Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Securing ML Models: A Systematic Survey of Poisoning Attacks and Defense Mechanisms

Mahdi Nsaif Jasim[a], Hanan Abed Alwally Abed Allah[b]

[a] *University of Information Technology and Communications, Department of Information Systems Management, Baghdad, Iraq.Email: mahdinsaif@uoitc.edu.iq*

[b]*University of Mustansiriyah, Department of Computer Science, Baghdad, Iraq.Email: hanan.cs88cs@uomustansiriyah.edu.iq.*

ARTICLE INFO

Article history:
Received: 29 /08/2024
Rrevised form: 10 /09/2024
Accepted : 1 /10/2024
Available online: 30 /12/2024

Keywords:

Poisoning Attack, Label Flipping, Clean Lebel, Watermarking, Adversarial Attack

ABSTRACT

In recent years, Machine Learning (ML) has brought about a significant revolution in several fields such as medicine, justice, cybersecurity, and other vital fields that require intelligent and urgent decision-making. With this development, a type of adversarial attack targeting ML models called a Poisoning Attack (PA) has emerged. One realistic attack scenario is for an adversary to subtly update samples or reverse some labels of training data, causing degradation to the model's overall accuracy during the testing phase. To gain a deeper understanding of this scenario, a survey will be conducted about the attack and how it is carried out against different models. In addition to the protection techniques to identify their weaknesses. Finally, some solutions will be proposed to maintain the availability, robustness, and integrity of ML models.

MSC..

https://doi.org/10.29304/jqcsm.2024.16.41776

1. Introduction

ML has recently gained popularity as one of the best techniques for solving classification and prediction problems. It is also widely utilized in the Internet of Things (IoT) environment, Recommendation Systems (RSs), Sensor Networks (SNS), Fraud Detection (FD), and Verification Code (VC) recognition [1]. The majority of ML models constantly require publicly available data. The data is typically sourced from untrustworthy sources. As a result, an attacker could take advantage of these weaknesses as part of what is known as a Poisoning Attack (PA), which affects the model's decision-making process and can lead to the faults during the testing phase. Smart City (SC) systems provide an obvious example: multiple sensors including cell-phones collect vast volumes of data. It is also anticipated that PA targeting SC systems could have disastrous effects; this kind of incident will likely happen due to the system's high reliance on public data [2][3]. Therefore, it is essential to encourage the development of advanced protection techniques (such as De-Pois, and kernel-based Support Vector Machine (K-SVM)) to maintain the robustness of ML models and thus mitigate the impact of poisoning [4]. The following is a list of this survey's primary contributions:

1. Various previous strategies of defense will be studied to identify their weakness.

∗Corresponding author: Hanan Abed Alwally Abed Allah

Email addresses: hanan.cs88cs@uomustansiriyah.edu.iq

Communicated by ‘sub etitor’

2. In this work, several solutions will be proposed to maintain ML models' availability, robustness, and integrity.

The paper is organized as follows: Types of ML algorithms are discussed in Section 2. Section 3 provides a comprehensive overview of the attack and its types. Security requirements, interest metrics, and attacker knowledge are presented in Sections 4, 5, and 6. While the previous works are provided in Section 7. Section 8 is the conclusion of the paper.

2. Classification of ML Algorithms

Generally, ML algorithms can be classified depending on learning style into supervised, unsupervised, semi-supervised or reinforcement learning as shown in Figure 1.

1. Supervised-Learning: During the training stage, every input is provided with the appropriate label, which could be a class label for a categorization or a continuous value for a regression model. Next, the model can be used to predict the label for a new input in the testing stage. It encompasses a variety of methods, such as logistic regression, SVM, and Naïve Bayes [5] [6].
2. Unsupervised-Learning: Refers to the process through which a technique trains with unlabeled data and generates predictions related to the dataset without supervision. Among the most popular unsupervised algorithms is clustering, which includes density-based, K-means, hierarchical, etc. [7] [8].
3. Semi-Supervised-Learning: This method works with data, some classified while others are unclassified. Even a tiny portion of labeled data can help the model train and produce an accurate outcome prediction [9].
4. Reinforcement-Learning: The process by which a model learns to adapt and improve itself through ongoing feedback from its output is called reinforcement learning. It functions as a reward, and the objective is to increase the positive benefits in order to raise performance. Among the most widespread reinforcement learning methods is quality-learning [10].

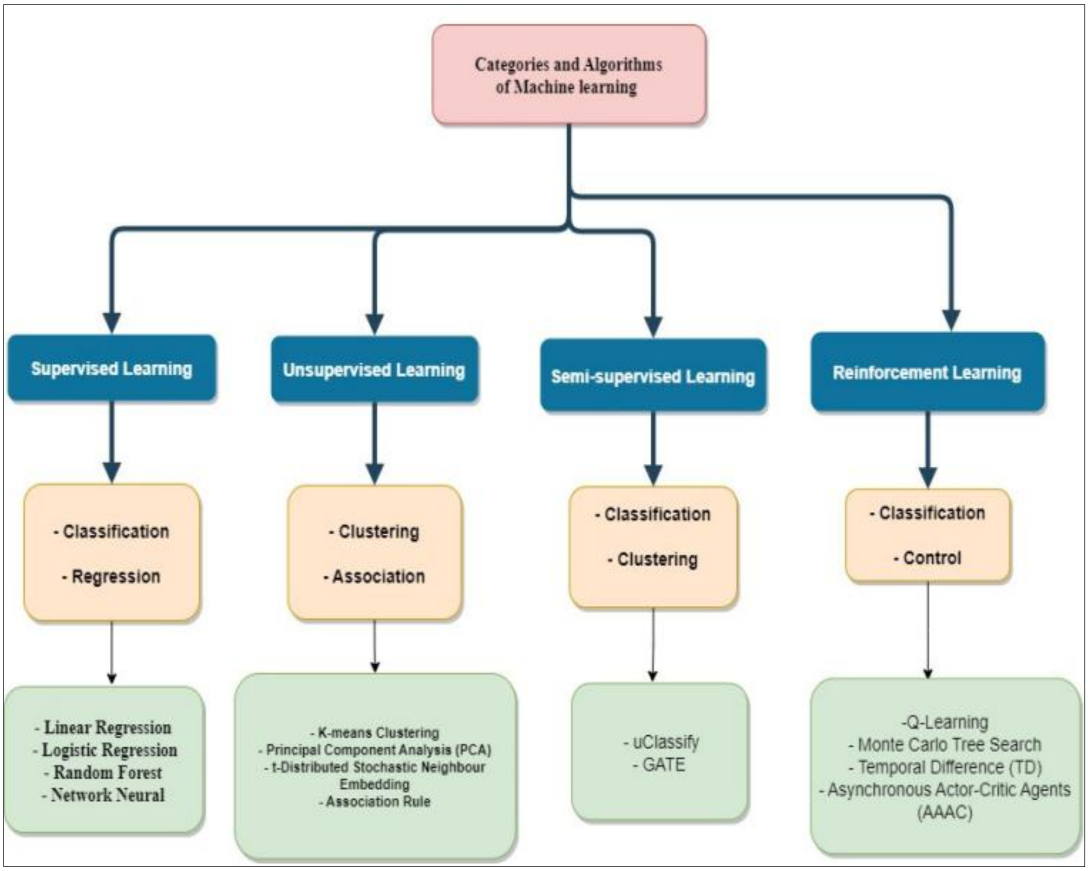

Figure 1. Categories of ML Algorithms [9].

