Detecting Data Poisoning Attacks on Security Datasets using Machine Learning

1st Hassan Raza Khan
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
hassanrazakhan1914@gmail.com

2nd Muhammad Awais Khan
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
ranasanwal721@gmail.com

3rd Aoun Muhammad
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

4th Umar Fayyaz
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
umar.fayyaz@iub.edu.pk

5th Sehrish Raza
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
sehrishraza6@gmail.com

***Abstract*—Machine Learning (ML) has been established as an important area within the context of present-day cybersecurity solutions, particularly in the context of intrusions and attack detection systems. The mechanism associated with the algorithms has been proved to be highly dependent on the quality associated with the data on which it operates. The impact associated with the poisoning carried out on the data by the attacker has been established to be highly negative within the context of the mechanism associated with these algorithms. This paper has addressed the problem related to the impact on Machine Learning-based intrusions detectors. A publicly available dataset within the context of the concerned area has been acquired from Kaggle. Simulated poisoning attacks have been done on the acquired dataset to represent the behavior of the attacker. The purpose has been achieved by the use of the Isolation Forest Algorithm in combination with the analysis of label consistency. The analysis has disclosed the fact that the poisoning of the data has adversely affected the performance, while the elimination of anomalous points has improved the performance. This has established the concept that integrity is the most important factor in the realm of ML-enabled cybersecurity systems..**

***Index Terms*—Machine Learning, Data Poisoning Attacks, Cybersecurity, Intrusion Detection, Anomaly Detection, Isolation Forest**

I. Introduction

Machine Learning (ML) is being increasingly integrated into contemporary cybersecurity frameworks because it has the potential to learn automatically from enormous data and identify complex, unknown threats [1], [2].Machine learning algorithms have been highly utilized in intrusion detection systems (IDS), algorithms for worm detection, spam filters, anomaly detection, and network flow analysis algorithms [3], [4]. Compared to other systems that depend upon a fixed signature, machine learning algorithms possess the ability to adjust to dynamic attacks, ensuring a higher detection rate [5].

Nevertheless, the rising dependence on ML has led to the expansion of the attack surface for cybersecurity systems as well. Instead of focusing on the attack on the operational systems, attackers now target the training process involved in machine learning systems [6]. This has led to the development of a novel stream of research referred to as the ‘’-learning ” or ‘’-learning” [7], [8], commonly referred to as the ‘’-Learning” or ‘’-Learning” [7], [8]. Within the ‘’-Learning” attack category, one of the most critical attacks is referred to as the ‘’-Poisoning” attacks or ‘’-Poisoning” attacks [9]–[11].

Data Poisoning Attacks: These are targeted attempts to harm the training process of an ML model by introducing malicious samples [12], [13]. Because of the nature of learning, where the model depends entirely on the samples, introducing just a few samples could affect the learning process greatly [14]. For instance, in the case of cyber threats, if some samples of malicious traffic are labeled with labels of legitimate samples, an IDS system could learn wrongly and, in the end, fail to identify real cyber threats [15]. This could lead to lack of confidentiality, availability, and integrity of critical infrastructure platforms..

This is further exacerbated by modern data collection practices. Most cybersecurity datasets are automatically collected from distributed sensors, logs, cloud platforms, or third-party repositories, with little human verification
citeb16. Publicly available datasets, shared to benchmark and reproduce research findings, might already be contaminated

DOI: https://doi.org/10.67535/tsp.000002.007
Published under the ISAISS 2026 Proceedings

with noise, bias, or mislabeled instances
citeb17. All this makes cybersecurity ML models particularly vulnerable to poisoning attacks exploiting weaknesses in data trust mechanisms and validation processes
citeb18.

Various defense strategies have been discussed in current studies on data poisoning, involving robust training algorithms citeb19,b20, federated learning-based defenses
citeb21,b22, and explainable AI approaches
citeb23. While these methods seem promising, many of them result in high computational costs, poor scalability, or require assumptions on the attacker's knowledge, which may not be valid in a real cybersecurity scenario
citeb24. Besides, several studies focus on theoretical attack models or domain-specific datasets, limiting their practical applicability to general intrusion detection scenarios
citeb25.

On the other hand, data sanitization and anomaly-based approaches are shown to be a feasible and model-agnostic defense strategy [26]. Since anomalous samples are identified before training the model, it is possible to remove the tainted data points without modifying the learning algorithm. Using unsupervised anomaly detectors such as Isolation Forest [27] and analysis of label consistency, it is possible to identify anomalous patterns in the security dataset [28].

