Evasion Attacks on Deep Learning Algorithms

Benjamin Jäger
Department of Computer Science
Ulm University of Applied Sciences (THU)
Ulm, Germany
jaegbe01@thu.de

***Abstract*—With the increasing adoption of machine learning and deep learning techniques in intrusion detection systems (IDS), log-based anomaly detection has gained significant importance. At the same time, growing system complexity and rapidly increasing volumes of log data further intensify the need for automated and reliable detection mechanisms.**

Despite their high detection accuracy, machine-learning-based models are vulnerable to adversarial attacks. One particularly critical class of attacks are evasion attacks, which are performed during inference time with the goal of subtly manipulating input data in order to bypass detection. Even highly accurate deep learning models have been shown to be susceptible to such attacks.

This work focuses on evasion attacks against deep-learning-based log anomaly detection systems used in IDS. Various attack and defense strategies are examined, and existing research findings in this field are summarized and analyzed. The study considers different attacker knowledge models and reviews both gradient-based and non-gradient-based evasion techniques, as well as corresponding defensive approaches.

The analysis shows that evasion attacks can be effective even in black-box settings and that current deep learning models lack sufficient robustness. As a result, single defense mechanisms are generally insufficient to reliably mitigate such attacks.

Finally, this work argues that IDS and deep-learning-based detection systems should not be optimized solely for accuracy, but also for robustness. A defense-in-depth strategy is therefore essential for the deployment of machine-learning-based IDS in real-world security environments.

I. Introduction

With the increasing complexity and usage of IT infrastructure, network traffic continues to grow. As a result, the volume of firewall and server-generated log data increases significantly. Greater system usage inevitably leads to a higher risk of cyber attacks, which are becoming more frequent. In particular, the current geopolitical situation further amplifies this trend. Consequently, the demand for intrusion detection systems is steadily rising. However, traditional rule-based IDS suffer from several limitations, as they often do not scale well and are unable to detect previously unseen attack patterns [2, 8].

These limitations can be addressed through the use of machine learning (ML) and deep learning techniques, which are capable of automatically identifying patterns in large volumes of data, even when new attack strategies emerge. Such approaches are widely adopted in modern IDS, particularly in log-based intrusion detection, network traffic analysis, and malware or phishing detection, and have become an essential component of contemporary security systems [2, 8].

Despite their advantages, machine learning models are not inherently robust or fully secure. Since ML models base their decisions on previously learned patterns, adversaries can deliberately manipulate attack characteristics in order to evade detection. This research area is commonly referred to as adversarial machine learning. A specific class of adversarial threats, known as evasion attacks, targets the inference phase of a model and is the primary focus of this work.

While evasion attacks have been studied in other domains [2, 8], comparatively little attention has been given to their impact on log-based intrusion detection systems. Therefore, this work aims to analyze evasion attack strategies and corresponding defense mechanisms in the context of log-based IDS, highlighting their practical relevance and associated challenges.

The research question guiding this study is:

How do evasion attacks compromise deep learning models, and which defense strategies can improve their robustness against adversarial inputs?

This question focuses on the effectiveness of evasion attacks, the robustness of deep-learning-based intrusion detection systems, and the evaluation of defense strategies designed to mitigate such threats.

While intrusion detection systems serve as an important application domain in this work, the primary contribution lies in the systematic analysis of evasion attacks as a general threat to deep learning models. IDS are therefore used to illustrate practical implications rather than to define the theoretical scope of this study.

The remainder of this work is structured as follows. First, fundamental concepts of adversarial machine learning and evasion attacks are introduced. Subsequently, common attack and defense strategies are discussed, followed by an evaluation of the effectiveness of evasion attacks on intrusion detection systems. Finally, the findings are discussed and summarized in a concluding section.

II. Background and Basics

A. Adversarial Machine Learning

In order to understand evasion attacks, it is necessary to first introduce the concept of adversarial machine learning (AML). The goal of AML is to deliberately manipulate machine learning models in order to induce incorrect predictions. Adversarial attacks can be broadly categorized into two distinct

strategies, which differ in both their objectives and the phase of the machine learning pipeline they target, namely the training phase and the inference phase [6].