3. Poisoning Attack (PA)

It also known as Data Poisoning Attack (DPA), is an attack used to modify the training data (by, for example, inverting labels, contaminating features, adjusting model configuration parameters, and changing model weights). There is an assumption that attackers can add to or manipulate the training data as shown in Figure 2. This affects the model's learning output, leading to a notable decline in accuracy [11]. There are several categories of PAs which are:

1. **Label Flipping (LF):** wherein a hacker modifies the labels of particular training samples to produce poisoned samples. For instance, a spam detection system uses a classified dataset of emails to gather information about spam email patterns. The method employs human annotators to ascertain the classification of emails as either spam or non-spam. However, a label flipping attack can compromise the system's performance because the model acquires erroneous connections between email characteristics and inaccurate spam/non-spam classifications. Hence, safeguarding against such attacks is of utmost importance [12].
2. **Watermarking:** in this attack, an attacker can merge and mask an intended sample onto the training set, introducing perturbations and producing poisoned instances instead of labels. In the context of DeepFake video detection systems, it is possible for an attacker to selectively focus on a subset of the training data and incorporate a subtle and invisible pattern or watermark inside a specific set of selected films. An example of this would be the attacker inserting a minute and imperceptible pattern into the entirety of the "authentic" movies within the training dataset. This watermark has the potential to be intentionally crafted to induce the model to incorrectly identify any films that exhibit the identical pattern as "authentic," despite the fact that the video is, in fact, a DeepFake [13].
3. **Clean Label (CL):** This occurs when a hacker produces poisoned samples that contain adversarial perturbations that are detrimental to ML models but "invisible" to human specialists. A sentiment analysis system is developed for an e-commerce platform to detect and categorize textual reviews provided by visitors as either "positive" or "negative". The assailant has the ability to selectively focus on a portion of the learning data and discreetly alter the reviews in order to modify their sentiment. As an illustration, the assailant may take an authentic "positive" review and make slight alterations, such as altering some words or phrases, to transform it into a "negative" review, while safeguarding the authenticity and credibility of the reviews [14].

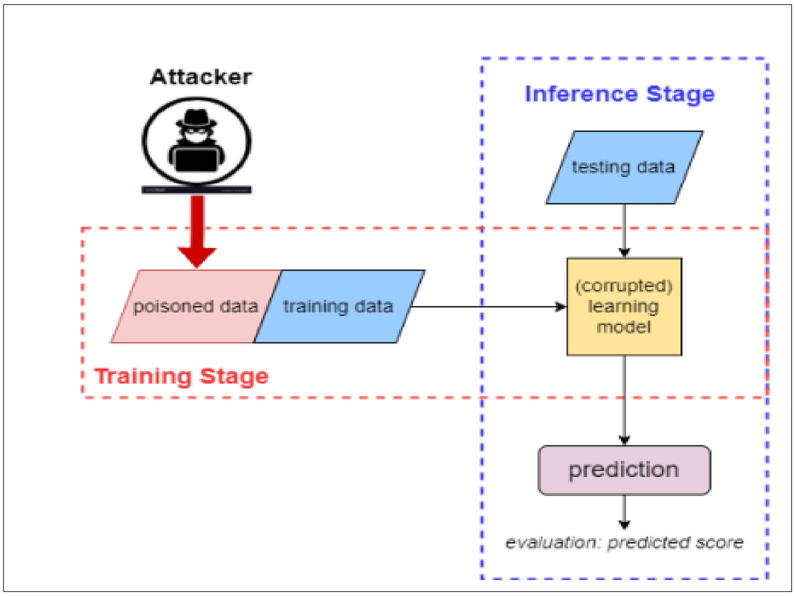

Figure 2. PAs Against ML Models [5].

4. Requirement of Security

The Requirements of Security in ML Models includes:

1. Integrity is the capacity of a model to behave understandably and predictably according to predetermined criteria. In PA, a hacker may alter the model's parameters during training, compromising the model's overall integrity. For example, the quantity of label- flipped or the ratio of the attack-affected models' parameters to the target model demonstrates [15] [16].
2. An ML model's availability is related to its capacity to function as predicted in the face of severe disruptions. One obvious way to measure availability is to analyze the decision boundary of the machine learning model; this is also represented in the accuracy metric. For example, the decision boundary completely collapses when a DP attack injects poisoned data points into the training sample [17].
3. The ability of the model to continue operations in a desired manner despite disturbances in the input distributions is known as robustness. Such perturbations could be purposefully created by a criminal, in which case training a model with poisoning data would significantly compromise the robustness of the model. Metrics such as the Area Under the Curve (AUC) and Receiver Operating Characteristics (ROC) curve can be applied [15].

5. Metrics of Interest

The degree of performance deterioration exhibited by the model being attacked through the testing stage is used to determine the impact of DP attacks. The total misclassification percentages in each class, which show True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), can be used to confirm this further. Furthermore, the attack's efficacy is demonstrated by a notable decline in total accuracy, often known as an accuracy deterioration. ROC, accuracy, and recall are a few of the artificial intelligence criteria that can also be employed [18] [19].

6. Knowledge of Attacker

The attacker's understanding of the five elements—feature space, classifier type, classifier learning method, classifier learning hyperparameters, and training dataset—necessitates several assumptions [20] as shown in Figure 3, which are:

1. A scenario known as a "white box" occurs when the attacker possesses complete knowledge of all five earlier-mentioned elements and any defense mechanisms that have been installed on top of the model [21] [22]
2. When the target model is unknown, this is known as a "black-box" situation. It is crucial to note that the attacker gains the advantage over any defender simply by having access to the training data [23].
3. The term "gray-box " refers to a situation in which the attacker is aware of the five criteria beforehand but is unaware of the protection mechanism. This is considered to be a compromise between white-box and black-box scenarios. Typically, it is employed to assess the defense against hostile assaults [24].

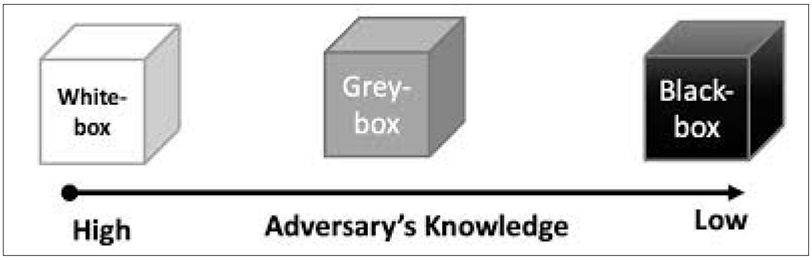

Figure 3. Knowledge of Attacker [20].

7. Poisoning -Attacks and Defense Mechanisms

Several contributions have been made to solve the problem of PAs; some of them are:

Mehran Mozaffari-Kerman et al. [25] created malicious instances from the original dataset to offer a general attack technique for ML algorithms and medical datasets. The assault tilts the model in favor of the class that is assaulting. Five datasets have been used to assess six ML algorithms under PAs: thyroid disease (TD), breast cancer (BC), acute inflammations (AI), echocardiograms, and molecular biology (MB). The algorithms include the Best-First (BF) decision tree, Ripple-Down (RD) rule learner, Naive Bayes (NB) decision tree, Nearest-Neighbor (NN) classifier, Multilayer Perceptron (MLP), and Sequential Support-vector minimal optimization (SMO). Halfway through the assault and at the very end, when all malicious instances have been added, with 15% and 30% of the initial dataset, the algorithms are evaluated. According to the findings, the most reliable technique is SMO as shown in Table 1. Additionally, the study shows how well the suggested countermeasure against poisoning assaults utilizing the Kappa statistic and correctly identified occurrences (CCI) works. As part of the countermeasure, a model is built using the training dataset, its accuracy is assessed using the validation dataset, and an alarm is raised if any questionable changes occur, such as the accuracy threshold the user selects being less than the ideal number.

TABLE 1. Effectiveness of Proposed Assault on ML Techniques.