In this research work, the same defense method of the model being pre-trained will be utilized to test its validity on a practical cybersecurity intrusion dataset. Through the comparison of model results on clean data, poisoned data, and cleaned data, the research work will prove how early poisoning removal could lead to the improved reliability of ML-based cybersecurity systems.

A. Problem Statement

Despite the pervasiveness of machine learning in the cybersecurity field, existing models for intrusion detection presume that the training data is clean and reliable [29]. In reality, security datasets could contain maliciously manipulated or improperly labeled samples as a result of automated data gathering, insider attacks, or other adversarial activities [30]. This is because traditional log analysis and rule-based methods lack the necessary intelligence, scalability, and adaptability for the detection of these covert data poisoning attacks [31]. Consequently, machine learning models trained on such poisoned samples could potentially yield a deterioration in model accuracy, enhanced false negatives, and flawed decision-making, which could pose serious threats to the operations of cyberspace security systems in general [32].

B. Research Objectives

The major aims of the research work are as follows:
itemize
The effect of data poisoning in machine learning models employed for intrusion detection in cyber security.
To identify the poisoned or outlier samples present within the security datasets via Isolation Forest and label consistency checks.

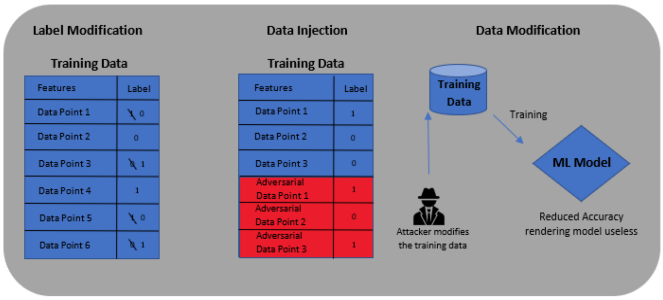

Fig. 1. Overview of common data poisoning attack types affecting machine learning during training.

TABLE I
OVERVIEW OF POISONING ATTACK TYPES, DEFENSE METHODS, AND LIMITATIONS

Attack Type	Defense Method	Main Limitation
Label Flipping	Label checks, RONI	May remove clean data
Data Injection	Outlier detection	Threshold sensitive
Feature Manipulation	Robust training	Hard to detect slowly
Backdoor Attacks	Anomaly detection	High computation
Evasion Attacks	Adversarial training	Poor generalization

Comparison of model performance on clean data, poisoned data, and cleaned data using standard metrics such as accuracy, precision, recall, and F1 score.
Ensuring the integrity of datasets by identifying and deleting poisoning data using AI-based systems for better robustness of ML-based cybersecurity systems.

II. RELATED WORK

Machine learning (ML) models are now ubiquitous in the area of cybersecurity, healthcare, finance, and many other areas [33], [34]. But making them dependent on training data also makes them susceptible to data poisoning attacks (DPAs), where an attacker intentionally tries to corrupt the training data to reduce the performance of the ML models or change their behavior [35], [36]. Because of this, DPAs have received much attention in the adversarial machine learning community [37]–[39].

Fig. 1 summarizes common data poisoning attack types and characteristics.

Table I presents an overview of common poisoning attack types along with typical defense methods and their main limitations discussed in the literature.

Traditional research works tried to classify poisoning attacks into categories depending on attacker goals and levels of data accessibility [40]. Some research works identify poisoning attacks as non-targeted, targeted, and backdoor poisoning attacks and explain different impacts on the reliability of models [41], [42]. Though the classifications are useful as basics, they might not include many types of poisoning attacks as experienced in practical models [**?**].

Later studies further widened the applicability of poisoning attacks with works on the taxonomy of poisoning attacks in both centralized and federated learning settings [43]. These

works further included new attack models such as label flipping attacks and poisoning sample injection attacks. While these are very useful works, they lack the applicability to the broader cybersecurity dataset because they are focused on learning paradigms.

Other more thorough studies investigated poisoning attacks from more than one angle, such as the objective of poisoning attacks, the knowledge available to the model, as well as the level of access the attacker has [9]. The interconnection between poisoning attacks has been investigated, including the strengths and weaknesses of several defensive strategies [11]. Despite this, most are still limited in the scope of the attacks they are considering, exploring only a few types [8] [b13], or are mostly concerned with image recognition tasks [10].

