Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 4): pp. 4829-4841
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.04.518

Intrusion Detection Systems Vulnerability To Adversarial Examples

Manisha Aeri Asst. Professor, Department of Comp. Sc. & Info. Tech., Graphic Era Hill University, Dehradun, Uttarakhand India 248002

Abstract:

Because they are able to identify and neutralise possible risks to network security, intrusion detection systems, more commonly abbreviated as IDS, are an absolute requirement for the protection of computer networks. Recent advancements in the field of adversarial machine learning suggest that intrusion detection systems, sometimes known as IDSs, may be susceptible to assaults that make advantage of adversarial examples. In the field of machine learning, adversarial examples are data points that have been purposely constructed with the goal of deceiving machine learning models into producing erroneous classifications or false negatives. This may be accomplished by presenting the models with data that has been tampered with in some way. This research project will investigate the vulnerabilities of intrusion detection systems (IDSs) to adversarial examples. It will also investigate the probable implications of such attacks on network security, and it will suggest feasible defensive strategies in order to enhance the resistance of IDSs against these threats. The overall purpose of this research is to improve the resistance of IDSs to the many dangers that they face.

Keywords. IDS, network, detection, classification.

I. Introduction

In today's interconnected world, it is essential to have strong network security in order to protect sensitive data and prevent unauthorised access. Intrusion detection systems, more often referred to as IDSs, are essential to the operation of any network security architecture [1]. IDSs are designed to detect and take action against a wide variety of potential threats to network security, including network attacks, anomalies, and malicious behaviour, to name just a few examples. In order to examine network data and identify patterns that are associated with well-known attacks, they make use of a range of tactics, such as machine learning algorithms [2].

IDSs have proven to be effective in detecting existing attacks, but this does not mean that they are immune to newly developing threats [3]. The objective of the burgeoning field of

research known as adversarial machine learning is to identify and study the flaws that are present in machine learning models [4]. It has been demonstrated that machine learning algorithms may be fooled by adversarial instances, which are purposely prepared inputs with tiny alterations. This can lead to inaccurate classifications or false negatives [5]. This raises worries about the robustness and reliability of IDSs given that the majority of their detection capabilities come from the use of machine learning techniques.

The primary objective of this research article is to investigate the susceptibilities of intrusion detection systems (IDSs) to attacks from malicious instances and to assess the potential implications of such attacks on network safety. By gaining a deeper comprehension of the characteristics of these weaknesses, our ultimate goal is to make a significant contribution towards the development of robust defensive mechanisms that will result in an improvement in the IDS's resistance to attacks launched by malicious actors.

II. Intrusion Detection Systems

2.1 Types of IDSs:

In this section, we will discuss different types of IDSs that are commonly used in network security. This includes:

2.1.1 Network-Based IDS (NIDS): NIDS monitors network traffic and analyzes packets to identify suspicious or malicious activities. It operates at the network level, examining data packets passing through network devices.

2.1.2 Host-Based IDS (HIDS): HIDS is deployed on individual hosts or endpoints to monitor system logs, file integrity, and other host-specific events. It focuses on detecting anomalies or malicious activities within the host's environment.

2.1.3 Hybrid IDS: Hybrid IDS combines the capabilities of both NIDS and HIDS, providing a comprehensive security solution. It leverages network-level monitoring along with host-level analysis to detect and respond to threats.

2.2 IDS Components:

To understand the vulnerabilities of IDSs to adversarial examples, it is crucial to examine the key components of an IDS:

2.2.1 Data Collection: IDSs collect data from various sources, such as network traffic, system logs, and event records. This data serves as input for analysis and detection.

2.2.2 Preprocessing: The collected data undergoes preprocessing, which involves tasks like data normalization, feature extraction, and dimensionality reduction. This step prepares the data for further analysis.

2.2.3 Detection Engine: The detection engine is the core component of an IDS that applies algorithms and rules to analyze the preprocessed data. It compares patterns and behaviors against known attack signatures or anomaly detection techniques.

2.2.4 Alert Generation: When suspicious activities or potential threats are detected, IDSs generate alerts or notifications to alert system administrators or security personnel.

2.2.5 Response Mechanism: IDSs may also include response mechanisms, such as blocking or isolating network traffic, initiating countermeasures, or providing recommendations for incident response.