Assault	Dataset kind	Misclassification Rate (from most to least susceptible)					
15% added	Thyroid Disease	NN (9%)	BFTree (7%)	RD (5%)	NBTree (4%)	MLP (3%)	SMO (3%)
30% added		MLP (20%)	RD (18%)	BFTree (18%)	NN (16%)	NBTree (13%)	SMO (12%)
15% added	Acute Inflammations	RD (9%)	BFTree (9%)	NN (8%)	NBTree (8%)	MLP (6%)	SMO (6%)
30% added		RD (21%)	BFTree (18%)	NN (18%)	MLP (14%)	NBTree (12%)	SMO (12%)
15% added	Molecular Biology	NN (9%)	BFTree (9%)	RD (7%)	NBTree (6%)	MLP (6%)	SMO (5%)
30% added		BFTree (18%)	NN (17%)	MLP (15%)	NBTree (15%)	RD (12%)	SMO (12%)
15% added	Breast Cancer	MLP (14%)	NN (11%)	BFTree (8%)	RD (4%)	NBTree (3%)	SMO (3%)
30% added		MLP (26%)	BFTree (23%)	RD (22%)	NN (18%)	NBTree (16%)	SMO (16%)
15% added	Echocardiograms	RD (8%)	NBTree (8%)	NN (7%)	MLP (6%)	BFTree (3%)	SMO (3%)
30% added		NBTree (20%)	NN (18%)	MLP (16%)	RD (16%)	BFTree (11%)	SMO (11%)

Huang Xiao et al. [26] examined how embedded feature selection methods like LASSO, RR, and EN can be hacked. Such algorithms use a dataset of maleware and 5951 harmless examples gathered from online content. The attacker can increase the classification error of algorithms by using one assault point at a time, iteratively changing and updating the current sample at each stage. The findings demonstrate that attackers can effectively create poisoning attacks using only surrogate data, even without access to training data. All techniques show a tiny classification error and dependable performance without an attack. For LASSO, the classification error increases almost tenfold when up to 20% (from 2% to 20%) of the training data is poisoned. This effect is marginally less pronounced for the elastic net and ridge, showing marginally better robustness against this danger.

Ricky Laishram and Vir Virander Phoha [27] utilized poisoning strategy with gradient ascent against SVM classifiers to increase the FP rate of the classifier. To do this, meticulously created data points must be inserted into the training set. Additionally, the Curie method suggested to protect the system-utilized SVM classifier. Curie is a filter before the buffered data is used to retrain the SVM. It exploits the fact that the poison data is a regular point in the feature space and has reversed labels. Once the data has been clustered in the feature space, the class label is used as a feature with a suitable weight when determining the average distance between each point in the same cluster. After that, the data points are removed from the training set with confidence levels less than 95%. The Curie approach has been evaluated on the MNSIT dataset. Table 2 displays FPR after varying percentages of PAs have been injected, as well as the accuracy of the SVM classifier with and without Curie.

TABLE 2. Accuracy and FPR of SVM.

Poisoning Points	Without Curie		With Curie	
	Accuracy (%)	FPR	Accuracy (%)	FPR
0	0.992	0.017	0.990	0.019
25	0.958	0.085	0.991	0.018
50	0.957	0.086	0.991	0.017
75	0.934	0.128	0.991	0.017
100	0.905	0.154	0.989	0.022
125	0.851	0.221	0.990	0.019

Bo Li et al. [28] using the MovieLens dataset—which has 20 million ratings and 465,000 tag applications applied to 27,000 films by 138,000 users—the study assesses the efficacy of a suggested poisoning assault method. The average rating for individual items and the Mean Square Error (RMSE) for anticipated unseen entries are the two measures used to compare the systems' performance before and after PA assaults. The distributions of rated items between malicious and legitimate users are compared using the paired t-test. Compared to other strategies, the projected gradient ascent (PGA) strategy produces the highest RMSE score. However, malicious users must rate every item evenly and randomly, which could reveal the negative profiles to a knowledgeable defence. Despite producing a slightly inferior attacker utility, the Stochastic Gradient Langevin Dynamics (SGLD) approach creates rogue users that are challenging to distinguish from legitimate users. Compared to uniform attacks, PGA and SGLD increase the attacker's utility. Ultimately, as a defensive tactic, analysis of hostile behavior has been used.

Patrick P. K. Chan et al. [29] presented a method for data sanitization based on data complexity, a measure of categorization difficulty. It uses measurements of class separability, geometry, topology, the density of manifolds, and the overlap of individual feature values to categorize the data. Class separability, linear separability, and class border complexity measures help determine the maximum feature efficiency of a linear classifier used to organize the data. The method distinguishes between valid and tainted samples using a label flip attack, which alters sample distribution and raises classification errors. The data complexity metrics accurately reflect the complexity of the classification problem. The approach uses Reject on Negative Impact (RONI) and SVM to measure performance in applications like letter recognition and spam filtering. The spam dataset from the UCI machine learning repository is used in a test, along with linear SVMs trained on tainted training sets. The suggested method is, on average, 3.76% more accurate than RONI when applied to linear SVMs under the furthest, far-rotate, and maxErr attacks. The performance of the suggested method could be better for those using no protection when there is RONI. The letter recognition dataset from the Statlog collection consists of 20,000 samples, 26 classes, and 16 attributes also used. Both linear and nonlinear SVMs achieve 97.73 and 99.56% accuracy without assaults. The recommended method offers superiorly accurate predictions with statistical significance at 95% compared to RONI and sanitized SVMs.

Nathalie Baracaldo et al. [30] developed a provenance-based technique for supervised ML to identify and remove dangerous instances from data that can be considered Entirely untrustworthy (EU) or Partially trustworthy (PT). The PT dataset approach uses supervised ML techniques (SVM and Logistic Regression (LR)) with provenance characteristics. Each portion of the untrusted data is examined for poisoning before being divided to share an identical provenance signature. The machine learning algorithm tests classifier performance with and without the segment as part of its training process. If a model performs better without the segment, it is permanently removed from both sets. This approach prevents poisonous points from evading detection and reduces training times by requiring only a tiny fraction of RONI's training times for each data segment. A scenario including the Internet of Things and numerous contributing devices was used to assess the RONI defense. The outcomes were used as a baseline, with the most precise score being the calibrated RONI. Biggio's method was applied in tests of the MNIST dataset. For poison EU, the baseline, which calls for at least 120 data points, was the better. Also, a synthetic dataset with 280 data points per signature to test the provenance defense has been used. The poor performance of the baseline defense suggests computing data points per signature to avoid false positives.

Sen Chen et al. [31] presented an adversarial crafting method that uses syntactic traits to build crafted poisoned samples that closely resemble real-world attacks. The three machine-learning detection systems- DREBIN, MAMADROID, and DROIDAPIMINER—can be tricked by our poisoning approach. for solving the problem, KUAFUDET was created. It has two parts: an online detection phase that uses the classifiers (SVM, RF, and K-Nearest Neighbor, KNN) that were trained in the first phase; and an offline training phase that picks features from the training set and extracts them. An automatic camouflage detector is implemented to filter suspicious false negatives and feed them back into the training phase, further mitigating the adversarial environment. This self-adaptive learning system connects these two phases. Based on tests, the KUAFUDET has proven its ability to improve detection accuracy by at least 15% and dramatically reduce FP or malicious apps that are poisoned. Research on over 250,000 mobile applications shows that KUAFUDET can obtain a detection rate of up to 96%.

Matthew Jagielski et al. [32] suggested a new poisoning regression approach known as the optimization-based Poisoning attack (OptP), which outperforms the Baseline Gradient Descent (BGD) assault by a factor of 6.83. To attain the mean square error (MSE), the optimal OptP attack chooses the optimization objective, optimization argument, and initialization approach. Also, a quick statistical assault (StatP) has been developed. TRIM as a novel protection technique that provides excellent robustness and resilience against poisoning attempts, was also available. Regression parameters are iteratively estimated, and residual points are eliminated using a trimmed loss function. The novel attacks and defenses have been evaluted in four linear regression models (OLS, LASSO, Ridge, and EN) and three datasets (healthcare, loan assessment, and real estate domains). Based on the tests, the presented attack demonstrated a notable improvement above baseline attacks and increased MSEs of 6.83 and 155.7 compared to un-poisoned models. At the same time, the TRIM showed an advance in Huber, RANSAC, and RONI by (71.13%), (75%) and (95.45%) respectively.