In view of the growing complexity in poisoning attacks, current literature has investigated the application of the following defensive techniques: outlier sanitization, reject-on-negative-impact (RONI), robust training, and anomaly detection-based techniques [12]. The idea in outlier detection techniques, such as the ones that are either cluster-based or statistical in nature, is to remove the suspected outliers before training. However, such techniques are very sensitive to the choice of the threshold value; thus, they can be easily mitigated through the application of the gradient-based poisoning attack [19].

More complex defense systems make use of machine learning-based detection mechanisms such as autoencoders, decision trees, and support vector machines (SVMs) to spot irregular or harmful training data [28]. Although these methods are effective in improving the accuracy of the detection mechanism, they are highly sensitive to the representation of the feature set and are dependent on the hyperparameters. Moreover, the methods currently demonstrated are tested on simulated or domain-specific data sets.

Although there is a gradually increased number of literature on these topics, existing works tend to encompass a wide variety of poisoning attacks but lack practical implementation capability and concentrate on a narrow variety of attacks with poor generalizability. Moreover, there exists a literature gap concerning works that include a variety of poisoning attacks and test a practical subset experimentally on existing cybersecurity datasets.

In contrast, this research offers a systematic survey of typical data poisoning attack types, as well as an experimental evaluation of some of those types, namely label tampering and data injection, based on anomaly detection methods, using realistic cybersecurity data sets.

III. Methodology

For this proposed research, a methodological approach to mitigate data poisoning attacks on machine learning-based intrusion detection systems has been discussed. This approach involves a number of steps, which include data preparation, preprocessing, simulation of data poisoning attacks, model formulation, analysis through anomalies, data cleaning, and analysis and evaluation. This approach has been demonstrated through an experiment [12].

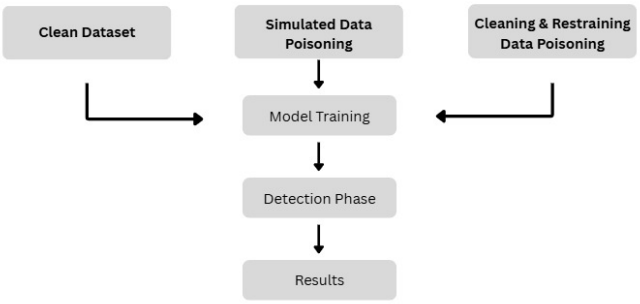

Fig. 2. Overview of the experimental workflow adopted in this study, showing model training on clean data, simulation of data poisoning, detection of anomalous samples, and subsequent data cleaning and retraining to assess performance degradation and recovery.

A. Dataset Description

These experiments are implemented based on the Cybersecurity Intrusion Detection Dataset, which is publicly available at the Kaggle platform [5]. These include features such as packet size, session duration, failed login attempts, protocol type, usage of encryption, browser type, which are numerical as well as categorical features respectively. These features are also the target point to be attacked while poisoning an environment based on the support machine learning system [3].

B. Data Preprocessing

The following preprocessing steps are done to make the data quality and consistency good enough for training: removing records with duplicate values, dropping instances containing missing values, and removing irrelevant identifiers that may include bias-session IDs.

The continuous variables are Min Max scaled to ensure all variables have values on a common scale for easy convergence of the model. The categorical variables are one-hot encoded for use in machine learning algorithms. Additionally, the produced dataset is split into the train and test datasets in a 70:30 ratio; the train dataset will be used for both learning and evaluation.

C. Baseline Model Training

A Logistic Regression classifier acts as the baseline classifier for the intrusion detection system since it requires less complexity on the part of the system with high utilization in different security contexts [4]. The Logistic Regression classifier is then trained on the clean set of training data as well as tested on the test dataset. The performance metrics are then calculated through the typical metrics such as accuracy, precision, recall, and the F1 measure, which form the basis for comparison later on.

D. Data Poisoning Simulation

A data poisoning attack is artificially induced into the training data in order to create simulated adversarial environment settings. About 10% of the attack samples are relabeled as benign-a common approach to simulate label-flipping attacks

used to deteriorate classifier performance
citeb13. Besides, feature poisoning is carried out by injecting selected numerical features of a subset of training samples with Gaussian noise. These poisoned samples are added only into the training set, while the test set stays clean to objectively assess performance degradation.