Attacks that target the training phase are referred to as poisoning attacks. In this scenario, the adversary injects malicious or misleading samples into the training dataset, thereby influencing the learned decision boundaries and degrading the overall model performance. As described in [6], poisoning attacks may introduce specific triggers that remain dormant during training but cause incorrect predictions once the model is deployed and used during inference. Due to their manipulation of the training process, poisoning attacks are conceptually distinct from evasion attacks, which exclusively operate at inference time.

Evasion attacks, in contrast, are performed during the inference phase and do not require any access to the training data or model retraining. The primary objective of an evasion attack is to achieve misclassification by crafting malicious input samples that are designed to bypass the model's decision boundary [6].

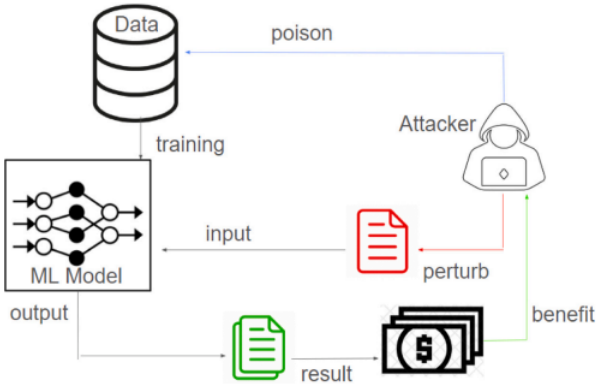

Fig. 1: Illustration of poisoning and evasion attacks against a machine learning model [3].

Figure 1 illustrates the fundamental difference between poisoning and evasion attacks. While poisoning attacks manipulate the training phase by injecting malicious samples into the dataset and thereby influencing the learned decision boundaries, evasion attacks operate exclusively at inference time by perturbing the input data to induce misclassification.

In the following, this work focuses exclusively on evasion attacks.

B. Evasion Attacks

Evasion attacks target trained machine learning models by exploiting their learned decision boundaries during inference time. Instead of modifying the training process, the adversary applies minimal perturbations to input samples in order to shift them into regions classified as benign, while keeping the modified inputs close to the original data distribution [1, 2].

In the context of log-based anomaly detection, evasion attacks typically involve minimal modifications, such as altering log keys or removing individual log events, in order to slightly shift the input across the decision boundary [2].

III. Evasion Attack Strategies

A. Attack Models

Depending on the information the adversary posses different strategies have to be chosen.

- **White-Box:** In a white-box setting, the attacker has full insight into the internal design of the detection system, including the learning method and feature processing pipeline. This knowledge allows the adversary to systematically study how the model reacts to variations in traffic characteristics. Instead of relying on random perturbations, the attacker can selectively adjust relevant traffic attributes to produce inputs that are classified as benign while still fulfilling the attacker's original objective. Such attacks typically exploit weaknesses in how the model separates normal and malicious behavior based on learned patterns rather than explicit rules [4, 5].
- **Black-Box:** In contrast, a black-box attacker has no information about the internal structure or parameters of the deployed IDS. The adversary therefore cannot directly tailor inputs to the target model. Instead, the attack relies on the empirical observation that different machine-learning-based detectors often exhibit similar sensitivities to certain input patterns. Malicious traffic samples that bypass one detector may thus also evade another, even if their internal implementations differ. This property enables effective attacks without any direct interaction with the target model's internals [4, 5].

In order to create adversarial examples (AEs), i.e., inputs that appear normal to humans but deceive a machine learning model, different attack approaches can be applied. The two main strategies are gradient-based attacks, which are mainly used in white-box settings, and non-gradient-based attacks, which are typically applied in black-box scenarios. Their key distinction lies in the use or non-use of gradient information of the loss function from the victim or a surrogate model [6].

