clothing designed to deceive facial-recognition systems or license-plate readers, have led to a niche industry of "stealth streetwear".[33]

An adversarial attack on a neural network can allow an attacker to inject algorithms into the target system.[34] Researchers can also create adversarial audio inputs to disguise commands to intelligent assistants in benign-seeming audio;[35] a parallel literature explores human perception of such stimuli.[36][37]

Clustering algorithms are used in security applications. Malware and computer virus analysis aims to identify malware families, and to generate specific detection signatures.[38][39]

In the context of malware detection, researchers have proposed methods for adversarial malware generation that automatically craft binaries to evade learning-based detectors while preserving malicious functionality. Optimization-based attacks such as GAMMA use genetic algorithms to inject benign content (for example, padding or new PE sections) into Windows executables, framing evasion as a constrained optimization problem that balances misclassification success with the size of the injected payload and showing transferability to commercial antivirus products.[40] Complementary work uses generative adversarial networks (GANs) to learn feature-space perturbations that cause malware to be classified as benign; Mal-LSGAN, for instance, replaces the standard GAN loss with a least-squares objective and modified activation functions to improve training stability and produce adversarial malware examples that substantially reduce true positive rates across multiple detectors.[4,5,6]

Emerging adversarial attacks are also highly effective in human-action recognition systems,[42] which are employed widely for real-world applications such as surveillance, autonomous vehicles, and in-house monitoring. In these attacks, attackers introduce noise into the representation of the human features to fool the recognition systems.[43] These attacks do not necessary require the access of the recognition systems themselves, and they can be inperceptible by the human users of such systems, thereby casting a significant security risk.[44]

Discussion

Researchers have observed that the constraints under which machine-learning techniques function in the security domain are different from those of common benchmark domains. Security data may change over time, include mislabeled samples, or reflect adversarial behavior, which complicates evaluation and reproducibility.[45]

Data collection issues

Security datasets vary across formats, including binaries, network traces, and log files. Studies have reported that the process of converting these sources into features can introduce bias or inconsistencies.[45] In addition, time-based leakage can occur when related malware samples are not properly separated across training and testing splits, which may lead to overly optimistic results.[45]

Labeling and ground truth challenges

Malware labels are often unstable because different antivirus engines may classify the same sample in conflicting ways. Ceschin et al. note that families may be renamed or reorganized over time, causing further discrepancies in ground truth and reducing the reliability of benchmarks.[45]

Concept drift

Because malware creators continuously adapt their techniques, the statistical properties of malicious samples also change. This form of concept drift has been widely documented and may reduce model performance unless systems are updated regularly or incorporate mechanisms for incremental learning.[7,8,9]

Feature robustness

Researchers differentiate between features that can be easily manipulated and those that are more resistant to modification. For example, simple static attributes, such as header fields, may be altered by attackers, while structural features, such as control-flow graphs, are generally more stable but computationally expensive to extract.[45]

Class imbalance

In realistic deployment environments, the proportion of malicious samples can be extremely low, ranging from 0.01% to 2% of total data. This unbalanced distribution causes models to develop a bias towards the majority class, achieving high accuracy but failing to identify malicious samples.[46]

Prior approaches to this problem have included both data-level solutions and sequence-specific models. Methods like n-gram and Long Short-Term Memory (LSTM) networks can model sequential data, but their performance has been shown to decline significantly when malware samples are realistically proportioned in the training set, demonstrating the limitations in realistic security contexts.[46]

To address this issue, one approach has been to adapt models from natural language processing, such as BERT. This method involves treating sequences of application activities as a form of "language" and fine-tuning a pre-trained BERT model on the specific task. A study applying this technique to Android activity sequences reported an F1 score of 0.919 on a dataset with only 0.5% malware samples. This result was a significant improvement over LSTM and n-gram models, demonstrating the potential of pre-trained models to handle class imbalance in malware detection.[10,11,12]

System design and learning

Feature engineering and training can causes issues. Data snooping is a common pitfall where a model is trained using information that would not be available in a real-world scenario.[47] Spurious correlations result when a model learns to associate artifacts with a label, rather than the underlying security-relevant pattern.[47] For example, a malware classifier might learn to identify a specific compiler artifact instead of

malicious behavior itself. Biased parameter selection is a form of data snooping where model hyperparameters are tuned using the test set.[47]

Performance evaluation

The choice of the evaluation metrics can impact the validity of the results. Having an inappropriate baseline involves failing to compare a new model against simpler, well-established baselines.[47] Inappropriate performance measures means using metrics that do not align with the practical goals of the system.[47] Reporting only "accuracy" is often described as insufficient for an intrusion detection system, where false-positive rates are considered critically important.[47] Base rate fallacy is a failure to correctly interpret performance in the context of large class imbalances.[47]

Deployment and operation

Deployment introduces challenges related to the performance and security in live environments. Lab-only evaluation is the practice of evaluating a system only in a controlled, static laboratory setting, which does not account for real-world challenges like concept drift and performance overhead.[47] Inappropriate threat model refers to failing to consider the ML system itself as an attack surface.[13,14,15]