E. Model Training on Poisoned Data

The Logistic Regression model is then trained again using the poisoned data. The set of evaluation metrics is computed again using the clean test data to assess the impact of the poisoning data. This step aims to show the impact of the poisoning data on the detection accuracy, particularly the accuracy related to the malicious data traffic.

F. Poisoning Detection Using Isolation Forest

For the identification of such potentially toxic data, a technique called Anomaly Detection using Isolation Forest is used upon the toxic training data. As the name indicates, Isolation Forest is an unsupervised learning algorithm that isolates data points by randomly partitioning the feature space [27]. The anomalous data identified using this method are labeled as suspicious and potentially toxic.

In addition to improving the reliability of detection, another label consistency check is conducted based on the preliminary classifier. Based on the anomalous behavior as well as the inconsistency in the label with low prediction confidence, the data is labeled as poisoned.

G. Data Cleaning and Model Retraining

All the suspicious samples are eliminated from the poisoned dataset. The Logistic Regression classifier is then retrained using the cleaned dataset. Class imbalance methods are employed for betterrecall values for the attack classes. The classifier is then evaluated on the original test dataset and compared with the poisoned and baseline classifiers.

H. Performance Evaluation

The effectiveness of the proposed method of mitigation is checked based on the comparison of the performance parameters on three phases. They are the clean state baseline, the state after the poisoning attack occurs, and the state after the mitigation process has been performed. There is an emphasis on the improvements made in the parameters of the recall value and the F1 score. Both the metrics are affected the most by the attack on the intrusion detection systems.

IV. Results and Discussion

A. Baseline Performance on Clean Data

First, the Logistic Regression classifier was trained and tested on the original data to determine the baseline performance. Based on the performance metrics achieved from the experiment, the performance of the classifier on separating the benign traffic from the attack traffic is satisfactory enough. With an accuracy of 73.77%, precision of 72.37%, recall of 66.50%, and an F1 score of 69.31%, the baseline performance is established, on which the effectiveness of the data poisoning attack and the proposed defense mechanism would be measured.

B. Impact of Data Poisoning Attack

To test the attack due to poisoning, a controlled poisoning attack is modeled. The attack was done such that the attack samples were partially relabeled as benign samples, with noise added feature-wise. Further retraining using the dataset reveals the difference in model performance.

Although it showed some increase in the accuracy at 74.25%, it is rather deceptive. The precision increased to 78.50%, but the recall drastically went down to 58.10%, thus meaning it was unable to detect correctly many instances of attacks. The F1 score also fell back to 66.78%. It proves that data poisoning could harm the effectiveness in the security of a machine learning system even when it seems stable in terms of accuracy [13].

C. Detection and Removal of Poisoned Samples

To counter the poisoning attack, the anomaly-based detection method was incorporated with the use of Isolation Forest in addition to label consistency. Out of the 5,299 training examples, 1,142 were labeled as anomalies, representing 21.6% of the training set, thus being removed. The other 4,157 were labeled as clean training examples for rebuilding the classifier.

This is just a proof-of-concept showing that a substantial amount of data could end up being affected during poisoning attacks, thus the significance of having detection methods prior to model training.

D. Performance After Mitigation

After retraining the model on the cleaned dataset, there was considerable improvement in the various metrics. Accuracy increased to stabilize at 73.50%, while the precision reached 71.44%, denoting an enhanced correctness in attack classification, with the recall value rising to 67.49%, unlike 58.10% in the case of poisoning. Even the F1-score improved to 69.41%, almost at par with the baseline.

Although there was a small decrease in precision relative to the poisoned state, the gain in recall accuracy is very important for cybersecurity purposes, where correctly detecting hostilities is more important than gaining high precision to a certain extent. The above findings affirm that the proposed method can successfully restore the reliability of the machine learning algorithm [19].

E. Comparative Analysis

A comparative summary of model performance across all three stages—clean data, poisoned data, and mitigated data—is presented in Table II. The results clearly show that data poisoning disproportionately affects recall, while the proposed mitigation technique successfully recovers lost detection capability.

To further illustrate these findings, Fig. 3 presents a visual comparison of performance metrics across stages, emphasizing the recovery in recall and F1-score after mitigation.