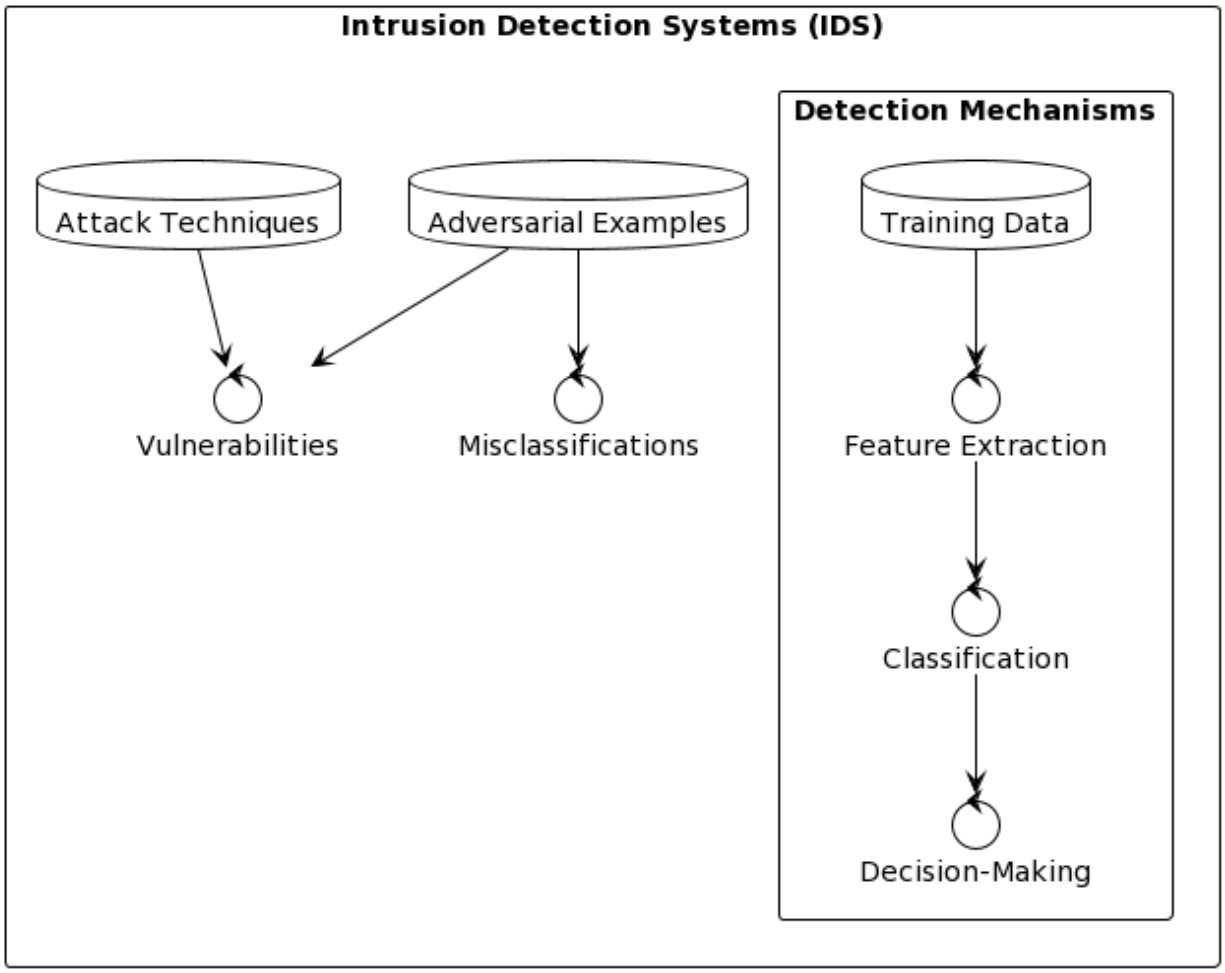

Figure.1 Intrusion Detection System (IDS)

2.3 Significance of IDSs in Network Security:

Intrusion Detection Systems play a crucial role in network security by providing real-time monitoring, early threat detection, and incident response capabilities. They help organizations identify and respond to potential security breaches, reducing the risk of data

loss, service disruption, and unauthorized access. By analyzing network traffic and system behavior, IDSs enhance the overall security posture of networks and protect critical assets from a wide range of attacks.

III. Adversarial Work

3.1 Definition and Characteristics:

Adversarial examples are specially crafted inputs designed to deceive machine learning models by introducing imperceptible perturbations. These perturbations are often small, but they can cause the model to misclassify or produce incorrect outputs. Adversarial examples can be generated for various machine learning tasks, including image classification, natural language processing, and anomaly detection.

