Exploring Zero-Day Attacks on Machine Learning and Deep Learning Algorithms

Marie Kovářová

Prague University of Economics and Business, Prague, Czech Republic

marie.kovarova@vse.cz

Abstract: In the rapidly evolving field of artificial intelligence, machine learning (ML) and deep learning (DL) algorithms have emerged as powerful tools for solving complex problems in various domains, including cyber security. However, as these algorithms become increasingly prevalent, they also face new security challenges. One of the most significant of these challenges is the threat of zero-day attacks, which exploit unknown and unpredictable vulnerabilities in the algorithms or the data they process. This paper provides a comprehensive overview of zero-day attacks on ML/DL algorithms, exploring their types, causes, effects, and potential countermeasures. The paper begins by introducing the concept and definition of zero-day attacks, providing a clear understanding of this emerging threat. It then reviews the existing research on zero-day attacks on ML/DL algorithms, focusing on three main categories: data poisoning attacks, adversarial input attacks, and model stealing attacks. Each of these attack types poses unique challenges and requires specific countermeasures. The paper also discusses the potential impacts and risks of these attacks on various application domains. For instance, in facial expression recognition, an adversarial input attack could lead to misclassification of emotions, with serious implications for user experience and system integrity. In object classification, a data poisoning attack could cause the algorithm to misidentify critical objects, potentially endangering human lives in applications like autonomous driving. In satellite intersection recognition, a model stealing attack could compromise national security by revealing sensitive information. Finally, the paper presents some possible protection methods against zero-day attacks on ML/DL algorithms. These include anomaly detection techniques to identify unusual patterns in the data or the algorithm's behaviour, model verification and validation methods to ensure the algorithm's correctness and robustness, federated learning approaches to protect the privacy of the training data, and differential privacy techniques to add noise to the data or the algorithm's outputs to prevent information leakage. The paper concludes by highlighting some open issues and future directions for research in this area, emphasizing the need for ongoing efforts to secure ML/DL algorithms against zero-day attacks.

Keywords: Zero-Day Attacks, Machine Learning, Deep Learning, Security, Models

1. Introduction

Recent advances in machine learning (ML) and deep learning (DL) have revolutionized many areas, from natural language processing through image data analysis to autonomous driving. However, these advances have also brought with them new security challenges.

The most serious attacks are those that are unknown and thus unpredictable, called zero-day attacks (Vaisla 2014). Zero-day attacks are undiscovered security vulnerabilities that hackers can exploit to attack a system. The term "zero-day" refers to the fact that the vendor or developer has only just become aware of the vulnerability i.e. has exactly zero days to fix it. A zero-day attack occurs when hackers discover a bug before the developer has a chance to fix it.

One of the most pressing areas is protection against zero-day attacks, which can target both the network infrastructure and the ML/DL algorithms themselves. An important point in this context is therefore the need to protect both the network infrastructure itself, through which the data flows, and the ML/DL algorithms that provide the processing and evaluation of the data.

2. Zero-Day Attacks

In addition to the phrase zero-day, words such as vulnerability, exploit and attack are usually used. Thus, a zero-day attack can be divided into the following areas (Kaspersky 2023)

- A "zero-day vulnerability" is a software flaw that was discovered by attackers before the manufacturer knew about it.
- "Zero-day exploits are a method hackers use to attack systems with undetected vulnerabilities.
- "A zero-day attack is the use of a zero-day exploit to corrupt or steal data from a system affected by a vulnerability.

2.1 Zero-day Attacks and ML/DL Algorithms

Zero-day attacks pose a challenge not only in protecting network infrastructure but also in protecting ML/DL algorithms.

A zero-day attack path occurs when a multi-step attack contains one or more zero-day exploits. Early detection of zero-day attack paths could enable early detection of zero-day threats. The authors (Sun et al. 2016) propose a probabilistic approach to identify zero-day attack paths and try to implement a prototype system called "ZePro". The "ZePro" system is based on a Bayesian network that is built on an instance graph. The system then detects high-probability attack instances and creates zero-day attack paths. According to the research, the system was able to effectively identify zero-day attack paths. The work of these authors (Sun et al. 2016) focuses more on the zero-day attack path within the network traffic and does not directly address attacks on applications on ML/DL algorithms. In her analysis, author did not find any work that addresses the identification of the path of zero-day attacks directly on ML/DL algorithms.