Bo Lee et al. [33] Created an algorithm that selects important examples and fluctuates them using the greedy method to affect the target function. A protection technique was used to apply labels to each case in the training kit using K-NN to enforce the standardization of labels via similar examples, especially in areas far from decision limits. The technique for each sample in the contaminated k-NNs training group is that the corresponding training sample is renamed if the percentage of data points with the most widespread label exceeds the threshold. Three sets of actual data from the UCI repository were also used: breast cancer, MNIST, and Spambase. The average classification error was increased by a factor of 2.8, 6.0, and 4.5 for breast cancer, MNIST, and spambase, indicating the effectiveness of the label reversal attack. At the same time, the defensive approach reduced error rates to 0.06, 0.03, and 0.02 for the three datasets.

Di Cao et al. [34] examined the link between the number of attackers, tainted samples (LF), and attack success rate using a federated learning system with the federated averaging (FAv.) technique. Several participants train a local model using a local parameter vector and send it to a central parameter server in a federated learning setup. Then, FAv. has been used to compare the local model or vector with a global model. Each participant uses the Stochastic Gradient Descent (SGD) algorithm to create a new local model after receiving several poisoned Samples (S). Multiple attackers who use the same loss function and hyper-parameters as honest participants assault a federated learning system. The effectiveness of distributed poisoning assaults is assessed using the MNIST dataset, and a Sniper technique to eliminate poisoned local models by solving a maximum clique problem is proposed. 50,000 samples are included in the training dataset, and 10,000 samples are for testing. The training data are distributed equally among ten participants in a federated learning system. Convolutional Neural Network (CNN) which has one complete connection layer and four convolution layers is also chosen. The global model's accuracy was assessed and the results showed 100%. However, combining local models resulted in an accuracy of 90%, showing that a more precise global model is possible.

Rahim Taheri et al. [35] Designed an attack to target Android malware detection systems named Silhouette Clustering-based Label Flipping (SCLF). This method relies on K-means clustering to divide the training samples into two clusters and compute silhouette values for the samples along with their predicted labels. In this context, a silhouette value close to 1 indicates that a sample fits well within its respective cluster, whereas values relative to -1 suggest misclassification. In the SCLF attack, the labels of samples are flipped if the silhouette value is less than zero. Additionally, two defense techniques, known as label-based semi-supervised (LSD) and clustering-based semi-supervised (CSD), have been introduced. In the LSD approach, a semi-supervised strategy based on label estimation and the Label Spreading (LS) technique is employed to determine the labels. LS is trained using validation data to create a model. Label Propagation (LP) is also applied for label prediction. Subsequently, a Convolutional Neural Network (CNN) is used as the third component of an ensemble learning approach, trained with validation data. The LSD method's final step involves voting among the results of these three methods and the poisoned label, with the resulting label becoming the label for the training samples.

Conversely, in the CSD approach, a proposed CNN model with validation data and four cluster metrics—the Rand Index (RI), Mutual Information (MI), Homogeneity Metric (HM), and Fowlkes-Mallows Index (FMI)—is used. This approach involves adding one training sample to the validation dataset, calculating clustering values with the four metrics, and comparing them to the base values. If the difference is less than 0.1 (considered a threshold), the sample is labeled adequately; otherwise, it is deemed mislabeled. Both defense algorithms were evaluated using three datasets: Drebin, Genome, and Contagio. The average accuracy for CSD was approximately 95%, 97.6%, and 98.5%. In LSD, it was about 80%, 79%, and 77% for all datasets, respectively.

Patrick P. K. Chan et al. [36] suggested a Causal Detection Technique (CAD) focusing on data complexity measurements to find polluted datasets. Geometric characteristics and a two-step secure classification paradigm are used to decrease causal attacks, improve generalization potential, and increase learning resistance. Practical data analysis is ensured using the UCI and the KEEL-dataset repository to identify causative attacks. Five security-related datasets including Steghide, MB1, F5, PDF, and Spam, were used in the suggested methodology. In addition to a standard SVM, a robust SVM with kernel matrix correction has also been used. The attack strength significantly impacts how accurately perfect, partial, and no-knowledge scenarios may be detected, such as the detection accuracies are higher than 82%, 80%, and 75% for the different scenarios, respectively, even though the attack strength is only 5%. The detection accuracy remains constant at 84%, 82%, and 80% when the attack strength is 15% to 20%.

Liu et al. [37] introduced an innovative attack called the Poisonous Label (PL) attack. The primary goal of this attack is to diminish the testing accuracy of Convolutional Neural Networks (CNNs) when exposed to PL images in a black-box scenario. The attack comprises two crucial stages: First, to create PL images, introducing counterfeit images that align with the target class label is imperative. To accomplish this, they employed Enhanced Conditional (EC)-DCGAN, a model based on Generative Adversarial Networks (GANs). Second, symmetric and asymmetric poisoning vectors were applied to produce PL images. MNIST, in addition to Fashion MNIST, was used in a series of studies. Table 3 presents the results of these tests.

TABLE 3. Percentage of Mistakes Made by Inserting Poisons.

Dataset	Attack Technique	Poison Percentage = 0.1			Poison Percentage = 0.2			Poison Percentage = 0.3		
		Max	Min	Mean	Max	Min	Mean	Max	Min	Mean
FashionMNIST	Symmetric Poisonous	0.784	0.273	0.553	0.822	0.266	0.583	0.864	0.238	0.602
FashionMNIST	Asymmetric Poisonous	0.997	0.851	0.948	0.999	0.951	0.982	1.000	0.971	0.990
MNIST	Symmetric Poisonous	0.989	0.679	0.913	0.999	0.677	0.946	1.000	0.661	0.954
MNIST	Asymmetric Poisonous	1.000	0.981	0.997	1.000	0.989	0.998	1.000	0.993	0.999

Hongpo Zhang et al. [38] presented two non-traditional LF attacks: LF depending on the Entropy (LF-E) approach and LF depending on k-medoids (LF-K medoids). The weight (Wj) of each attribute in the training set was determined using E in the LF-E. The scores (si) associated with each malicious example (xi) were then calculated using these weights (Wj). Then, all negative examples were rearranged according to their scores (si), and the labels of the instances with lower si values were reversed. K-medoids were utilized to calculate the central value of both kinds of examples, and the fraudulent instances were rearranged according to their distance from the center of the standard samples. The test set's classification error was maximized by altering the labels of the negative instances that were closest to the center of the normal one. This study evaluated the resilience of the Naive Bayes (NB) classifier on three datasets (Spambase, TREC 2006c, and TREC 2007 in the spam domain) under varying noise levels. The accuracy of the NB is shown in Table 4.

TABLE 4. Accuracy (%) Of NB.

Noise (%)	Attack_Method	Spambase	TREC 2006c	TREC 2007
0	-	90.55	93.13	89.93
5	Random	89.9	93.07	89.87
	E_method	89.58	89.8	88.13
	k - medoids	89.47	92.93	87.2
10	Random	89.25	91.87	88.8
	E_method	87.73	87.4	84.73
	k - medoids	88.49	87.53	78.07
15	random	88.06	91.53	87.6
	E_ method	85.56	86.13	80.13
	k - medoids	86.43	85.53	74.93
20	Random	86.75	90.6	85.73
	E_method	83.93	84.53	74.93
	k - medoids	85.02	83.4	71.4

Bingyin Zhao and Yingjie Lao [39] suggested two methods of class-oriented poisoning attacks, extending adversarial objectives to a per-class basis. The first method (class-oriented error-generic (COEG)) DNN model creates poisoned data focusing on the supplanter class. This method preserves feature information related to the ground-truth class while reducing other classes' features, reducing model accuracy and increasing the Complement Fixation Test (CFT) rate, outperforming baseline attacks (FL). Despite the bias introduced by training with a single data point, the technique achieves a CFT rate of 85.60% through 20 iterations. In the second method (class-oriented error-specific (COES)), a gradient-based approach is used to preserve non-victim performance while reducing the precision of the targeted victim class. The COES attack targets one class, affecting the performance of other categories. The attack can raise the victim class's CFT rate to 51.14%, outperforming baseline assaults. The approach for ImageNet achieves a CFT rate of 62% for the victim class while keeping non-victim classes' CFT rates low. To assess the effect of a poisoning attack on CIFAR-10, DNN, and CNN, 1000 training images and 9,000 testing images were employed in the study. With a CFT rate and test error above 50%, the suggested strategy outperformed the FL assault. A possible countermeasure is periodic checking of the accuracy and loss of learning models, such as averaged stochastic gradient classifiers and combinatorial models like bagging.