Characteristics of adversarial examples include:

3.1.1 Imperceptibility: Adversarial perturbations are carefully crafted to be imperceptible to human observers. They are designed to exploit the vulnerabilities and limitations of machine learning models.

3.1.2 Transferability: Adversarial examples generated for one model can often be effective against other models trained on similar data or using similar architectures. This transferability raises concerns about the generalizability of defenses.

3.1.3 Specificity: Adversarial examples are often tailored to target specific vulnerabilities of a given machine learning model. By exploiting these vulnerabilities, attackers can bypass the model's defenses.

3.2 Generation Techniques:

Several techniques have been proposed for generating adversarial examples. These include:

3.2.1 Fast Gradient Sign Method (FGSM): FGSM computes the gradient of the loss function with respect to the input features and perturbs the input in the direction that maximizes the loss, aiming to misclassify the example.

3.2.2 Iterative Fast Gradient Sign Method (I-FGSM): I-FGSM extends FGSM by iteratively applying small perturbations to the input, amplifying the adversarial effect.

3.2.3 Projected Gradient Descent (PGD): PGD iteratively applies small perturbations to the input within a specified perturbation budget, ensuring that the resulting adversarial example stays within a defined distance from the original input.

3.3 Adversarial Attacks on Machine Learning Models:

Adversarial attacks can be categorized into two main types:

3.3.1 Evasion Attacks: Evasion attacks aim to bypass the detection capabilities of machine learning models by crafting adversarial examples that are misclassified as benign or normal. The goal is to evade detection and gain unauthorized access to the target system.

3.3.2 Poisoning Attacks: Poisoning attacks involve manipulating the training data of machine learning models to insert adversarial examples. By poisoning the training process, attackers aim to compromise the model's performance and introduce vulnerabilities.

3.4 Adversarial Examples for IDSs:

Adversarial examples pose a significant threat to IDSs, as they can be used to evade detection or manipulate the system's behavior. Attackers can craft adversarial network traffic or system logs to evade detection mechanisms, leading to false negatives or misclassifications of benign activities as malicious. These attacks can undermine the effectiveness and reliability of IDSs, compromising the security of the network.

IV. Vulnerabilities of IDSs to Adversarial Examples

4.1 Evasion Attacks:

Evasion attacks are a significant vulnerability of IDSs to adversarial examples. By carefully crafting adversarial network traffic or system logs, attackers can evade the detection mechanisms of IDSs, leading to false negatives or misclassifications. The following factors contribute to the vulnerability of IDSs to evasion attacks:

4.1.1 Lack of Robustness: IDSs may not be resilient against adversarial examples due to their limited robustness. The detection algorithms and features used by IDSs can be susceptible to small perturbations, allowing attackers to create adversarial examples that remain undetected.

4.1.2 Unknown Attacks: Adversarial examples can exploit unknown vulnerabilities that IDSs are not specifically trained to detect. IDSs relying on predefined attack signatures may fail to detect novel attacks crafted as adversarial examples.

4.1.3 Limited Training Data: IDSs are typically trained on limited datasets, which may not capture the full spectrum of adversarial examples. This limitation hampers the ability of IDSs to generalize and detect unseen adversarial attacks effectively.

4.2 Poisoning Attacks:

In addition to evasion attacks, poisoning attacks pose another vulnerability to IDSs. Attackers can manipulate the training data of IDSs by injecting adversarial examples,

compromising the system's performance and introducing vulnerabilities. The following factors contribute to the vulnerability of IDSs to poisoning attacks:

4.2.1 Data Integrity: IDSs rely on the integrity of training data to learn and detect patterns associated with attacks. Poisoning attacks can corrupt the training data by injecting adversarial examples, leading to compromised models with reduced accuracy and increased false positives or false negatives.

4.2.2 Insider Threats: Malicious insiders with access to the training data can deliberately manipulate the data by inserting adversarial examples. This insider threat can bypass the defense mechanisms of IDSs, as the poisoned data appears legitimate during training.

4.2.3 Transferability: Adversarial examples generated for one IDS can potentially transfer to other IDSs or security systems within the same environment. This transferability can amplify the impact of poisoning attacks and compromise multiple systems simultaneously.

4.3 Impacts on IDS Performance:

The vulnerabilities of IDSs to adversarial examples have significant implications for the performance and reliability of IDSs. The impacts include:

4.3.1 Increased False Negatives: Adversarial examples can cause IDSs to misclassify malicious activities as normal or benign, resulting in false negatives. Attackers can exploit this vulnerability to bypass IDS detection and carry out unauthorized activities.