Zero-day attacks pose a serious threat because they exploit vulnerabilities that have not yet been discovered or patched. In the case of ML/DL algorithms, these attacks can cause significant breaches in the confidentiality, integrity and availability of systems that often rely on these algorithms for their operation.

2.1.1 Existing Research

Currently, there are already several researches aimed at influencing ML/DL algorithms. For example, a collective of authors (Biggio et al. 2013) have addressed zero-day attacks on Support Vector Machines (SVM) algorithms, which are often used for data classification. In this work, the authors proposed methods by which attackers can exploit vulnerabilities in SVM algorithms by creating so-called "poisoned" data that is embedded in a training set. This poisoned data can lead to reduced classification accuracy and reliability, as well as dangerous errors in the results. The authors of this paper also propose defense mechanisms against these attacks. Other research focuses on DL algorithm attacks in natural language processing (Zhang et al. 2020). The authors of (Zhang et al. 2020) show how attackers can insert subtle modifications to input data to mislead models into making incorrect decisions. These attacks can have serious consequences, for example, in introducing models for sentiment classification or generating misleading messages.

Other studies focus on the vulnerability of convolutional neural networks (CNNs) to attacks (Li et al. 2023). The authors (Li et al. 2023) focus on introducing a visual analytics approach to understanding adversarial attacks (Sciforce 2022), using two main questions:

- The first question asks which neurons are more vulnerable to attack.
- The second question answers which features are captured by these vulnerable neurons during prediction.

These attacks can be used, for example, to deceive autonomous vehicles or monitoring systems.

Another study reveals the vulnerability of Graph Neurol Networks (GNN) (Zügner et al. 2020). Where they concentrate on attacks on Graph Neural Networks (GNN) which are used for graph analysis and social networking. The authors investigate the possibilities of attacks on GNNs, including the generation of misleading graphs that can lead to incorrect analysis decisions.

2.2 Attacks Against ML/DL Algorithms

To protect against attacks on ML/DL algorithms, it is necessary to know their weaknesses and the types of possible attacks. As mentioned in the introduction, there are already several studies done on zero-day attacks on ML/DL algorithms by simulating them. For example, using "poisoned" data (Biggio et al. 2013) or using a visual analytics approach (Li et al. 2023), etc. In this chapter, the author elaborates more on possible attacks on ML/DL algorithms.

2.1.2 Data Poisoning Attack

In "Data Poisoning" attacks, according to the authors (Lin et al. 2021), attackers try to manipulate the training data to:

- reduce the overall performance (accuracy) of the ML model,
- cause misclassification of a particular sample or subset of a test sample, or

- extend training time.

Data poisoning attacks can manipulate the training process. Specifically, in data poisoning, attackers attempt to manipulate the training set by, for example, poisoning features, flipping labels, manipulating the model configuration settings, or altering the model weights to affect the learning model (Lin et al. 2021).

If the attacker has a designated target label to which a particular test sample is misclassified, the attack is called a "targeted data poisoning attack", otherwise it is an "untargeted data poisoning attack" (Lin et al. 2021).

Data poisoning attacks can be divided into two categories:

- Model skewing
- Feedback weaponization

Model skewing

In a "Model skewing" attack, the attackers want to falsify (misrepresent) data and get the organization to make the wrong decision in favour of the attacker. There are two registered variants of skewing attacks (Bursztein 2018)

a) ML data poisoning attacks

In "ML data poisoning attacks," attackers modify the training data used by a machine learning algorithm and cause the algorithm to make incorrect decisions.

b) Web analytics skewing

In "Web analytics skewing", attackers modify analytics data from platforms such as Google Analytics or Adobe Analytics by executing a large number of automated queries using bots. The goal is to make it appear that visitors to the site are taking certain actions more often than they actually are.

Feedback weaponization

Feedback weaponization is the misuse of the user feedback system to attack legitimate users and content. The attacker's goal is to either lower or raise the rating of a given application. As the author (Bursztein 2018) states, one of the worst attempts of this feedback misuse was in 2017, when a group of "4chan" users decided to destroy CNN's app ratings on the Play Store and App Store by leaving thousands of one-star reviews. This kind of attack is used for a number of reasons e.g. to destroy competitors, revenge or cover their tracks.