Results

Adversarial attacks and training in linear models

There is a growing literature about adversarial attacks in linear models. Indeed, since the seminal work from Goodfellow at al.[78] studying these models in linear models has been an important tool to understand how adversarial attacks affect machine learning models. The analysis of these models is simplified because the computation of adversarial attacks can be simplified in linear regression and classification problems. Moreover, adversarial training is convex in this case.[79]

Linear models allow for analytical analysis while still reproducing phenomena observed in state-of-the-art models. One prime example of that is how this model can be used to explain the trade-off between robustness and accuracy.[80] Diverse work indeed provides analysis of adversarial attacks in linear models, including asymptotic analysis for classification [81] and for linear regression.[82][83] And, finite-sample analysis based on Rademacher complexity.[84]

A result from studying adversarial attacks in linear models is that it closely relates to regularization.[85] Under certain conditions, it has been shown that

adversarial training of a linear regression model with input perturbations restricted by the infinity-norm closely resembles Lasso regression, and that
adversarial training of a linear regression model with input perturbations restricted by the 2-norm closely resembles Ridge regression.[16,17,18]

Adversarial deep reinforcement learning

Adversarial deep reinforcement learning is an active area of research in reinforcement learning focusing on vulnerabilities of learned policies. In this research area, some studies initially showed that reinforcement learning policies are susceptible to imperceptible adversarial manipulations.[78][86] While some methods have been proposed to overcome these susceptibilities, in the most recent studies it has been shown that these proposed solutions are far from providing an accurate representation of current vulnerabilities of deep reinforcement learning policies.[87]

Adversarial natural language processing

Adversarial attacks on speech recognition have been introduced for speech-to-text applications, in particular for Mozilla's implementation of DeepSpeech[88] and Google's Speech Commands model.[89] Various pre-processing techniques have been proposed to defend against such attacks [90], but these methods may not be resilient to attackers aware of those defenses.

Specific attack types

There are a large variety of different adversarial attacks that can be used against machine learning systems. Many of these work on both deep learning systems as well as traditional machine learning models such as SVMs[7] and linear regression.[91] A high level sample of these attack types include:

Adversarial examples[92]
Trojan and backdoor attacks[93]
Model inversion[94]
Membership inference[95]

Adversarial examples

An adversarial example refers to specially crafted input that is designed to look "normal" to humans but causes misclassification to a machine learning model. Often, a form of specially designed "noise" is used to elicit the misclassifications. Below are some current techniques for generating adversarial examples in the literature (by no means an exhaustive list).[18,19]

Gradient-based evasion attack[8]
Fast Gradient Sign Method (FGSM)[96]
Projected Gradient Descent (PGD)[97]
Carlini and Wagner (C&W) attack[98]
Adversarial patch attack[99]

Black box attacks

Black box attacks in adversarial machine learning assume that the adversary can only get outputs for provided inputs and has no knowledge of the model structure or parameters.[16][100] In this case, the adversarial example is generated either using a model created from scratch, or without any model at all (excluding the ability to query the original model). In either case, the objective of these attacks is to create adversarial examples that are able to transfer to the black box model in question.

Conclusion

As artificial intelligence becomes woven into critical applications such as healthcare, finance, autonomous systems, and cybersecurity, adversarial threats to machine learning models have grown into one of the most pressing concerns in the field. Adversarial machine learning studies how attackers exploit weaknesses in model architectures and data pipelines, manipulating inputs to trigger misclassification, extract sensitive information, or quietly degrade system performance. This article offers a detailed overview of the security risks associated with adversarial attacks, including evasion attacks carried out at inference time, data poisoning that corrupts the training process, backdoor insertion that hides dormant triggers inside a model, and model inversion that leaks private information back out of a trained system. In response to these threats, the discussion evaluates a wide range of defense strategies designed to strengthen the robustness and reliability of AI systems, including adversarial training, robust optimization, defensive distillation, anomaly

detection, and privacy-preserving techniques such as differential privacy and federated learning. Particular emphasis is placed on weaving these defenses into every stage of the AI development lifecycle and on cultivating a threat-aware mindset before models are ever deployed into real-world environments. By drawing together current research, mathematical foundations, and practical implementation experience, this article traces the evolving landscape of adversarial machine learning and offers actionable guidance for developers, researchers, and policymakers who are working to secure AI-driven applications against increasingly sophisticated attacks.[20]

REFERENCES:

1. Kianpour, Mazaher; Wen, Shao-Fang (2020). "Timing Attacks on Machine Learning: State of the Art". Intelligent Systems and Applications. Advances in Intelligent Systems and Computing. Vol. 1037. pp. 111–125. doi:10.1007/978-3-030-29516-5_10. ISBN 978-3-030-29515-8. S2CID 201705926.
2. Goodfellow, Ian; McDaniel, Patrick; Papernot, Nicolas (25 June 2018). "Making machine learning robust against adversarial inputs". Communications of the ACM. 61 (7): 56–66. doi:10.1145/3134599. ISSN 0001-0782.[permanent dead link]
3. Geiping, Jonas; Fowl, Liam H.; Huang, W. Ronny; Czaja, Wojciech; Taylor, Gavin; Moeller, Michael; Goldstein, Tom (2020-09-28). Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching. International Conference on Learning Representations 2021 (Poster).
4. El-Mhamdi, El Mahdi; Farhadkhani, Sadegh; Guerraoui, Rachid; Guirguis, Arsany; Hoang, Lê-Nguyên; Rouault, Sébastien (2021-12-06). "Collaborative Learning in the Jungle (Decentralized, Byzantine, Heterogeneous, Asynchronous and Nonconvex Learning)". Advances in Neural Information Processing Systems. 34. arXiv:2008.00742.
5. Tramèr, Florian; Zhang, Fan; Juels, Ari; Reiter, Michael K.; Ristenpart, Thomas (2016). Stealing Machine Learning Models via Prediction {APIs}. 25th USENIX Security Symposium. pp. 601–618. ISBN 978-1-931971-32-4.
6. "How to beat an adaptive/Bayesian spam filter (2004)". Retrieved 2023-07-05.
7. Biggio, Battista; Nelson, Blaine; Laskov, Pavel (2013-03-25). "Poisoning Attacks against Support Vector Machines". arXiv:1206.6389 [cs.LG].
8. Biggio, Battista; Corona, Igino; Maiorca, Davide; Nelson, Blaine; Srndic, Nedim; Laskov, Pavel; Giacinto, Giorgio; Roli, Fabio (2013). "Evasion Attacks against Machine Learning at Test Time". Machine Learning and Knowledge Discovery in Databases. Lecture Notes in Computer Science. Vol. 8190. Springer. pp. 387–402. arXiv:1708.06131. doi:10.1007/978-3-642-40994-3_25. ISBN 978-3-642-38708-1. S2CID 18716873.
9. Szegedy, Christian; Zaremba, Wojciech; Sutskever, Ilya; Bruna, Joan; Erhan, Dumitru; Goodfellow, Ian; Fergus, Rob (2014-02-19). "Intriguing properties of neural networks". arXiv:1312.6199 [cs.CV].
10. Biggio, Battista; Roli, Fabio (December 2018). "Wild patterns: Ten years after the rise of adversarial machine learning". Pattern Recognition. 84: 317–331. arXiv:1712.03141. Bibcode:2018PatRe..84..317B. doi:10.1016/j.patcog.2018.07.023. S2CID 207324435.
11. Kurakin, Alexey; Goodfellow, Ian; Bengio, Samy (2016). "Adversarial examples in the physical world". arXiv:1607.02533 [cs.CV].
12. Gupta, Kishor Datta; Dasgupta, Dipankar; Akhtar, Zahid (2020). Applicability issues of Evasion-Based Adversarial Attacks and Mitigation Techniques. 2020 IEEE Symposium Series on Computational Intelligence (SSCI). doi:10.1109/SSCI47803.2020.9308589.

13. Lim, Hazel Si Min; Taeihagh, Araz (2019). "Algorithmic Decision-Making in AVs: Understanding Ethical and Technical Concerns for Smart Cities". Sustainability. 11 (20): 5791. arXiv:1910.13122. Bibcode:2019arXiv191013122L. doi:10.3390/su11205791. S2CID 204951009.
14. "Google Brain's Nicholas Frosst on Adversarial Examples and Emotional Responses". Synced. 2019-11-21. Retrieved 2021-10-23.
15. "Responsible AI practices". Google AI. Retrieved 2021-10-23.
16. Adversarial Robustness Toolbox (ART) v1.8, Trusted-AI, 2021-10-23, retrieved 2021-10-23
17. amarshal. "Failure Modes in Machine Learning - Security documentation". docs.microsoft.com. Retrieved 2021-10-23.
18. Biggio, Battista; Fumera, Giorgio; Roli, Fabio (2010). "Multiple classifier systems for robust classifier design in adversarial environments". International Journal of Machine Learning and Cybernetics. 1 (1–4): 27–41. doi:10.1007/s13042-010-0007-7. hdl:11567/1087824. ISSN 1868-8071. S2CID 8729381. Archived from the original on 2023-01-19. Retrieved 2015-01-14.
19. Brückner, Michael; Kanzow, Christian; Scheffer, Tobias (2012). "Static Prediction Games for Adversarial Learning Problems" (PDF). Journal of Machine Learning Research. 13 (Sep): 2617–2654. ISSN 1533-7928.
20. Apruzzese, Giovanni; Andreolini, Mauro; Ferretti, Luca; Marchetti, Mirco; Colajanni, Michele (2021-06-03). "Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems". Digital Threats: Research and Practice. 3 (3): 1–19. arXiv:2106.09380. doi:10.1145/3469659. ISSN 2692-1626. S2CID 235458519.