In white-box attacks, the loss function is the original training loss of the victim model. It is known to the attacker and can be used to compute input gradients for generating adversarial examples.

$$\mathcal{L}(f(x), y) = -\sum_{i=1}^{C} y_i \log\big(f_i(x)\big) \tag{1}$$

Illustrative cross-entropy loss [7]

Gradient-based evasion attacks exploit the gradient of the loss function with respect to the input features. By computing the gradient $\nabla_x \mathcal{L}(f(x), y)$, an adversary can identify small input perturbations that increase the classification loss and push the input across the model's decision boundary. Importantly, the specific choice of the loss function is secondary, as the attack mechanism relies on gradient information rather than the semantic meaning of the input [6].

The loss function shown above is provided for illustrative purposes only and serves to highlight the general mechanism exploited by gradient-based evasion attacks.

B. Gradient-Based Attacks

In gradient-based attacks, the adversary leverages the gradient information of the loss function. It is assumed that the attacker has access to the gradients of the victim model or employs a surrogate model. In some cases, the loss function may also be estimated, making this strategy partially applicable in black-box settings. Typical characteristics of gradient-based attacks include that perturbations are directly computed from gradients and that attacks effective on one model may also succeed on another model. This property is referred to as transferability. Gradient-based attacks are generally more effective in white-box scenarios [6].

- **Fast Gradient Sign Method (FGSM):** FGSM is a one-step gradient-based attack, where a single update step is sufficient to generate an adversarial example. It works by computing the gradient of the loss function with respect to the input, taking its sign, multiplying it by a small coefficient α, and adding the result to the input. FGSM is fast and easy to implement but is often less effective than iterative approaches[1, 3, 6].
- **Basic Iterative Method (BIM):** BIM extends FGSM by applying it iteratively with small step sizes. This leads to more precise perturbations at the cost of increased computational effort [6].
- **Projected Gradient Descent (PGD):** PGD is an advanced version of BIM. The attack starts from a random initialization, and after each iteration the perturbed input is projected back into an α-bounded region. This prevents outliers and results in a strong and effective attack [3, 6].
- **Jacobian-based Saliency Map Attack (JSMA):** JSMA is based on the observation that not all features are equally important. It computes feature importance and modifies only the most influential features while minimizing the total number of changes. While JSMA is efficient for a small number of features, it incurs high computational costs for high-dimensional inputs, which limits its scalability [6].
- **Carlini & Wagner (C&W) Attack:** The C&W attack aims to find the smallest possible perturbation that successfully deceives the model. The attack is formulated as an optimization problem and utilizes the ℓ_0, ℓ_2, or ℓ_∞ norm to measure perturbation size. It is highly effective and can bypass many defense mechanisms, but it is computationally expensive [3, 6].
- **DeepFool:** DeepFool approximates the decision boundary of the model using a linear approximation and iteratively computes the minimal perturbation required to cross it. This makes DeepFool efficient, faster than C&W, and effective in generating adversarial examples [6].

C. Non-Gradient-Based Attacks

In contrast to gradient-based attacks, non-gradient-based attacks follow different strategies that do not rely on gradient information. These attacks are well suited for black-box settings and employ optimization, learning, or heuristic-based techniques [6].

- **Reinforcement Learning (RL):** In this approach, an agent learns how to deceive the victim model by modifying input features and receiving rewards when the model makes incorrect predictions. While this strategy is computationally expensive, it is highly flexible and does not require gradient access [6].
- **GAN-based Attacks:** GAN-based attacks aim to directly learn how to generate realistic adversarial examples. A generator produces perturbed inputs, while a discriminator evaluates whether the output appears real or artificial. Training instabilities and mode collapse are common challenges, which motivated the introduction of Wasserstein GANs (W-GANs) with improved cost functions to enhance data quality [6].
- **Zeroth Order Optimization (ZOO):** ZOO estimates gradients numerically by observing model outputs, without requiring access to internal model information. Inspired by the C&W attack, ZOO operates in a black-box setting but requires a large number of queries, which makes it computationally expensive [6].

In Section VI, selected surveys and empirical studies are discussed to analyze the relative effectiveness of the presented attack strategies.

IV. Evasion Defense Strategies

This section focuses on defense mechanisms against evasion attacks. While some strategies may also provide partial protection against other adversarial threats such as poisoning attacks, the primary emphasis lies on defenses targeting inference-time manipulation.