2.1.3 Adversarial Input Attack

An adversarial ML attack is a deceptive technique that "tricks" machine learning models using faulty input. The goal of an adversarial attack is to cause the ML/DL model to malfunction. (Sciforce 2022) Examples of possible adversarial attacks are when a self-driving car takes a stop sign as a speed limit, or when the autopilot moves in the opposite direction than intended. Other serious attacks can happen in the medical field, where medical machine learning can classify benign birthmarks as malignant. Therefore, adversarial attacks are a major threat in the future. Adversarial attacks are divided into white-box attacks, black-box attacks and grey-box attacks.

- "white-box" - the attacker has access to the ML model parameters and knows the target model completely. This knowledge of the model makes it easier for the attacker to generate hostile examples and create poisoned input data.
- "black-box" - the attacker has no access to the ML model parameters. Instead, the attacker only knows the model outputs (logit, confident score or label). These attacks are commonly observed in online learning techniques used in anomaly detection systems.
- "grey-box" - the attacker has incomplete knowledge of the overall structure of the target model.

White-box attacks mean that the attacker has access to the parameters of the model; black-box attacks, on the other hand, do not.

Mathematically, according to (Sciforce 2022), an adversarial attack looks like this:

$$f\,(x + d)! = y$$

Where the model *f* using the input *x* can produce a prediction *y* that is not equal to the model's prediction *f* with input *x*.

$$L(d) < T$$

Where *L* is a general function that measures the norm *d*, a *T* is the upper limit of that norm. *L* is the perturbation boundary that measures the magnitude of the perturbation *d*, the norm is usually used *Lp* (Sciforce 2022). Based on these parameters, there is an extensive family of algorithms that can be used to generate such perturbations. Figure 1 shows their basic distribution.

According to the example taxonomy, there can be many types of attacks. According to (Malhar 2021), the adversary threat can be modeled as in Figure 1.

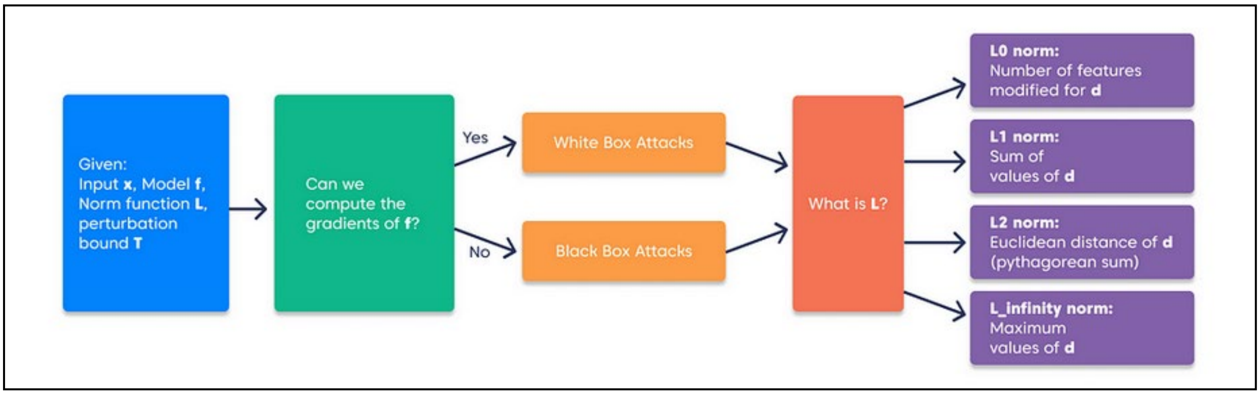
Given:
Input **x**, Model **f**, Norm function **L**, perturbation bound **T**
Can we compute the gradients of **f**?
Yes
No
White Box Attacks
Black Box Attacks
What is **L**?
L0 norm: Number of features modified for **d**
L1 norm: Sum of values of **d**
L2 norm: Euclidean distance of **d** (pythagorean sum)
L_infinity norm: Maximum values of **d**

Figure 1: Taxonomy of different adversarial examples (Sciforce 2022)

2.1.4 Gradient Access

Gradient access controls who have access to the model *f* and who doesn't have access. We can divide them into white-box adversaries and black-box adversaries.

White-box adversaries usually have full access to the model parameters, architecture, training routines and training hyperparameters, and are usually the most powerful attacks. These attacks use information from the gradient to find adversarial examples.