Richa Sharma et al. [40] presented a new LF assault; optimal training set instances are identified using sequential clustering (SHC); then only 20% are flipped, reducing validation loss. Also, an innovative SCOAP feature-based strategy has been used to identify poisoned samples in the Categorical Boosting (CatBoost) model and through the use of k-NN relabel them to their respective classes to prevent Hardware Trojan (HT) networks (used as an example) from wrongly categorizing throughout prediction. Based on a Trust-Hub benchmark test, the proposed attack lowers average accuracy by 67%. In comparison, the defense strategy improves CatBoost model prediction performance by 32.57%, resulting in an accuracy of 99.66%, as indicated in Table 5.

TABLE 5. Impact of LF on Trust-Hub Benchmarks.

	Metrics (Without Defense %)						
	Loss	Accuracy	Recall	ROC-AUC score	TNR	FPR	FNR
LF	58.5	67.09	50.53	58.7	68.72	32.57	46.16
	Metrics (With Defense %)						
	Loss	Accuracy	Recall	ROC-AUC score	TNR	FPR	FNR
	1.43	99.66	89.45	92.2	99.82	0.1725	10.53688

Qingru Li et al. [41] introduced LF as a unique attack that divides instances of a data set— Agglomerative Hierarchical Clustering (AHC) with the drebin and genome—into normal and aberrant clusters. Subsequently, the Silhouette Clustering (SC) value is computed based on the clustering findings to identify the samples that are most likely to be contaminated. It is then used to damage a dataset that can be used to train different classifiers including the Logistic Regression (LR), AdaBoost, Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), and Multilayer Perceptron (MLP). Finally, label flipping is meant to be prevented using a defense method called TrAdaBoost. Weights for the training set are modified in this way: if an instance from the contaminated dataset is mislabeled, its weight is decreased; otherwise, it is increased.

In contrast, the weight of a sample from the uncontaminated category is increased when it is incorrectly classified and decreased when it is correctly classified. The classification model is then retrained using the modified training set. The classifiers were tested at a 20% rate of LF, and the outcomes are found in Table 6.

TABLE 6. Accuracy of Classifiers Under 20% Percentage of LF.

Attack Method	**Accuracy Without Defense (%)**				
LF	RF (71.7)	SVM (69.5)	DT (71.5)	LR (69)	MLP (72)
	Accuracy With Defense (%)				
	RF (89.6)	SVM (90)	DT (88.5)	LR (89.2)	MLP (89.5)

Hesamodin Mohammadian et al. [42] executed an innovative LF attack against network-based intrusion detection systems (NIDS). The study uses DNN with two datasets: CIC-IDS2017 and CSECIC-IDS2018. For the network's attack classification, it first employs MLP with 256 neurons and the RelU activation function. After that, the assault (LF) scenario starts with an evasion strategy that uses examples from the original dataset to produce adversarial instances, and then it trains a DNN on tainted datasets. According to the experimental findings, a deep learning model performs noticeably worse as the percentage of flipped labels from malicious to benign rises. When 70% of the labels (CIC-IDS2017) are reversed, the average accuracy falls from 98.06% without flipped labels to 29.25%. A similar outcome was found with the CSE-CIC-IDS2018 dataset, with the average accuracy falling from 97.36% with no flipped labels to 29.2%. The results of comparisons are presented in Table 7.

TABLE 7. Results of ML Methods

ML Methods	**CIC-IDS2017**			**CSE-CIC-IDS2018**		
	F1-score	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**
DT	99.84	99.76	99.92	97.70	99.73	96.14
Naïve Bayes	28.47	32.43	73.75	48.29	49.48	72.79
LR	36.71	39.76	34.96	57.79	64.13	56.84
RF	96.58	99.79	94.24	94.23	99.81	90.97
DNN	97.99	92.98	88.09	97.97	97.51	97.46

8. Discussion

In Table 8, a comparison of previous studies has been made based on the proposed attack method, targeted ML models, and the defense methods used in addition to the strength and Limitations and their implications for real-world applications.

TABLE 8. A Comparison of Earlier Studies for PAs.

Reference and Proposed attack	**Method of Defense**	**Target ML model**	**Strength**	**Limitations and their implications for real-world applications**
algorithm-	Monitor and	NN,BFTree,	Early Detection:	1) Sensitivity to noise: the

independent attack [25]	identify variations in metrics (CCI and Kappa) related to accuracy.	RD,NBTree, MLP, SMO	Observation and identification of fluctuations in these measures can facilitate early detection of PA, thereby mitigating their risk of significantly compromising the model's performance.	protection technology depends on tracking and detecting deviations in accuracy metrics, the reversal of data and noise will have an effect on detecting process. 2) In the context of medical applications aimed at detecting early indications of diseases or tracking the advancement of medical disorders, establishing thresholds for CCI and Kappa to detect PA in patient data, such as vital signs and diagnostic testing results, might pose challenges due to the presence of noise in biological data.
CL [26]	_	LASSO, EN, and RR	1) L1 regularization employed by LASSO enhances its robustness against outliers in the training data, a critical factor in PA detection models. 2) The use of L1 and L2 regularization in EN can increase generalization, thereby providing advantages against poisoning attempts that seek to diminish the model's performance on uncontaminated data. 3) RR is demonstrated to be efficacious in addressing the issue of multi-collinearity within datasets, a prevalent attribute observed in poisoned datasets.	1) LASSO may exhibit reduced efficacy when challenged by advanced PAs, such as CL and watermarking, which can bypass feature selection procedure. 2) The efficacy of EN relies heavily on the appropriate choice of two hyper-parameters (α and λ). As a result, these parameters can directly influence EN's robustness against poisoning assaults. 3) The linear characteristics of Ridge Regression may impose constraints on its capacity to efficiently address poisoning assaults that specifically target unstructured data, such as text, photos, or audio components. 4) Within the context of Intrusion Detection System (IDS) applications, the limitations of these methodologies can potentially result in higher rates of false positives or false negatives, thereby diminishing the system's resilience and reliability when faced with poisoning

				attack scenarios
LF [27]	Curie	SVM	1) The Curie algorithm is specifically developed to detect anomalies or outliers within the dataset that exhibit deviations from the established regular patterns. it is useful in identifying potential poisoning instances that add data points that deviate from the anticipated distribution. 2) Due to its lack of reliance on labeled training data, this has the potential to enhance its resilience against specific forms of PAs, such as LF.	Curie's dependence on the identification of anomalies within the data may occasionally result in an increased frequency of false positive detections. Unwarranted alarms or disruptions may arise, thereby diminishing the overall efficacy of systems such as Intrusion Detection Systems (IDS).
LF [28]	analysis of hostile behavior	PGA and SGLD	By prioritizing the examination of belligerent conduct, the defensive mechanism can enhance its ability to detect aberrations from typical user or system interactions that could potentially signify an ongoing PA	1) Highly skilled assailants' ability to modify their poisoning tactics in order to avoid detection through hostile behavior analysis is particularly evident when they have a comprehensive understanding of the defense system. 2) Financial losses stemming from successful fraudulent activities or supply chain disruptions
LF [29]	Data sanitization based on data complexity	SVM	It entails identifying, removing, or neutralizing complex or aberrant data. This practice serves to uphold the integrity and resilience of the underlying ML models, hence enhancing their ability to respond effectively to poisoning attacks.	1) The process of accurately measuring the complexity of data points and establishing suitable thresholds for sanitization can present significant challenges, particularly in sectors characterized by numerous and varied sources of data. 2) In the context of fraud detection systems, the sanitization techniques encounter challenges when dealing with big money transfers that are undertaken as part of