A. Defenses Against Evasion Attacks

Evasion attacks primarily target the inference phase rather than the training phase. Defense strategies can be broadly divided into model-level defenses, which aim to make the model itself more robust against adversarial manipulation, and input-level defenses, which seek to prevent attackers from manipulating the input data. Neither approach is sufficient on its own; therefore, a combination of multiple defense mechanisms is generally recommended to improve robustness against evasion attacks [1, 6].

B. Model-Level Defense

- **Adversarial Training (AT):** Adversarial training is a widely used defense mechanism in which a model is trained on both clean samples and adversarial examples. The objective is to expose the model to adversarial perturbations during training in order to learn more robust decision boundaries. This approach improves robustness against attacks such as FGSM, PGD, and Carlini–Wagner, but often leads to increased computational costs and may reduce performance on clean data. In intrusion detection system settings, adversarial training has shown to be particularly effective for models such as DNNs and XGBoost [1, 3, 6].

- **Defensive Distillation:** In defensive distillation, a student model is trained using soft labels generated by a pre-trained teacher model. This process reduces the sensitivity of the model to small input perturbations and results in smoother decision boundaries. However, prior work indicates that this defense is no longer considered state-of-the-art, as adaptive adversaries are often able to circumvent it [6].
- **Gradient Regularization / Adversarial Regularization:** This defense strategy penalizes large gradients in the loss function, thereby encouraging smoother decision boundaries and reducing sensitivity to small feature changes. Compared to adversarial training, gradient regularization is computationally less expensive but also provides a lower level of robustness against evasion attacks [1].
- **Ensemble-Based Defenses (EB):** Ensemble-based defenses rely on multiple models working jointly to detect evasion attacks or anomalies in intrusion detection systems. The underlying idea is that while a single model may be fooled, deceiving multiple diverse models simultaneously is more difficult. Although this approach increases the effort required by an adversary, it also raises the computational and operational cost for the defender [6].

C. Input-Level Defense

- **Noise Injection and Randomization:** This defense operates by introducing random noise or perturbations into the input data, thereby disrupting carefully crafted adversarial manipulations. Common techniques include Gaussian noise injection and random feature perturbations. While such methods can slightly improve robustness against evasion attacks, they do not provide complete protection and are therefore often considered baseline defenses [1, 3, 6].
- **Feature Transformation:** Feature transformation modifies input features before they are processed by the model. Typical techniques include normalization, quantization, and feature aggregation, which reduce the attack surface available to adversaries. Although these transformations can improve robustness against evasion attacks, they may also result in the loss of fine-grained information and thus affect detection performance [1, 6].

D. General Defense Strategies

In addition to input- and model-specific defenses, there are a variety of other strategies that organizations can implement to protect machine learning systems against adversarial attacks. These strategies focus on general principles regarding how models are trained, managed, and deployed.

For instance, [3] argues that access to the laboratory environment should be restricted to a minimal number of authorized personnel, and that sensitive data should always be encrypted to prevent attackers from gaining access to training data. Furthermore, models should be trained in a way that enables accurate and reliable predictions on previously unseen data, while controlling model complexity through bias-variance considerations.

Additionally, information such as training data details, employed algorithms, physical locations, and researcher identities is advised not to be made publicly available, as such disclosures may facilitate white-box attacks. The authors further suggest avoiding the use of public datasets or keeping them hidden whenever possible, thereby increasing the difficulty for adversaries to construct effective attack strategies and potentially forcing them to rely on black-box assumptions.

Moreover, organizations are encouraged to continuously test their models for potential security vulnerabilities in order to maintain effective defenses against adversarial evasion attacks. Finally, rather than publicly distributing trained models, it is recommended to provide controlled access via cloud-based environments, limiting an attacker's ability to extract architectural details [3]. These measures can further reduce the effectiveness of evasion attacks by limiting the attacker's available knowledge.