The black-box adversaries have little or no access to the model parameters and the model is generalized as an API. Attacks on the black-box adversaries according to (Malhar 2021) can be performed using non-gradient based optimization methods such as genetic algorithms, random search and evolutionary strategies according to (Malhar 2021).

2.1.5 Perturbation Bound

Perturbation limits determine the size of the perturbation *d* (Malhar 2021). Usually measured by some mathematical standard, here by Lp. (Poursaeed et al. 2018) Where we can classify these selected standards:

- L0 norm: represents an attack that is limited by the norm L0. It involves modifying a number of properties of the input signal to the model. These attacks are often very realistic and can be performed on real systems. A common example would be to put a sticker on a stop sign to force a self-driving car not to slow down. (Malhar 2021)
- L1 norm: represents an attack that is an upper bound on the sum of the total perturbation values. (Malhar 2021)
- L2 norm: attacks that are constrained by this norm represent upper bounds on the Euclidean distance / Pythagorean distance perturbation d. These attacks are relatively frequent, given the mathematical meaning of the norm L2 in linear algebra and geometry. (Malhar 2021)
- L_infinity norm: represents an attack limited by a norm L_infinity. This attack includes an upper bound on the maximum perturbation value d. According to (Malhar 2021), L_infinity attacks are the most studied of all, due to their simplicity and mathematical convenience in robust optimization.

According to (Malhar 2021), up to 8 different types of attacks (Table 1) can be combined within perturbation boundaries if we use the standard as a robustness metric Lp.

Table 1: Examples of commonly known adversarial examples categorized by gradient access and attack norm type (Malhar 2021)

Norm bound				
Access to compute gradients	L0 standards	L1 standards	L2 standard	L_infinity standard
White Box	SparseFool (Modas et al. 2019), JSMA (Papernot at al. 2017)	Elastic-net attacks (Chen et al. 2018)	Carlini-Wagner (Carlini 2017)	PGD (Madry 2019) i-FGSM (Goodfellow et al. 2015), Carlini-Wagner (Carlini 2017)
Black Box	Adversarial Scratches (Jere 2020), Sparse-RS (Croce et al. 2022)	-	GenAttack (Alzantot et al. 2019), Sim (Guo et al. 2019)	GenAttack (Alzantot et al. 2019), SIMBA (Guo et al. 2019)

There are several other domain-specific ways to quantify the magnitude of the disturbance *d*, but all attack norms can be generalized to all types of inputs. Although these types of attacks are exclusive to images, their general principles can be applied to any model *f*.

Adversarial attacks can be performed in different application domains (Lin et al. 2021) for example audio (Carlini 2018), text (Sato et al. 2018), network signals (Corona et al. 2013) and images (Engstrom et al. 2019).

2.3 Model Stealing Techniques Attack

ML/DL models represent intellectual property for society. ML/DL models are trained on company data such as financial transactions, medical information or user transactions. Ensuring the security of ML/DL models trained on sensitive user data, such as data related to a selected person's financial situation or health status, is very important. Therefore, such ML models can be misused to reveal sensitive user information.

"Model stealing" is known as model theft or model extraction. Like many other attacks, this one works by querying the target model with samples and using the model responses to spoof the replicated model. What is valuable about the ML/DL model is its functionality, which can be restored by stealing its trained parameters (weights *w*) or its decision boundaries. According to (Irolla 2019), model stealing can be represented by Eq:

$$y = f(x,w)$$

Where *x* represents the input and *y* output. By submitting a large number of samples to the target model and storing its responses, it is possible to collect enough equations to construct a solvable system of equations where *w* are the unknown variables to be found. This is very effective for all kinds of models, assuming that the attacker knows the dimension *w* and the architecture of the model f (i.e., the relationship between the input *x*, the weights *w* and the output *y*). Therefore, this attack works best in "grey-box" approaches where we have some information about the model. In the case where the attacker does not have any information about the model, he can use a surrogate ML model called a shadow model. This is a deep learning model that the attacker trains to learn the relationship between the inputs that are presented to the target model and its responses. Given enough inputs, the Shadow model can learn the decision boundaries of the target model, thus reproducing its functionality.