				legitimate commercial activity. The probable consequence of this action is the premature elimination of any such data, resulting in their identification as potential instances of fraudulent activity.
Injected CL [30]	Provenance-based method	LR and SVM	The method is highly flexible as it can be combined with additional protection strategies such as anomaly detection or extreme value analysis to efficiently reduce false alarms and accurately detect a poisoning attack.	Poisoning of authentic data sources occurs through internal gathering procedures that appear to have a genuine origin. Therefore, the presence of poison data has the potential to circumvent provenance-based identification systems. Healthcare apps utilizing medical equipment data can be exploited by malicious actors to gain unauthorized access and introduce poisoning attacks.
CL [31]	KuafuDet	RR, KNN and SVM	KuafuDet's adaptive learning skills allow it to continually improve its repository of data and models, making the system more resistant to poisoning attempts.	The integration of additional samples into the training procedure, which involves the potential exclusion of FN, can pose difficulties in ensuring the accuracy and origin of the data utilized for system updates. This absence can make the system vulnerable to deliberate PA such as Cyber Secure Solutions.
OptP, and StatP [32]	TRIM	OLS, LASSO, ridge, and EN	The TRIM model's inherent adaptability allows it to effectively identify and adjust to changes in the data distribution resulting from poisoning experiments.	TRIM needs more processing power than conventional regression techniques. This additional processing complexity can be a drawback, particularly when resources are limited or real-time is of the essence. The inclusion of TRIM in a real-time fraud detection system introduces additional complexity that may impede the system's ability to promptly make choices, thereby enabling poisoning assaults to evade detection.
LF [33]	KNN	Greedy	Reconfiguring KNN model by including fresh data or retraining it to accommodate advanced poisoning threats can be simpler in comparison to	1) KNN, when used in imbalanced datasets, can bias forecasts towards the majority class; PAs can take advantage of this bias, leading to misclassifications or biased forecasts.

			more intricate machine learning models. The model effectively responds to mitigation strategies of PAs.	2) The KNN algorithm is susceptible to the curse of dimensionality. In feature spaces with a large number of dimensions, the informativeness of distances between data points decreases, reducing KNN's efficacy in PA detection. 3) In the context of a network-based IDS, the presence of high-dimensional network traffic data can potentially result in misclassification of false positive information or the failure to identify specific forms of false negatives generated by inserted poisoned data. So, the system's reliability and trustworthiness may suffer.
LF [34]	Sniper Method	CNN	The implementation of a proactive strategy can effectively identify Distributed PAs prior to their potential to seriously compromise the global model, hence enhancing the overall security of the federated learning system.	The Sniper method monitors client behavior and updates it to detect DPAs in a federated learning system, which introduces additional computational overhead. Also, variations in dataset distribution, client characteristics, and learning settings can impact performance. Autonomous car manufacturers may exhibit divergent priorities, safety standards, and update strategies, thereby introducing variances in the federated learning process and the operational dynamics of the Sniper approach. Consequently, these discrepancies can manifest in the detection of PAs.
SCLF [35]	LSD and CSD	CNN	The LSD algorithm demonstrates superior performance in leveraging the limited quantity of labeled data present in the learning system. Consequently, LSD exhibits enhanced accuracy in detecting poisoning attacks when compared to unsupervised approaches	The effectiveness of LSD is greatly influenced by the accessibility and proficiency of the labeled data, posing a considerable obstacle in the identification of label flipping and clean label PAs. The efficacy of CSD is intrinsically linked to the quality of the clustering algorithm used. Inadequate

			that simply rely on unlabeled data. CSD has the capability to identify patterns and anomalies within client data, which could potentially serve as indicators of poisoning attacks.	clustering might result in decreased accuracy in detecting poisoning and an increased occurrence of false positives, since the model may have difficulties in differentiating between harmless and harmful client actions as in fraud detection application.
Causative LF [36]	CAD	SVM, and ensemble of decision trees	By prioritizing causal linkages above mere correlations, the CAD method can enhance the resilience of the defense against data distribution changes resulting from PAs.	CAD models can generate false detections, which can lead to system downtime or undetected PAs. This is particularly true in mission-critical applications such as autonomous vehicles or medical systems, where the repercussions of false detections can be significant, potentially resulting in system failures.
Symmetric and Asymmetric Poisonous [37]	-	CNN	CNNs are intrinsically engineered to exhibit resilience against spatial fluctuations in the input data, such as shifts, rotations, or scaling. This characteristic is useful in the identification of poisoned samples that attempt to avoid detection.	For training, CNN-based defenses rely on a substantial, annotated dataset of both clean and poisoned instances. However, acquiring such datasets can be a challenge in IoT devices, which often function in distributed and dynamic environments. Thus, the limited training data available for these cases may restrict the efficacy of CNN-based countermeasures, possibly resulting in heightened susceptibility to developing poisoning assaults specific to IoT.
LF-K medoids and LF-EM [38]	_	NB	NB algorithm is recognized for its strong resilience against noise in the input data. This characteristic makes it valuable for detecting PAs that cause noise.	In cybersecurity applications, when the input features include numerical values such as the number of login attempts or file sizes, appropriately scaling or normalizing these features can distort the Naive Bayes model's decision-making process. This distortion can lead to poisoned samples being misclassified as benign or vice versa.
COEG, and COES [39]	Frequently verify the accuracy	DNN, CNN	Validating the accuracy of the poisoning detection model on a	Accuracy verification and updates of the poisoning detection model may necessitate temporary system

			regular basis can facilitate early detection of any performance deterioration, allowing for prompt intervention and model change.	downtime or the implementation of modifications that may affect regular operation. This may occur in financial systems, healthcare monitoring systems, or industrial control systems that leads to significant repercussions, such as service failures or delays in imperative decision-making processes.
LF [40]	SCOAP features with k-NN	CatBoost	SCOAP's outlier identification skills, when combined with k-NN's ability to identify data points that differ from the regular data distribution, can significantly improve the total capacity via identifying and separating out poisoning assaults from data.	1) SCOAP and k-NN algorithm depend on the choice of several parameters, such as the number of closest neighbors (k) or the distance metric employed. Inadequate parameter selection can result in decreased detection accuracy, heightened occurrence of false positives or false negatives, and overall deterioration in the efficacy of the PA detection system. 2) ML models are increasingly used in the healthcare domain for disease detection, pharmaceutical research, and patient monitoring. Implementing the SCOAP-k-NN method can help reduce PAs' impact on these models. However, the susceptibility to biased parameter selection may erode the confidence.
LF [41]	TrAdaBoost	AdaBoost, RF, SVM, DT, MLP, and LR	The TrAdaBoost approach effectively uses transfer learning to apply acquired information from the source domain to identify poisoning assaults. This enables the model to adjust to changing attack patterns or new data distributions of PAs.	The TrAdaBoost method entails the repetitive training of many weak learners, which can exhibit significant computational complexity, particularly when dealing with large-scale or high-dimensional datasets. In real-time systems like financial transactions or critical infrastructure monitoring, the computational complexity can provide difficulties when the detection system must promptly react to possible poisoning assaults.

