INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Mitigating Gradient-Based Data Poisoning Attacks on Machine Learning Models: A Statistical Detection Method

Lavanya Sanapala[1]*, **Lakshmeeswari Gondi**[2]

1 Research Scholar, Department of CSE, GITAM School of Technology, GITAM Deemed to be University, India
2 Associate Professor, Department of CSE, GITAM School of Technology, GITAM Deemed to be University, India

OPEN ACCESS

Received: 02-04-2024

Accepted: 12-05-2024

Published: 29-05-2024

Citation: Sanapala L, Gondi L (2024) Mitigating Gradient-Based Data Poisoning Attacks on Machine Learning Models: A Statistical Detection Method. Indian Journal of Science and Technology 17(21): 2218-2231. https://doi.org/10.17485/IJST/v17i21.1035

* **Corresponding author**.

lavanya.sanapala@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: This research paper aims to develop a novel method for identifying gradient-based data poisoning attacks on industrial applications like autonomous vehicles and intelligent healthcare systems relying on machine learning and deep learning techniques. These algorithms performs well only if they are trained on good quality dataset. However, the ML models are prone to data poisoning attacks, targeting the training dataset, manipulate its input samples such that the machine learning algorithm gets confused and produces wrong predictions. The current detection techniques are effective to detect known attacks and lack generalized detection to unknown attacks. To address this issue, this paper aims to integrate security elements within the machine learning framework, guaranteeing effective identification and mitigation of known and unknown threats and achieve generalized detection. **Methods:** ML Filter, a unique attack detection approach integrates ML-Filter Detection Algorithm and the Statistical Perturbation Bounds Identification Algorithm to determine the given dataset is poisoned or not. DBSCAN algorithm is used to divide the dataset into several smaller subsets and perform algorithmic analysis for detection. The performance of the proposed method is evaluated in terms of True positive rate and significance test accuracy. **Findings:** The probability distribution differences between original and poisoned datasets vary with change in perturbation size rather than the datasets and ML models use for application. This finding lead to determine the perturbation bounds using statistical pairwise distance metrics and corresponding significance tests computed on the results. ML Filter demonstrates a high detection rate of 99.63% for known attacks and achieves a generalized detection accuracy of 98% for unknown attacks. **Novelty:** A secured ML architecture and a unique statistical detection approach ML-Filter, effectively detect data poisoning attacks, demonstrating significant advancements in detecting both known and unknown threats in industrial applications utilizing machine learning and deep learning algorithms.

Keywords: Privacy and security; Adversarial machine learning; Secured ML Architecture; ML-Filter; Statistical Perturbation Bounds Identification Algorithm

1 Introduction

ML algorithms like Support Vector Machine (SVM), Decision Trees, Random Forest, and the sensational Convolutional Neural Networks for image classification tasks are few examples for supervised classification problems. These algorithms are used to automate a variety of security-critical tasks, including malware detection, traffic forecasting for unmanned vehicles, real-time object detection, online fraud detection, and many more. In supervised learning, the training dataset provides the model with patterns of input data and matching class labels. The training dataset's quality determines how well the machine learning model performs. However, traditional machine learning architectures are susceptible to potential vulnerabilities and lack security measures.

Data poisoning attacks have become prevalent in recent years, specifically targeting machine learning algorithms used for classification problems using supervised learning. Fahri Anıl Yerlikaya et al.[1] have demonstrated that several machine learning and deep learning models used for spam, malware, and cancer detection datasets are susceptible to attacks by testing their resilience. ML hackers deploy a variety of attack methods to undermine the effectiveness of the model, breach security measures aimed at the availability and integrity of the dataset, and ultimately compromise the dependability of the ML model's end-to-end usefulness. According to a recent survey by the authors of[2], data poisoning attacks rank first among ML threats that significantly impact industrial machine learning applications. Additionally, identifying these attacks is challenging.

Training datasets are targets of data poisoning attacks. The integrity of the dataset is put at risk by this attack, which inserts false or attack samples into the training data. Neither a machine learning algorithm nor humans can distinguish between these samples. When such misleading occurrences exist in the training data, the model learns erroneous decision limits, which reduces the efficiency of the machine learning process and its outcomes. Poisoning attack samples can be constructed using a number of methods and incorporated to a training set for offensive purposes. This work focuses exclusively on gradient-based techniques for contaminating the training dataset. Gradient-based techniques include, for instance, Madrey et al.'s attack (MAD), Basic Iterative Method (BIM), Momentive Iterative Method (MIM), Carlini and Wagner Attack (CW), Projected Gradient Descent (PGD), and Fast Gradient Sign Method (FGSM)[3]. These techniques take the original training samples from the dataset and use a specific method to artificially alter the selected training samples into a poisoned data set. Poisoned datasets are those that contain samples that have been contaminated; non-poisoned or clean datasets are those that do not contain samples that have been contaminated.