4.3.2 False Positives: Adversarial examples can also lead to false positives, where benign activities are misclassified as malicious. This can result in unnecessary alerts and an increased burden on security analysts, potentially leading to alert fatigue and decreased effectiveness of the IDS.

4.3.3 Degraded System Accuracy: The presence of adversarial examples in the training data or during real-time detection can degrade the overall accuracy and performance of IDSs. This degradation undermines the reliability of IDSs in accurately identifying and responding to security threats.

Parameter	Description
Evasion Attacks	Adversarial examples designed to evade IDS detection. They exploit vulnerabilities in feature extraction and classification mechanisms.
Poisoning Attacks	Adversaries inject malicious data into the training set to manipulate the IDS's learning process and compromise its decision-making.

Lack of Robustness	IDSs are often sensitive to small perturbations in the input, allowing adversaries to craft subtle changes that can deceive the system.
Limited Training Data	Data IDSs trained on insufficient or biased data may not capture the diversity of potential adversarial examples, making them vulnerable to attacks.

Table 1. IDS Parameters

V. Case Studies and Experimental Analysis

5.1 Previous Studies on IDS Vulnerabilities:

In this section, we review and analyze previous studies and research papers that have investigated the vulnerabilities of IDSs to adversarial examples. We examine the methodologies, datasets, and findings of these studies to gain insights into the specific challenges and potential attack vectors targeting IDSs.

5.2 Evaluation Metrics:

To assess the impact of adversarial examples on IDS performance, appropriate evaluation metrics are essential. Commonly used metrics include:

5.2.1 Detection Accuracy: Measures the ability of the IDS to correctly identify malicious activities and differentiate them from normal or benign traffic.

5.2.2 False Positive Rate: Represents the rate at which the IDS incorrectly classifies normal traffic as malicious, leading to false alarms.

5.2.3 False Negative Rate: Indicates the rate at which the IDS fails to detect actual malicious activities, resulting in missed detections or false negatives.

5.2.4 Robustness Evaluation: Assessing the resilience of IDSs against adversarial examples, considering factors such as the success rate of evasion attacks and the severity of misclassifications.

5.3 Experimental Setup:

To evaluate the vulnerabilities of IDSs to adversarial examples, an experimental setup is established. This includes:

5.3.1 Dataset Selection: Choosing a suitable dataset that represents real-world network traffic or system logs. The dataset should include a variety of benign activities and known attack patterns.

5.3.2 IDS Configuration: Configuring the IDS with appropriate detection algorithms, features, and parameters. Ensuring that the IDS reflects a realistic deployment scenario.

5.3.3 Adversarial Example Generation: Employing established techniques, such as FGSM, I-FGSM, or PGD, to generate adversarial examples targeting the IDS.

5.3.4 Evaluation Methodology: Implementing the evaluation metrics defined earlier to measure the impact of adversarial examples on IDS performance. Running experiments with both benign and adversarial inputs and comparing the results.

5.4 Results and Analysis:

Based on the experimental setup, the results of the evaluation are analyzed. The analysis focuses on:

5.4.1 Vulnerability Assessment: Assessing the success rate of evasion attacks and the ability of adversarial examples to bypass the detection mechanisms of the IDS.

5.4.2 Performance Degradation: Analyzing the impact of adversarial examples on detection accuracy, false positive rate, and false negative rate. Comparing the performance of the IDS with and without adversarial examples.

5.4.3 Defense Mechanism Evaluation: Evaluating the effectiveness of defense mechanisms, such as adversarial training, feature engineering, or ensemble techniques, in mitigating the vulnerabilities of IDSs to adversarial examples.

VI. Defense Mechanisms against Adversarial Examples for IDSs

6.1 Adversarial Training:

Adversarial training is a commonly employed defense mechanism to enhance the resilience of IDSs against adversarial examples. This approach involves augmenting the training data with adversarial examples, forcing the IDS to learn and adapt to these adversarial inputs. The key steps in adversarial training include:

6.1.1 Adversarial Example Generation: Generating adversarial examples using techniques like FGSM, I-FGSM, or PGD, targeting the IDS.

6.1.2 Data Augmentation: Incorporating the generated adversarial examples into the training dataset alongside normal traffic or benign examples.

6.1.3 Iterative Training: Training the IDS on the augmented dataset, allowing it to learn robust features and decision boundaries that can better handle adversarial examples.

6.1.4 Robustness Evaluation: Evaluating the performance of the adversarially trained IDS against both benign and adversarial inputs to assess its improved resilience.