LF [42]	_	DNN	DNNs have the ability to autonomously acquire hierarchical and intricate feature representations from unprocessed data. This feature extraction capability can be useful in situations where the PAs' attributes are not clearly defined or may change over time.	The occurrence of poisoning attacks is relatively low compared to harmless inputs, resulting in a notable imbalance in the training data classification. Consequently, the DNN model may be biased towards the majority class, leading to inadequate identification of the minority class (i.e., poisoned samples). This weakness can affect financial fraud detection and healthcare data analysis.

Although several defence mechanisms have been used, they contain weaknesses that make them insufficient to overcome this attack, requiring the development of more sophisticated defence strategies to identify and neutralize them. As a result, the following proposals will be made to protect and maintain the availability, robustness, and integrity of ML models with potential challenges in their implementation:

1. Deep Reinforcement Learning (DRL): These techniques can detect DPAs by training an agent to interact with the model and actively detect adversarial samples.

Challenge: Developing a reward function that accurately reflects the intended detection of DPA behavior.

2. Transfer Learning for Attack Detection: It involves utilizing knowledge gained from one domain into a different one. DPAs in the desired domain can be identified via a model that involves training on an original domain with recognized normal data and then applying the information gained to the interest domain.

Challenge: Achieving an adequate quantity of authentic poisoned samples for training and evaluation can present difficulties, as these samples are necessary for the fine-tuning or adaptation of the pre-trained model.

3. Evolutionary Computing (EC): The methods of EC (like genetic algorithms and particle swarm optimization) can be applied to find optimum or aberrant regions in the input space. EC might detect DPAs by analyzing the model's behavior on several samples.

Challenge: To achieve effective poisoning attack detection, evolutionary computing techniques required diverse and representative training datasets. As a result, acquiring an adequate quantity of annotated data for poisoning assaults, such as watermarking, can be challenging.

4. Ensemble methods are used to improve the detection of DPAs using techniques like bagging and boosting. By using various approaches or different portions of the data, training numerous models makes it easier to find discrepancies or inconsistencies in the predictions. Disagreements or inconsistencies throughout the samples of the ensemble may indicate potential assaults.

Challenge: Effective ensemble methods frequently depend on the significance and pertinence of input features to generate precise predictions. In order to achieve effective ensemble-based poisoning attack detection, it is critical to develop strong feature selection and significance estimation methods that can withstand clean label attacks.

5. The purpose of out-of-distribution detection (ODD) is to identify instances that significantly deviate from the distribution of the data set used for training. OOD instances could be an indication of upcoming DPA attempts. Generative models, density estimations, or Bayesian approaches can be employed to find OOD examples.

Challenge: Poisoning techniques, such as clean label assaults, aim to deliberately get the contaminated data closer to the original data. Hence, precisely delineating the demarcation between the "in-distribution" (normal) data and the "out-of-distribution" (anomalous or poisoned) data can be a challenging endeavor.

6. Model watermarking involves adding distinct IDs or signatures to the data used for training using methods such as Watermar-kNN or DeepSigns. Suppose the contaminated data was utilized for training or modifying the model. If that's the case, it can be discovered by searching for these watermarks when deploying or deriving models.

Challenge: The development of strong watermark recognition algorithms that can accurately differentiate between watermarked and non-watermarked models, especially when faced with noise or distortions, as well as model alterations, can be a tough task.

7. Analysis of Reconstruction Mistakes (RM): If autoencoders were used in the model's training, DPAs could be discovered by looking at the REs. REs from harmful examples can be identified as possible assault targets by establishing thresholds high enough to distinguish them from harmless examples.

Challenge: Establishing the appropriate thresholds for reconstruction errors to distinguish between data that follows a normal distribution and data that is not (poisoned) can be challenging.

8. Analyzing the features of the model to ascertain which are altered known as Feature Importance (FI) analysis. The model's predictions will be assessed by applying techniques like permutation importance or SHAP values to determine the impact of each feature. A DPA attack may be indicated by unusually high or low relevance rankings for specific features.
9. Challenge: Establishing robust feature significance calculation techniques that can withstand adversarial perturbations induced by clean labels is crucial.

10. Conclusion and Future Work

Artificial intelligence technologies are developing and flourishing very quickly; simultaneously, hostile attacks are increasing, including PAs, considered one of the most dangerous attacks targeting ML. The lack of information and the unavailability of attack databases hinder finding more appropriate solutions. This article aimed to give a complete and accurate overview of the different types of attacks and how they are used against diverse ML models, along with the countermeasures that were carefully studied to find out their determinants. Several solutions have also been proposed to maintain the availability, robustness, and integrity of ML models. Increasingly complex security risks are expected to emerge regularly, so security concerns related to ML will persist, requiring more attention from experts in the field. To strengthen the resilience against poisoning efforts, it is advisable for future work to incorporate adversarial training using suggested protective techniques such as deep reinforcement learning or ensemble methods. Given the unavailability of poisoned samples, potential methods to improve the effectiveness of adversarial training include the clustering techniques that produce inlier samples rather than outliers. Moreover, it is recommended that future research should address the application of poisoning attacks to novel areas, such as SQL injection and DDOS.

Acknowledgment

The authors' thankful Department of Computer Science, Collage of Science, Mustansiriyah University, for supporting this work.

References

[1] A. A. Springborg, M. K. Andersen, K. H. Hattel, and M. Albano, "Towards a secure API client generator for IoT devices", 2022. http://arxiv.org/abs/2201.00270

[2] A. Schwarzschild, M. Goldblum, A. Gupta, J. P. Dickerson, and T. Goldstein, "Just How Toxic is Data Poisoning? A Unified Benchmark for Backdoor and Data Poisoning Attacks", 2020. http://arxiv.org/abs/2006.12557

[3] A. H. Ali, M. Z. Abdullah, S. N. Abdul-Wahab, and M. Alsajri, "A Brief Review of Big Data Analytics Based on Machine Learning," Iraqi Journal for Computer Science and Mathematics, vol. 1, no. 2, pp. 13–15, 2020. http://dx.doi.org/10.52866/ijcsm.2020.01.02.002

[4] M. Naumov et al., "Deep Learning Recommendation Model for Personalization and Recommendation Systems," May 2019. https://doi.org/10.48550/arXiv.1906.00091

[5] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," Applied Sciences (Switzerland), vol. 9, no. 20. MDPI AG, 2019. https://doi.org/10.3390/app9204396

[6] M. Haqi Al-Tai, B. M. Nema, and A. Al-Sherbaz, "Deep Learning for Fake News Detection: Literature Review," Al-Mustansiriyah Journal of Science, vol. 34, no. 2, pp. 70–81, Jun. 2023. https://doi.org/10.23851/mjs.v34i2.1292

[7] Charu C. Aggarwal, and Chandan K. Reddy, "Date Clustering Algorithms and Applications" Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2014. https://doi.org/10.1201/9781315373515

[8] M. M. Mijwil, I. E. Salem, and M. M. Ismaeel, “The Significance of Machine Learning and Deep Learning Techniques in Cybersecurity: A Comprehensive Review,” Iraqi Journal for Computer Science and Mathematics, vol. 4, no. 1. College of Education, Al-Iraqia University, pp. 87–101, 2023. https://doi.org/10.52866/ijcsm.2023.01.01.008

[9] M. M. Taye, “Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions”, Computers, vol. 12, no. 5. MDPI, 01, 2023. https://doi.org/10.3390/computers12050091

[10] M. Naeem, S. T. H. Rizvi, and A. Coronato, “A Gentle Introduction to Reinforcement Learning and its Application in Different Fields,” IEEE Access, vol. 8, pp. 209320–209344, 2020. https://doi.org/10.1109/ACCESS.2020.3038605

[11] J. Lin, L. Dang, M. Rahouti, and K. Xiong, “ML Attack Models: Adversarial Attacks and Data Poisoning Attacks”, 2021. https://doi.org/10.48550/arXiv.2112.02797

[12] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, “The security of machine learning,” Mach Learn, vol. 81, no. 2, pp. 121–148, 2010. http://doi.org/10.1007/s10994-010-5188-5