TABLE II
COMPARATIVE PERFORMANCE METRICS ACROSS EXPERIMENTAL STAGES

Stage	Accuracy	Precision	Recall	F1-Score
Clean Baseline	73.77%	72.37%	66.50%	69.31%
Poisoned	74.25%	78.50%	58.10%	66.78%
Mitigated	73.50%	71.44%	67.49%	69.41%

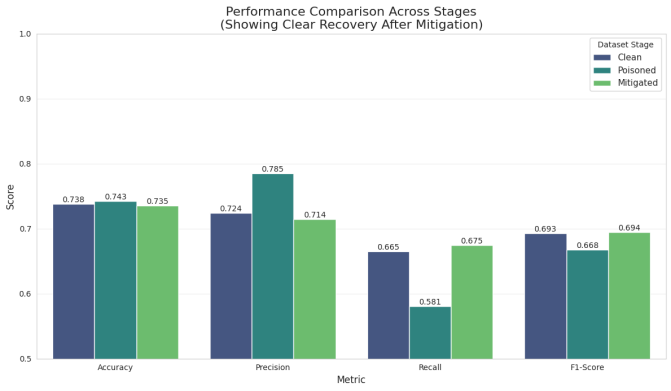

Fig. 3. Bar chart comparison of performance metrics (accuracy, precision, recall, F1-score) across clean, poisoned, and mitigated stages.

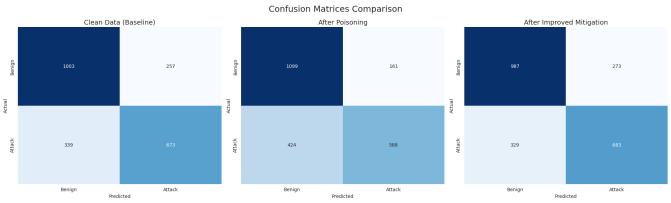

Fig. 4. Confusion matrices for the Logistic Regression model on clean, poisoned, and mitigated datasets.

Additionally, Fig. 4 shows confusion matrices for each stage, providing insight into misclassification patterns and highlighting the reduction in false negatives after data cleaning.

V. CONCLUSION

In this paper, an attempt has been made to assess how data poisoning attacks affect machine learning-based systems used in cybersecurity, as well as an anomaly-based remediation technique designed to counteract it. The experimental results have clearly shown how data poisoning can severely affect the reliability of machine learning models, especially regarding recall, even if there are no problems in terms of accuracy.

To address this problem, the unsupervised detection method involving the Isolation Forest algorithm along with the consistency of labels was used to remove any anomalous training dataset. The technique was successful in removing the poisoned training dataset, thus restoring the recall and F1 values. The results were satisfactory as it was close to the other results achieved through the clean dataset.

Results highlight the importance of protecting the training dataset as much as protecting the deployed systems, especially in the context of security applications involving ML. Being model agnostic and offline-friendly makes the technique useful and feasible to implement in cybersecurity systems.

VI. FUTURE WORK

Although the proposed methodology detects data poisoning and reduces its effect on an offline intrusion detection dataset effectively, the approach can further be extended on various fronts to make it more practical and robust. Future works can be done to evaluate the proposed approach against some concrete poisoning strategies, including backdoor attacks and adaptive adversarial poisonings, which explicitly aim to evade the detection based on anomaly mechanisms [41].

The current study utilizes a traditional machine learning classifier with tabular features. Future research may extend this framework to deep learning models and hybrid architectures to assess whether comparable mitigation effectiveness can be achieved in more complex learning environments. Similarly, experiments on large-scale and more diverse cybersecurity datasets could drive more in-depth analysis for the scalability issue.

Another area that offers great potential is the development of approaches for poisoning detection that can be done in real-time while the learning process takes place. Using explainable AI can be very helpful in understanding why certain data points are marked as malicious by deep learning models.

In essence, the proposed future outlooks for machine learning aim to not only ensure robustness against data-centric attacks but also facilitate the applications of trust-based cyber security solutions.

REFERENCES

[1] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart. 2016.

[2] Y. Xin *et al.*, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[3] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Security Privacy*, May 2010, pp. 305–316.

[4] M. Idhammad, K. Afdel, and M. Belhadi, "Distributed intrusion detection system based on ensemble learning for IoT," *IEEE Access*, vol. 9, pp. 146571–146585, 2021.

[5] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Security Privacy*, May 2017, pp. 39–57.

[6] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1467–1474.

[7] L. Huang *et al.*, "Adversarial attacks on neural network policies," 2017, arXiv:1702.02284.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[9] J. Steinhardt, P. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 3517–3529.