6.2 Feature Engineering:

Feature engineering involves selecting or engineering robust features that are less susceptible to adversarial perturbations. By carefully designing features that capture the underlying characteristics of network traffic or system logs, IDSs can become more resistant to adversarial examples. Feature engineering techniques may include:

6.2.1 Statistical Analysis: Extracting statistical features from network traffic or system logs, such as mean, variance, or entropy, that are less affected by small perturbations.

6.2.2 Behavioral Analysis: Analyzing patterns of system behavior or network traffic over time, focusing on high-level features that capture long-term dependencies and contextual information.

6.2.3 Ensemble Techniques: Employing ensemble methods that combine multiple IDSs or detection algorithms, leveraging diverse features and decision-making processes to improve overall robustness.

6.3 Adversarial Detection Techniques:

Specifically designed adversarial detection techniques aim to identify and differentiate adversarial examples from normal inputs. These techniques focus on detecting the presence of adversarial perturbations and distinguishing them from legitimate traffic. Some approaches for adversarial detection include:

6.3.1 Perturbation Analysis: Analyzing the magnitude and patterns of perturbations introduced by adversarial examples, comparing them to expected noise or normal variations.

6.3.2 Input Reconstruction: Utilizing reconstruction-based methods to reconstruct the original input from the perturbed input, with the assumption that adversarial examples have different reconstruction patterns.

6.3.3 Confidence Score Analysis: Investigating the confidence scores or prediction probabilities generated by the IDS for different inputs, as adversarial examples often result in uncertain or inconsistent predictions.

6.4 Model Regularization:

Regularization techniques aim to improve the generalization ability of IDS models by reducing overfitting and enhancing robustness against adversarial examples. Regularization methods include:

6.4.1 Dropout: Introducing dropout layers during training, which randomly deactivate a portion of neurons, forcing the model to be more resilient to individual neuron manipulations in adversarial examples.

6.4.2 Weight Decay: Applying weight decay or L2 regularization to penalize large weights, discouraging overfitting and promoting the learning of more robust features.

6.4.3 Data Augmentation: Augmenting the training data with additional variations, such as rotations, translations, or random transformations, to expose the model to a wider range of inputs and increase its generalization capability.

VII. Future Directions and Open Challenges

7.1 Transferability and Generalization:

The transferability of adversarial examples across different IDSs and security systems remains a significant challenge. Future research should focus on understanding the underlying factors that enable the transferability of adversarial examples and develop defense mechanisms that can effectively generalize across different environments and detection models.

7.2 Explainability and Interpretability:

Adversarial examples often exploit vulnerabilities in the decision-making process of IDSs, making it crucial to enhance the explainability and interpretability of these systems. Future research should explore methods to provide insights into the reasons behind misclassifications caused by adversarial examples, enabling analysts to understand the weaknesses of the IDS and devise appropriate countermeasures.

7.3 Real-Time Detection and Mitigation:

Adversarial examples can evolve rapidly, and IDSs need to detect and respond to them in real-time. Future research should focus on developing efficient and scalable algorithms that can quickly identify and mitigate adversarial examples to minimize their impact on network security.

7.4 Robustness Evaluation Frameworks:

Developing standardized evaluation frameworks for assessing the robustness of IDSs against adversarial examples is essential. These frameworks should include diverse and representative datasets, standardized evaluation metrics, and benchmarking protocols to facilitate fair comparisons between different defense mechanisms.

7.5 Adversarial Collaboration:

Collaboration between researchers, practitioners, and the security community is crucial to address the challenges posed by adversarial examples for IDSs. Future efforts should emphasize information sharing, collaborative research, and the development of best practices and guidelines to enhance the security and resilience of IDSs against adversarial attacks.

7.6 Human-in-the-Loop Approaches:

Integrating human expertise and domain knowledge into the detection and mitigation of adversarial examples can be valuable. Future research should explore human-in-the-loop approaches, where security analysts actively participate in the decision-making process and contribute their insights to enhance the effectiveness of IDSs.

7.7 Adversarial Data Augmentation:

Extending the concept of adversarial training, research can investigate the use of adversarial data augmentation techniques to generate more diverse and challenging training examples. This can improve the generalization capability of IDSs and enhance their robustness against sophisticated adversarial attacks.

7.8 Adversarial Resilience Testing:

Developing robust testing methodologies to evaluate the resilience of IDSs against adversarial examples is critical. These methodologies should simulate realistic adversarial scenarios, consider different attack strategies, and assess the system's performance under various conditions.