V. Evasion Attacks on Intrusion Detection Systems

Machine learning models are widely deployed in modern intrusion detection systems and cyber-physical systems to protect network infrastructures and servers from malicious activities. Typical application areas include malware detection in PDF files [5], e-mail surveillance for phishing attacks based on textual features, and log-based firewall analysis [8]. In these systems, machine learning models are tasked with classifying observed activities as benign or malicious in order to prevent unauthorized access.

In log-based intrusion detection systems, this task is commonly referred to as log anomaly detection. Such systems rely on features extracted from log data, including source and destination ports, elapsed time, transferred bytes, and similar attributes [8]. Attacks are identified by detecting abnormal correlations or temporal patterns within these features. For example, a brute-force attack typically manifests as a large number of authentication attempts originating from a single IP address within a short time frame and targeting a specific service or subsystem.

Although log-based machine learning models often achieve very high classification accuracy under benign conditions, they are particularly susceptible to evasion attacks. Random Forest (RF) and K-Nearest Neighbors (KNN), which are among the most commonly used models for firewall log classification, have been reported to achieve accuracies of up to 99% in non-adversarial settings. However, as illustrated in Table 1, a recent study from 2024 shows that classification accuracy drops significantly under adversarial conditions, reaching approximately 82% for KNN and 84% for RF [8].

Furthermore, when targeted evasion attacks are applied, performance degrades even more severely, with accuracy decreasing to 75% for RF and 79.55% for KNN. In addition to accuracy loss, other evaluation metrics such as precision, recall, F-measure, and ROC score are also negatively affected.

TABLE I: Classification accuracy before and during evasion attacks. Values were taken from [8]

Model	Accuracy (Benign)	Accuracy (Evasion)
Random Forest (RF)	84.09%	75.00%
K-Nearest Neighbors (KNN)	81.82%	79.55%

These results demonstrate that even widely used and high-performing log-based machine learning models remain vulnerable to evasion attacks, highlighting the practical relevance of robust defense mechanisms for intrusion detection systems.

A. Examples of Log Manipulation

In the following, illustrative examples of log manipulation are discussed, based on prior work in the field of adversarial machine learning [2, 8].

In an intrusion detection system, machine learning models learn to classify events based on temporal patterns and statistical correlations observed in log data. A typical attack scenario is a brute-force attack, which is characterized by a large number of similar log entries originating from the same IP address within a short time period.

In log-based anomaly detection systems, evasion attacks aim to minimally modify the log stream in order to avoid triggering malicious classifications. Such manipulations include removing individual log events, replacing them with syntactically valid but semantically benign log templates, or slightly modifying numerical features such as timestamps or transferred bytes [2, 8]. It is crucial that these modifications remain minimal and preserve the overall validity of the log data. In particular, the payload itself is typically not manipulated, as this would increase the risk of detection.

Even small perturbations can be sufficient to cause misclassification, as they may shift the input across the model's decision boundary. This is because log-based machine learning models do not reason about the semantic meaning of events, but instead rely on learned statistical patterns and feature correlations. As a result, carefully crafted modifications can cause malicious log sequences to be classified as benign [1].

This example highlights why high detection accuracy alone is insufficient and motivates the need for robust defense mechanisms, which are discussed in the following section.

VI. Discussion

A. Comparison of realistic attack scenarios (white-box vs. black-box)

In order to assess the relevance of previously discussed attack and defense strategies, it is essential to consider realistic threat scenarios. White-box attacks assume full knowledge of the target model, including its architecture and parameters, which makes them largely impractical in real-world intrusion detection systems. In contrast, black-box attacks rely solely on observing the input-output behavior of a system and therefore represent a significantly more realistic threat model in IDS environments.

As a result, white-box attacks primarily serve as a worst-case benchmark to evaluate the theoretical vulnerability of machine-learning-based log analysis systems, whereas black-box attacks better reflect practical adversarial capabilities. In such scenarios, attackers aim to craft malicious inputs that resemble legitimate log patterns in order to evade detection. Notably, prior studies indicate that evasion attacks can still be successful even without access to the internal model structure, although their effectiveness is typically reduced compared to white-box settings [1, 2].