[13] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, and B. Li, “Untargeted Backdoor Watermark: Towards Harmless and Stealthy Dataset Copyright Protection”, Advances in Neural Information Processing Systems, vol. 35, pp. 13238-13250., 2022. https://doi.org/10.48550/arXiv.2210.00875

[14] L. Cassiday, “Clean label: The next generation,” International News on Fats, Oils and Related Materials, vol. 28, no. 8, pp. 6–10, Sep. 2017. http://dx.doi.org/10.21748/inform.09.2017.06

[15] J. Horkoff, “Non-functional requirements for machine learning: Challenges and new directions”, Proceedings of the IEEE International Conference on Requirements Engineering, IEEE Computer Society, pp. 386–391, 2019. https://doi.org/10.1109/RE.2019.00050

[16] L. Obiora Nweke, “Using the CIA and AAA Models to explain Cybersecurity Activities”, PM World Journal, vol. 6, no.12, pp. 1-3., 2017. https://pmworldlibrary.net/article/using-the-cia-and-aaa-models-to-explain-cybersecurity-activities/

[17] H. bediar Hashim, “Challenges and Security Vulnerabilities to Impact on Database Systems,” Al-Mustansiriyah Journal of Science, vol. 29, no. 2, pp. 117–125, Nov. 2018. https://doi.org/10.23851/mjs.v29i2.332

[18] B. Biggio and F. Roli, “Wild patterns: Ten years after the rise of adversarial machine learning,” Pattern Recognit, vol. 84, pp. 317–331, 2018. https://doi.org/10.1016/j.patcog.2018.07.023

[19] H. H. Ali, J. R. Naif, and W. R. Humood, “A New Smart Home Intruder Detection System Based on Deep Learning,” Al-Mustansiriyah Journal of Science, vol. 34, no. 2, pp. 60–69, 2023. https://doi.org/10.23851/mjs.v34i2.1267

[20] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning”, Proceedings - IEEE Symposium on Security and Privacy, Institute of Electrical and Electronics Engineers Inc., pp. 739–753, 2019. http://doi.org/ 10.1109/SP.2019.00065

[21] B. Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning”, Conference on Computer and Communications Security, pp.603-613, 2017. https://doi.org/10.1145/3133956.3134012

[22] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, “Exploiting unintended feature leakage in collaborative learning,” in Proceedings - IEEE Symposium on Security and Privacy, Institute of Electrical and Electronics Engineers Inc., pp. 691–706, 2019. https://doi.org/10.1109/SP.2019.00029

[23] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership Inference Attacks Against Machine Learning Models”, Proceedings - IEEE Symposium on Security and Privacy, Institute of Electrical and Electronics Engineers Inc., pp. 3–18, 2017. https://doi.org/10.1109/SP.2017.41

[24] W. Xu, Y. Qi, and D. Evans, “Automatically Evading Classifiers A Case Study on PDF Malware Classifiers”, Network and Distributed System Security Symposium, 2016. http://dx.doi.org/10.14722/ndss.2016.23115

[25] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, “Systematic poisoning attacks on and defenses for machine learning in healthcare,” IEEE J Biomed Health Inform, vol. 19, no. 6, pp. 1893–1905, Nov. 2015. https://doi.org/10.1109/JBHI.2014.2344095

[26] H. Xiao XIAOHU, G. Fumera, and F. Roli, “Is Feature Selection Secure against Training Data Poisoning?”, international conference on machine learning, pp. 1689-1698, 2015. https://pure.manchester.ac.uk/ws/portalfiles/portal/32297390/

[27] R. Laishram and V. V. Phoha, “Curie: A method for protecting SVM Classifier from Poisoning Attack”, 2016. https://doi.org/10.48550/arXiv.1606.01584

[28] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, “Data Poisoning Attacks on Factorization-Based Collaborative Filtering”, Conference on Neural Information Processing Systems, vol. 29, 2016. https://dl.acm.org/doi/10.5555/3157096.3157308

[29] P. P. K. Chan, Z. M. He, H. Li, and C. C. Hsu, “Data sanitization against adversarial label contamination based on data complexity,” International Journal of Machine Learning and Cybernetics, vol. 9, no. 6, pp. 1039–1052, 2018. http://doi.org/10.1007/s13042-016-0629-5

[30] N. Baracaldo, B. Chen, H. Ludwig, and J. A. Safavi, “Mitigating poisoning attacks on machine learning models: A data Provenance based approach”, AISec 2017 - Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, co-located with CCS 2017, Association for Computing Machinery, Inc, pp. 103–110, 2017. http://dx.doi.org/10.1145/3128572.3140450

[31] S. Chen et al., “Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach,” Comput Secur, vol. 73, pp. 326–344,2018. http://doi.org/10.1016/j.cose.2017.11.007

[32] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, “Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning”, Proceedings - IEEE Symposium on Security and Privacy, Institute of Electrical and Electronics Engineers Inc., pp. 19–35, 2018. http://doi.org/10.1109/SP.2018.00057

[33] A. Paudice, L. Muñoz-González, and E. C. Lupu, “Label Sanitization against Label Flipping Poisoning Attacks”, Conference on Machine Learning and Knowledge Discovery in Databases, vol. 11329, pp. 5-15, 2018. https://doi.org/10.1007/978-3-030-13453-2_1

[34] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, “Understanding distributed poisoning attack in federated learning,” in Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS, IEEE Computer Society, pp. 233–239, 2019. http://doi.org/10.1109/ICPADS47876.2019.00042

[35] R. Taheri, R. Javidan, M. Shojafar, Z. Pooranian, A. Miri, and M. Conti, “On defending against label flipping attacks on malware detection systems,” Neural Comput. Appl, vol. 32, no. 18, pp. 14781–14800, 2020. http://doi.org/10.1007/s00521-020-04831-9

[36] P. P. K. Chan, Z. He, X. Hu, E. C. C. Tsang, D. S. Yeung, and W. W. Y. Ng, “Causative label flip attack detection with data complexity measures”, International Journal of Machine Learning and Cybernetics, vol. 12, no. 1, pp. 103–116, 2021. http://doi.org/10.1007/s13042-020-01159-7

[37] H. Liu, D. Li, and Y. Li, “Poisonous Label Attack: Black-Box Data Poisoning Attack with Enhanced Conditional DCGAN,” Neural Process Lett, vol. 53, no. 6, pp. 4117–4142, 2021. http://doi.org/10.1007/s11063-021-10584-w

[38] H. Zhang, N. Cheng, Y. Zhang, and Z. Li, “Label flipping attacks against Naive Bayes on spam filtering systems,” Applied Intelligence, vol. 51, no. 7, pp. 4503–4514, 2021. http://doi.org/10.1007/s10489-020-02086-4

[39] B. Zhao and Y. Lao, “Towards Class-Oriented Poisoning Attacks Against Neural Networks”, Conference on Applications of Computer Vision (WACV), 2022. https://doi.org/10.1109/WACV51458.2022.00230

[40] R. Sharma, G. K. Sharma, and M. Pattanaik, “A CatBoost Based Approach to Detect Label Flipping Poisoning Attack in Hardware Trojan Detection Systems,” Journal of Electronic Testing: Theory and Applications (JETTA), vol. 38, no. 6, pp. 667–682, 2022. http://doi.org/10.1007/s10836-022-06035-6

[41] Q. Li, X. Wang, F. Wang, and C. Wang, “A Label Flipping Attack on Machine Learning Model and Its Defense Mechanism,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Science and Business Media Deutschland GmbH, pp. 490–506, 2023. http://doi.org/10.1007/978-3-031-22677-9_26

[42] H. Mohammadian, A. Lashkari, and A. Ghorbani, “Evaluating Label Flipping Attack in Deep Learning-Based NIDS,” INSTICC, pp. 597–603, 2023. http://dx.doi.org/10.5220/0010867900003120