A study (Correia-Silva et al. 2018) shows that it is possible to steal a model with any type of input, even if it has no relation to the original problem. The authors managed to steal different models for facial expression recognition, general object classification and satellite intersection classification with roughly the same recovery rate using related images. By stealing a model, an attacker can achieve several different goals, for example:

- Reverse engineering: an attacker can perform an engineering analysis of the model to understand how it works, so that they can uncover sensitive information or business process secrets that the model contains.
- Attacks on privacy: ML models may contain information that reveals sensitive data about the individuals on whom the model was trained. An attacker could use this information to invade privacy.
- Creating a copy or fake model: an attacker can steal a model to create their own copy to use in a competing product or service.

- Exploiting the model for further attacks: an attacker can analyse the model to discover weaknesses that could be exploited to create targeted attacks on the system or the people who use the system.
- Financial abuse: if the model is part of a commercial product, its theft may involve a financial loss for the model owner, while the attacker may sell or exploit the technology for financial gain.
- Manipulation of models: If an attacker has access to a model, they can potentially modify or alter it to produce biased or malicious results. This can be used to discredit or disrupt the operation that the model supports.

In practice, stealing a model could look like this:

1. Gaining access to the target model: an attacker could target a web service or application that provides an interface to a model for facial expression recognition, object classification, or any other task they want to steal.

2. Generating or collecting related images: an attacker collects or generates a set of images that are similar to those used to train the target model. For example, these could be similar faces, objects, or satellite intersections.

3. Creating a new model: the attacker will use the collected or generated images as input for a new model, which will be a mimic of the target model.

4. Model tuning: an attacker can use techniques such as transfer learning or fine-tuning to fine-tune a new model to achieve similar results as the target model.

5. Attack: an attacker could then use this new model for his purposes, such as face identification, object classification or satellite intersection recognition, without having to know and use the original training data.

3. Protection Against Zero-Day Attacks on ML/DL Algorithms

There are many ways for an attacker to compromise the security of algorithms and thus network traffic. Algorithms that are part of network traffic detection and prevention systems today will be more vulnerable and therefore need to be protected from third party attacks. One of the biggest challenges is identifying new types of attacks that exploit specific vulnerabilities in the algorithms or datasets that are used to train them.

The basic methods for protecting against zero-day attacks on ML/DL algorithms include:

1. Anomaly detection: Using statistical methods and pattern recognition to detect anomalous input data that may indicate zero-day attacks (Kotu 2019). This method focuses on detecting unusual patterns of algorithm behaviour that may indicate tampering.
2. Model verification and validation: performing thorough verification and validation of ML/DL models as they are deployed and regularly monitoring their behaviour in the production environment (Sargent 2011). This can identify unusual changes in model outputs that could indicate a zero-day attack.
3. Federated learning and differential privacy: the use of techniques that minimize the transfer of sensitive data between different participants. It is a machine learning technique that trains an algorithm through several independent sessions, each using its own data set (McMahan 2017).

4. Conclusion

In this paper is provided an examination of the critical challenge of zero-day attacks on machine learning (ML) and deep learning (DL) algorithms, which is becoming increasingly significant in the field of cybersecurity. Zero-day attacks exploit previously unknown vulnerabilities and present a considerable threat to the integrity, confidentiality, and availability of systems dependent on ML/DL algorithms. These attacks can have far-reaching consequences in various domains such as facial expression recognition, object classification, and satellite intersection recognition, potentially affecting user experience, safety, and national security.

The research has categorized and detailed three primary types of zero-day attacks on ML/DL algorithms: data poisoning attacks, adversarial input attacks, and model stealing attacks. Each of these attacks presents unique challenges and requires specific strategies for mitigation.

The paper has also shed light on the different techniques used in these attacks, such as manipulating training data, injecting adversarial inputs, or stealing model parameters, all of which can severely compromise the performance and reliability of ML/DL systems.

In response to these threats, basic protection methods against zero-day attacks on ML/DL algorithms are introduced. These include anomaly detection techniques to identify unusual data patterns or algorithm behaviour, model verification and validation methods to ensure algorithmic correctness and robustness, federated learning approaches to maintain the privacy of training data, and the application of differential privacy techniques to add noise and prevent information leakage.

However, there are still open issues and is necessary to continue research in this area. The evolving nature of zero-day threats means that the field must constantly adapt and improve defensive strategies. Future research should focus on developing more robust and adaptive models, enhancing data security measures, and exploring new methodologies for early detection and prevention of zero-day attacks. As ML/DL algorithms continue to permeate various sectors, ensuring their security against zero-day attacks remains a paramount concern, demanding ongoing attention and innovation in cybersecurity research and practice.