B. Practical effectiveness of defense mechanisms

The primary objective of defense mechanisms against evasion attacks is to improve model robustness while maintaining acceptable performance on clean data. However, achieving this goal is challenging due to an inherent trade-off between robustness and accuracy. Strengthening defenses often leads to a reduction in performance on benign inputs, in addition to increased computational cost and latency.

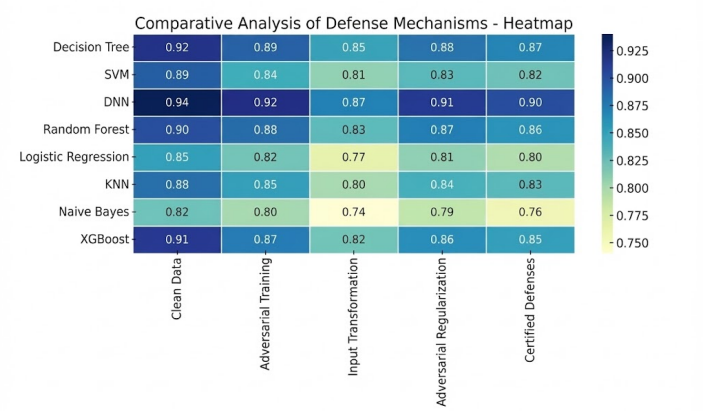

Fig. 2: Comparison of defense mechanisms across multiple machine learning models under adversarial conditions [1].

The heatmap illustrates the performance of multiple models on clean data after applying different defense strategies. It can be observed that adversarial training achieves the highest level of robustness against evasion attacks, while only slightly reducing accuracy on benign data. Input transformation methods provide limited protection and only moderate robustness improvements. Adversarial regularization offers a balanced trade-off between robustness and performance, whereas certified defenses, which represent a model-level defense approach, provide strong theoretical guarantees at the cost of reduced accuracy.

Overall, no single defense mechanism offers absolute protection against evasion attacks. Instead, the results highlight the persistent trade-off between robustness and clean-data performance in machine-learning-based intrusion detection systems [1].

To improve readability and facilitate comparison, the numerical results shown in the heatmap are also reported in Table II.

TABLE II: Comparative Analysis of Defense Mechanisms. Values were taken from [1]

Model	Clean	Adv. Train.	Input Transf.	Adv. Reg.	Cert. Def.
Decision Tree	0.92	0.89	0.85	0.88	0.87
SVM	0.89	0.84	0.81	0.83	0.82
DNN	0.94	0.92	0.87	0.91	0.90
Random Forest	0.90	0.88	0.83	0.87	0.86
Logistic Regression	0.85	0.82	0.77	0.81	0.80
KNN	0.88	0.85	0.80	0.84	0.83
Naive Bayes	0.82	0.80	0.74	0.79	0.76
XGBoost	0.91	0.87	0.82	0.86	0.85

C. Limitations of log-based benchmark datasets

A major limitation in evaluating evasion attacks on log-based intrusion detection systems is the absence of standardized and representative benchmark datasets. Due to privacy and security concerns, real-world log data is often not publicly available, forcing researchers to rely on synthetically generated logs and simulated attack scenarios. As a consequence, these datasets may fail to capture the full complexity and diversity of real-world adversarial behavior, which limits the generalizability of experimental results.

Furthermore, log data is highly context-dependent. For example, firewall logs differ substantially from system or application logs in both structure and semantics. This variability makes it difficult to design models that generalize across different environments and complicates the evaluation of defense mechanisms against evasion attacks [2].

D. Cat-and-mouse dynamic between attackers and defenders

Defense against evasion attacks is inherently dynamic. As new defense mechanisms are developed, adversaries continuously adapt their strategies to circumvent them. This ongoing arms race results in a cat-and-mouse dynamic, in which fixed or static defenses are unlikely to remain effective over extended periods of time.

Consequently, intrusion detection systems must be continuously monitored, updated, and retrained to remain resilient against evolving attack techniques. Prior work emphasizes that adaptive adversaries are often capable of bypassing even state-of-the-art defenses if sufficient time and feedback are available [1, 2].