[10] A. Paudice, L. Muñoz-González, and E. C. Lupu, "Label sanitization against label flipping poisoning attacks," in *ECML PKDD Workshop Adversarial Learn.*, 2018.

[11] M. Jagielski *et al.*, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Security Privacy*, May 2018, pp. 19–35.

[12] M. D. DiBenedetto *et al.*, "Poisoning attacks and data sanitization mitigations for machine learning models in network intrusion detection systems," in *Proc. IEEE Canadian Conf. Elect. Comput. Eng.*, Sep. 2021, pp. 1–6.
[13] B. Biggio *et al.*, "Poisoning attacks on machine learning," *IEEE Secur. Privacy*, vol. 19, no. 4, pp. 34–43, Jul./Aug. 2021.
[14] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.
[15] X. Chen *et al.*, "Data poisoning attacks on machine learning models," *IEEE Access*, vol. 8, pp. 156693–156706, 2020.
[16] S. Wang *et al.*, "Data poisoning attacks and defenses in machine learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, 2022.
[17] T. A. Tang *et al.*, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. Int. Conf. Wireless Netw. Mobile Commun.*, Oct. 2016.
[18] Y. Mirsky *et al.*, "Data poisoning in machine learning for cybersecurity," *IEEE Secur. Privacy*, vol. 20, no. 3, pp. 58–67, May/Jun. 2022.
[19] J. Chen *et al.*, "De-Pois: An attack-agnostic defense against data poisoning attacks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3412–3425, 2021.
[20] A. Shafahi *et al.*, "Poison frogs! Targeted clean-label poisoning attacks on neural networks," in *Adv. Neural Inf. Process. Syst.*, 2018.
[21] G. Xia *et al.*, "Poisoning attacks in federated learning: A survey," *IEEE Access*, vol. 11, pp. 10708–10722, 2023.
[22] E. Bagdasaryan *et al.*, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020.
[23] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.
[24] D. Su *et al.*, "Is robustness the cost of accuracy? — A comprehensive study on the robustness of 18 deep image classification models," in *Proc. Eur. Conf. Comput. Vis.*, 2018.
[25] Q. Xiao *et al.*, "Data poisoning attacks on machine learning models," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2020.
[26] B. Nelson *et al.*, "Exploiting machine learning to subvert your spam filter," in *Proc. 1st Usenix Workshop Large-Scale Exploits Emergent Threats*, 2008.
[27] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.
[28] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2001, pp. 37–46.
[29] S. J. Stolfo *et al.*, "Cost-based modeling for fraud and intrusion detection," in *Proc. DARPA Inf. Survivability Conf. Expo.*, 2000.
[30] P. Laskov *et al.*, "Machine learning in adversarial environments," *Mach. Learn.*, vol. 81, no. 2, pp. 115–119, 2010.
[31] R. Perdisci, D. Ariu, and G. Giacinto, "Network intrusion detection with machine learning techniques," in *Encyclopedia of Biometrics*, 2009.
[32] K. A. P. Costa *et al.*, "Internet traffic classification using data poisoning," *IEEE Access*, vol. 9, pp. 10306–10318, 2021.
[33] A. Ferrag *et al.*, "Deep learning for cyber security intrusion detection," *IEEE Access*, vol. 8, pp. 164469–164492, 2020.
[34] Z. Chkirbene *et al.*, "A survey on adversarial attacks and defenses in machine learning-enabled cybersecurity," *IEEE Access*, vol. 10, pp. 102218–102247, 2022.
[35] X. Yuan *et al.*, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.
[36] N. Papernot *et al.*, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Security Privacy*, Mar. 2016.
[37] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018.
[38] M. Barreno *et al.*, "Can machine learning be secure?," in *Proc. ACM Symp. Inf. Comput. Commun. Security*, 2006.
[39] L. Muñoz-González *et al.*, "Poisoning attacks on deep learning," in *Proc. IEEE Symp. Security Privacy Workshops*, May 2017.
[40] P. W. Koh *et al.*, "Stronger data poisoning attacks break data sanitization defenses," *Mach. Learn.*, vol. 111, pp. 1–47, 2022.
[41] M. Goldblum *et al.*, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, 2020.
[42] T. Gu *et al.*, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," 2017, arXiv:1708.06733.
[43] V. Shejwalkar and A. Houmansadr, "Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *Proc. NDSS*, 2021.