Existing methods may not be sufficient or effective against novel and sophisticated attacks that exploit unknown vulnerabilities in the algorithms or the data. New methods should be able to identify and mitigate such attacks in a timely and robust manner. In future research, the author will focus on developing new methods to detect and prevent zero-day attacks on ML/DL algorithms.

Another aspect that can be further developed is ethical and legal frameworks for attack prevention on ML/DL algorithms. Attacks on ML/DL algorithms raise ethical and legal issues, such as privacy, accountability, liability, and trust. There is a need for clear and consistent frameworks that define the rights and responsibilities of the stakeholders involved in the development, deployment, and use of ML/DL algorithms, as well as the regulations and standards that govern their security and robustness.

This approach can be combined with promoting collaboration and information sharing among researchers, practitioners, and policymakers. Addressing the security issues of ML/DL algorithms requires a multidisciplinary and coordinated effort. This includes sharing data, tools, methods, results, and insights, as well as developing common standards and best practices.

The approach mentioned in the two paragraphs above won't stop the attacks but can create an environment that will make the attacks harder for the attackers. It can make additional protection levels in addition to new algorithmic methods, which the author of this paper wants to develop.

The main contributions of this paper lie in the innovative application of machine learning and deep learning techniques to pre-emptively identify and mitigate zero-day attacks. This proactive methodology sets it apart from related works, which often focus on reactive measures. By integrating predictive ML algorithms with DL models that refine detection processes, this paper advances the field of cybersecurity. It is imperative for future research to extend this preliminary analysis, employing a more extensive data analysis, modelling and simulations.

References

Alzantot, M., Sharma, Y., Chakraborty, S., Zhang, H., Hsieh, C.-J., & Srivastava, M. B. (2019). *GenAttack: practical black-box attacks with gradient-free optimization*. Proceedings of the Genetic and Evolutionary Computation Conference, 1111-1119. https://doi.org/10.1145/3321707.3321749

Biggio, B., Nelson, B., & Laskov, P. (2013). *Poisoning Attacks against Support Vector Machines* (arXiv:1206.6389). arXiv. https://doi.org/10.48550/arXiv.1206.6389

Bursztein, E. (2018). *Attacks against machine learning-An overview*. Elie Bursztein's Site. Retrieved June 20, 2023, from https://elie.net/blog/ai/attacks-against-machine-learning-an-overview/

Carlini, N., & Wagner, D. (2017). *Towards Evaluating the Robustness of Neural Networks*. 2017 IEEE Symposium on Security and Privacy (SP), 39-57. https://doi.org/10.1109/SP.2017.49

Carlini, N., & Wagner, D. (2018). *Audio Adversarial Examples: Targeted Attacks on Speech-to-Text*. 2018 IEEE Security and Privacy Workshops (SPW), 1-7. https://doi.org/10.1109/SPW.2018.00009

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., & Hsieh, C.-J. (2018). *EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples*. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), Article 1. https://doi.org/10.1609/aaai.v32i1.11302

Corona, I., Giacinto, G., & Roli, F. (2013*). Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Information Sciences*, 239, 201-225. https://doi.org/10.1016/j.ins.2013.03.022

Correia-Silva, J. R., Berriel, R. F., Badue, C., de Souza, A. F., & Oliveira-Santos, T. (2018). *Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data*. 2018 International Joint Conference on Neural Networks (IJCNN), 1-8. https://doi.org/10.1109/IJCNN.2018.8489592

Croce, F., Andriushchenko, M., Singh, N. D., Flammarion, N., & Hein, M. (2022). Sparse-RS: *A Versatile Framework for Query-Efficient Sparse Black-Box Adversarial Attacks*. Proceedings of the AAAI Conference on Artificial Intelligence, 36(6), Article 6. https://doi.org/10.1609/aaai.v36i6.20595

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., & Madry, A. (2019). *Exploring the Landscape of Spatial Robustness.* Proceedings of the 36th International Conference on Machine Learning, 1802-1811. https://proceedings.mlr.press/v97/engstrom19a.html.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). *Explaining and Harnessing Adversarial Examples* (arXiv:1412.6572). arXiv. https://doi.org/10.48550/arXiv.1412.6572

Guo, C., Gardner, J., You, Y., Wilson, A. G., & Weinberger, K. (2019). *Simple Black-box Adversarial Attacks*. Proceedings of the 36th International Conference on Machine Learning, 2484-2493. https://proceedings.mlr.press/v97/guo19a.html