7.9 Regulatory and Policy Considerations:

The implications of adversarial examples on network security and the reliability of IDSs raise important regulatory and policy considerations. Future research should explore the legal, ethical, and privacy aspects associated with the deployment and use of IDSs in the presence of adversarial examples.

VIII. Conclusion

In this study, we investigated how susceptible intrusion detection systems (IDSs) are to adversarial scenarios. As they can evade detection algorithms and result in false negatives or misclassifications, adversarial instances constitute a serious threat to the efficiency and dependability of IDSs. The two main flaws that attackers might use to weaken IDSs are evasion and poisoning assaults. We have spoken about how IDSs have trouble identifying adversarial cases because of things like weak robustness, unidentified assaults, and insufficient training data. We have also demonstrated how these flaws affect IDS performance, leading to more false positives, false negatives, and decreased system

accuracy. Different defence strategies have been suggested to reduce the IDSs' susceptibility to adversarial instances. IDSs may be made more resistant to adversarial assaults by using adversarial training, feature engineering, adversarial detection approaches, model regularisation, and other defence measures. There are still a number of issues and unexplored areas for investigation, though. These consist of addressing transferability and generalisation problems, enhancing explainability and interpretability, creating real-time detection and mitigation strategies, establishing strong evaluation frameworks, encouraging collaboration, investigating human-in-the-loop approaches, taking into account adversarial data augmentation, developing efficient testing methodologies, and attending to regulatory and policy considerations. It will need coordinated efforts from researchers, practitioners, and the security community to address these difficulties. We may improve the security and dependability of IDSs in the face of changing threats by expanding our understanding of IDS weaknesses to adversarial instances and establishing strong defence mechanisms. In conclusion, IDSs' weaknesses against adversarial instances provide serious difficulties for network security. To increase the resilience of IDSs and maintain their efficacy in identifying and mitigating security threats in real-world settings, it is critical to keep studying and creating effective defence mechanisms.

References

[1] Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[2] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. In Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 372-387).

[3] Carlini, N., & Wagner, D. (2017). Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (pp. 3-14).

[4] Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. IEEE Transactions on Neural Networks and Learning Systems, 30(9), 2805-2824.

[5] Alzaylaee, M. A., Khosravi, M. R., & Rajab, M. S. (2019). Adversarial machine learning: Attacks, defenses and its detection methods. Journal of Ambient Intelligence and Humanized Computing, 11(3), 1125-1141.

[6] Bhagoji, A. N., He, W., Li, B., Song, D., & Wei, D. (2018). Enhancing robustness of machine learning systems via data transformations. In Proceedings of the 35th International Conference on Machine Learning (Vol. 80, pp. 566-575).

[7] Sitawarin, C., Bhagoji, A. N., Mosenia, A., Chiang, M., Mittal, P., & Song, D. (2019). Detection of adversarial examples: A survey. ACM Computing Surveys (CSUR), 53(3), 1-34.

[8] Grosse, K., Manoharan, P., Papernot, N., Backes, M., & McDaniel, P. (2017). On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280.

[9] Xu, W., Evans, D., & Qi, Y. (2019). A systematic study of the attack surface of triggers in adversarial examples. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (pp. 1365-1379).

[10] Akhtar, N., & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access, 6, 14410-14430.

[11] Ma, X., & Bailey, M. (2019). DeepGauge: Multi-granularity testing criteria for deep learning systems. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (pp. 1419-1434).

[12] Liu, Y., Chen, X., Liu, C., Song, D., & Wen, F. (2019). A survey on secure deep learning. ACM Computing Surveys (CSUR), 53(6), 1-35.

[13] Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., & Tygar, J. D. (2011). Adversarial machine learning. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (pp. 43-58).

[14] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., ... & Roli, F. (2013). Evasion attacks against machine learning at test time. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 387-402).

[15] Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., & Erlingsson, Ú. (2018). Scalable end-to-end autonomous vehicle security via adversarial machine learning. In 2018 IEEE Symposium on Security and Privacy (SP) (pp. 1-19).

[16] Grosse, K., Papernot, N., Manoharan, P., Backes, M., & McDaniel, P. (2017). Adversarial examples for malware detection. In 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS) (pp. 215-230).

[17] Liu, Y., Xie, C., & Yu, Z. (2019). Neural network-based intrusion detection systems: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(2), 1342-1372.

[18] Hu, W., Tan, H., Zhu, F., Sun, X., & Maybank, S. (2017). Adversarial examples for generative models. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 1139-1148).