E. Combining multiple defense strategies

It is widely accepted in adversarial machine learning research that no single defense mechanism can effectively protect against all evasion attacks. Instead, organizations aiming to secure machine-learning-based intrusion detection systems should adopt a defense-in-depth approach. This includes combining robust model architectures, adversarial training, input validation techniques, and system-level security measures.

By leveraging multiple complementary defense strategies, the overall robustness of the system can be significantly improved compared to relying on a single method. Such layered defenses reduce the likelihood that an attacker can successfully exploit a single weakness within the system [1, 8].

VII. Conclusion

The motivation of this work was to highlight the increasing relevance of machine learning and deep learning in intrusion detection systems, with a particular focus on the importance of log-based anomaly detection. This work analyzed evasion attacks with an emphasis on ML- and DL-based IDS and discussed a variety of attack and defense strategies.

In summary, evasion attacks are a highly effective means of misleading machine learning models, even when only minimal perturbations are applied. Furthermore, evasion attacks represent a realistic threat in black-box settings. It was shown that log-based intrusion detection systems are generally not considered highly robust, and that even high-performing and high-accuracy models such as Random Forest and K-Nearest Neighbors are susceptible to evasion attacks.

The research question, *How do evasion attacks compromise deep learning models, and which defense strategies can improve their robustness against adversarial inputs?*, is addressed by the analysis presented in this work. The results indicate that evasion attacks compromise models by exploiting decision boundaries, leveraging the concept of transferability - where attack strategies effective against one model may also succeed against another - and through realistic manipulation of log features. Model robustness can be improved through various defense mechanisms. Adversarial training is among the most effective approaches, while ensemble-based and regularization methods can further enhance robustness at the cost of increased computational overhead. Input-level defenses, however, provide only limited effectiveness. Ultimately, no single defense strategy offers complete protection, and only a combination of multiple defenses can provide a reasonable level of security.

The findings of this work are relevant because intrusion detection systems should not be optimized solely for accuracy, but also for robustness against evasion attacks. A defense-in-depth approach, combining multiple complementary defense mechanisms, should therefore be adopted. This is particularly relevant for real-world security operations and productive machine-learning-based systems.

Finally, it should be noted that research in this area is challenged by the lack of publicly available real-world log datasets, which often necessitates the use of synthetic benchmarks. Future work could focus on the use of real-world log data, adaptive and online defense mechanisms, and the integration of explainability techniques or detection pipelines.

REFERENCES

[1] N. Jehan, N. M. Ansari, Z. Ashraf, M. A. Bashir, H. Gul, and A. Raza, "Adversarial machine learning for cyber security defense: Detecting model evasion, poisoning attacks, and enhancing the robustness of ai systems," *Global Research Journal of Natural Science and Technology*, vol. 3, no. 2, pp. 261–200, 2025.

[2] D. Herath and A. Foo, "Real-time evasion attacks against deep learning-based anomaly detection from distributed system logs," in *Proceedings of the 2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2021.

[3] R. Muthalagu, P. Pawar *et al.*, "Detection and prevention of evasion attacks on machine learning models," *Expert Systems with Applications*, vol. 266, p. 126044, 2025.

[4] Q. Yan, X. Li, W. Zhang, R. Wang, H. Li, X. Zhao, F. Li, and X. Lin, "Automatic evasion of machine learning-based network intrusion detection systems," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 159–173, 2021.

[5] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2013)*, ser. Lecture Notes in Artificial Intelligence, vol. 8190. Springer, 2013, pp. 387–402.

[6] S. Wang, R. K. L. Ko, G. Bai, N. Dong, T. Choi, and Y. Zhang, "Evasion attack and defense on machine learning models in cyber-physical systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 930–971, 2024.

[7] E. Ichetovkin and I. Kotenko, "A technique for protecting machine learning components of intrusion detection systems from evasion attacks," in *2025 International Russian Smart Industry Conference (SmartIndustryCon)*. IEEE, 2025, pp. 736–740.

[8] M. Permpoon *et al.*, "Analysis of classification models of firewall log data attacked by data poisoning and evasion attacks," in *Proceedings of the 2024 9th International Conference on Business and Industrial Research (ICBIR)*, 2024, pp. 1419–1424.