Irolla, P. (2019). *What is model stealing and why it matters*. ML-SECURITY. https://www.mlsecurity.ai/post/what-is-model-stealing-and-why-it-matters

Jere, M., Rossi, L., Hitaj, B., Ciocarlie, G., Boracchi, G., & Koushanfar, F. (2020). *Scratch that! An Evolution-based Adversarial Attack against Neural Networks* (arXiv:1912.02316). arXiv. http://arxiv.org/abs/1912.02316

Kaspersky. (2023). What is a Zero-day Attack? - Definition and Explanation.Www.Kaspersky.Com. https://www.kaspersky.com/resource-center/definitions/zero-day-exploit

Kotu, V., & Deshpande, B. (2019). *Chapter 13-Anomaly Detection*. In V. Kotu & B. Deshpande (Eds.), Data Science (Second Edition) (pp. 447-465). Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-814761-0.00013-7

Li, Y., Wang, J., Fujiwara, T., & Ma, K.-L. (2023). *Visual Analytics of Neuron Vulnerability to Adversarial Attacks on Convolutional Neural Networks*. ACM Transactions on Interactive Intelligent Systems. https://doi.org/10.1145/3587470

Lin, J., Dang, L., Rahouti, M., & Xiong, K. (2021). *ML Attack Models: Adversarial Attacks and Data Poisoning Attacks* (arXiv:2112.02797). arXiv. https://doi.org/10.48550/arXiv.2112.02797

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2019). Towards Deep Learning Models Resistant to Adversarial Attacks (arXiv:1706.06083). arXiv. https://doi.org/10.48550/arXiv.1706.06083

Malhar. (2021). *A Practical Guide To Adversarial Robustness*. Medium. https://towardsdatascience.com/a-practical-guide-to-adversarial-robustness-ef2087062bec

McMahan, B., & Ramage, D. (2017). *Federated Learning: Collaborative Machine Learning without Centralized Training Data*. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

Modas, A., Moosavi-Dezfooli, S.-M., & Frossard, P. (2019). SparseFool: *A Few Pixels Make a Big Difference*. 9087-9096. https://openaccess.thecvf.com/content_CVPR_2019/html/Modas_SparseFool_A_Few_Pixels_Make_a_Big_Difference_CVPR_2019_paper.html

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). *Practical Black-Box Attacks against Machine Learning*. Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 506-519. https://doi.org/10.1145/3052973.3053009.

Poursaeed, O., Katsman, I., Gao, B., & Belongie, S. (2018). Generative Adversarial Perturbations. 4422-4431. https://openaccess.thecvf.com/content_cvpr_2018/html/Poursaeed_Generative_Adversarial_Perturbations_CVPR_2018_paper.html

Sargent, R. G. (2011). *Verification and validation of simulation models*. Proceedings of the 2011 Winter Simulation Conference.

Sciforce. (2022). *Adversarial Attacks Explained (And How to Defend ML Models Against Them)*. Sciforce. https://medium.com/sciforce/adversarial-attacks-explained-and-how-to-defend-ml-models-against-them-d76f7d013b18

Sato, M., Suzuki, J., Shindo, H., & Matsumoto, Y. (2018). *Interpretable Adversarial Perturbation in Input Embedding Space for Text* (arXiv:1805.02917). arXiv. https://doi.org/10.48550/arXiv.1805.02917

Sun, X., Dai, J., Liu, P., Singhal, A., & Yen, J. (2016). *Towards probabilistic identification of zero-day attack paths*. 2016 IEEE Conference on Communications and Network Security (CNS), 64-72. https://doi.org/10.1109/CNS.2016.7860471

Vaisla, K. S., & Saini, R. (2014*). Analyzing of Zero Day Attack and its Identification Techniques*.

Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). *Adversarial Attacks on Deep-learning Models in Natural Language Processing: A Survey*. ACM Transactions on Intelligent Systems and Technology, 11(3), 24:1-24:41. https://doi.org/10.1145/3374217

Zügner, D., Borchert, O., Akbarnejad, A., & Günnemann, S. (2020). *Adversarial Attacks on Graph Neural Networks: Perturbations and their Patterns.* ACM Transactions on Knowledge Discovery from Data, 14(5), 57:1-57:31. https://doi.org/10.1145/3394520