There are a plethora of data-poisoning attack detection methods developed by the researchers in this field. Anisie Uwimana et al., in their study, assess the reliability of a Mahalanobis distance-based confidence score identifying malaria-parasitized and uninfected cells. The classification prediction confidence score for every class is determined by computing Mahalanobis distance to detect anomalies in their distributions. The authors then improve the performance of deep learning models on adversarial and out-of-distribution samples by training the neural networks with a plausible additional noise to the input data. While the detection accuracy of attack samples on text data exceeds 95%, its performance in detecting attack samples on image data is substandard[4]. Yubo Hou and his co-authors trained the neural network using the Generative Adversarial Network (MDAN), which computes the Mahalanobis distance score for a given input. A high score indicates an anomaly. This approach only obtained a 63% detection accuracy on the image data, which is not desirable for security critical applications[5]. Another method of thresholding the distance from a Gaussian distribution fitted to the target class representations, the author of[6] investigates a technique for identifying adversarial samples and states that the Mahalanobis distance detecting technique is the most vulnerable to attack. Fabio Carrara and fellow authors have proposed ENAD, an ensemble approach for adversarial detection that improves performance by integrating layer-specific scores from three independent detectors (LID, Mahalanobis, and OCSVM), achieving significantly enhanced performance on benchmark datasets, methods, and attacks but requiring training[7]. Ibrahim Aliyu et al., in their study, have performed statistical analysis on the attack samples with the help of distance-based statistical tests to understand the statistical deviations of the attack samples from the original ones and detect their presence in the training set. Then, the ML model trains these attack samples to learn their patterns and identify them as malicious ones. This entire procedure is called adversarial training (AT)[8]. Although these techniques perform well on known attacks, they are still susceptible to unknown attacks and lack generality. The current detection strategies either protect themselves by improving robustness of model or rely on additional ML model which is trained on the attack patterns. Either of the strategies are ineffective when the adversary is aware of the detection methods and employ new attack to explode them[9]. Scaling to large models becomes difficult due to increased cost and resource requirements. Hence, a generalized approach to identify antagonistic situations and a secured ML architecture is sought.

The main contributions of this paper are:

1. A Secured Machine Learning Architecture to safeguard against gradient-based data poisoning attacks.
2. A novel detection method, ML-Filter Detection Algorithm (MLF-DA), to identify data poisoning attacks in lieu of adversarial training.
3. A Statistical Perturbation Bounds Identification Algorithm (SPBIA) was developed to determine the perturbation bounds of the attack dataset.
4. The ML-Filter efficiently detects known and unknown attacks with high detection rates. Thus, the proposed method achieves generalized detection, which was a limitation of the earlier methods.

The remaining paper is organized into following sections. Section 2 provides the design and implementation of secured ML architecture and ML-Filter. Section 3 presents the results and discussion. Section 5 concludes with some final thoughts.

2 Methodology

The design and implementation of the proposed Secured Machine Learning Architecture (SMLA) is presented in this section. The SMLA integrates the ML-Filter (MLF) as a secured feature into the traditional ML Architecture to safeguard against the gradient-based data poisoning attacks. An ML-Filter Detection Algorithm (MLF-DA) along with Statistical Perturbation Bounds Method (SPBM) contributes in identifying the presence of Poisoned data in the training dataset.

2.1 Secured ML Architecture

The proposed machine learning architecture introduces a Machine Learning Filter (ML-Filter) between the data input and ML model. When the adversary tries to input the malicious data (i.e., the poisoned data samples) into the system, this data is redirected to the ML-Filter rather than the ML Model directly, as shown in Figure 1. The ML-Filter checks if the input data contains malicious information or not. If the ML-Filter detects some malicious content, it blocks the data from entering the system and ensures that the poisonous attack does not affect the ML model. The maliciousness of the input data is determined by the ML-Filter detection Algorithm.

The design overview of ML-Filter - Detection Algorithm model is shown in Figure 2. Firstly, the input dataset is redirected to ML-Filter. When it enters the ML-Filter, it triggers the Detection Algorithm. The ML-Filter Detection Algorithm (MLF-DA) captures the input dataset, employs the Statistical Perturbation Bounds Identification Algorithm (SPBIA) to perform statistical operations, and determines whether it contains poisonous samples or not. The detailed procedure is explained in methodology section 5.

The symbols used to explain the methodology are $\boldsymbol{D}$ - Dataset, $\boldsymbol{P}$ - Poisoned Sample, S(C) - Set of Clean samples, S($\boldsymbol{P}$) - Set of Poisoned samples, $\boldsymbol{M}$ - Machine Learning Model (Classifier), ED - Prior Experimental data, S(ED) - Set of experimental data, $\boldsymbol{L}$ - Lower bound, $\boldsymbol{U}$ - Upper bound, $\tau \boldsymbol{I}$ - Threshold Interval, ϕ - Resulting Statistical Deviation measure vector, LDM - Laplacian Distance Metric, SPBIA - Statistical Perturbation Bounds Identification Algorithm.

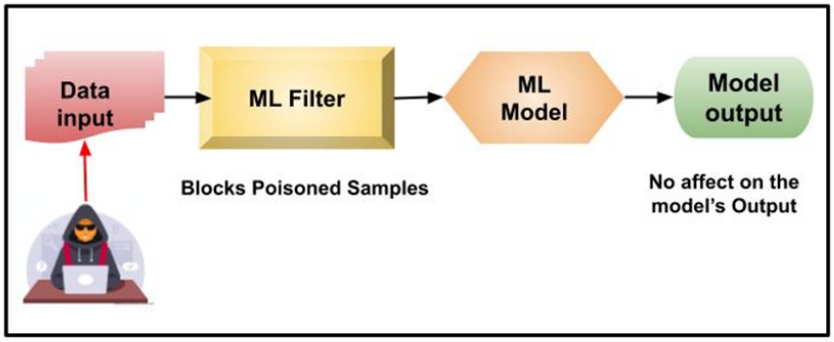


Fig 1. The Proposed Secured Machine Learning Architecture

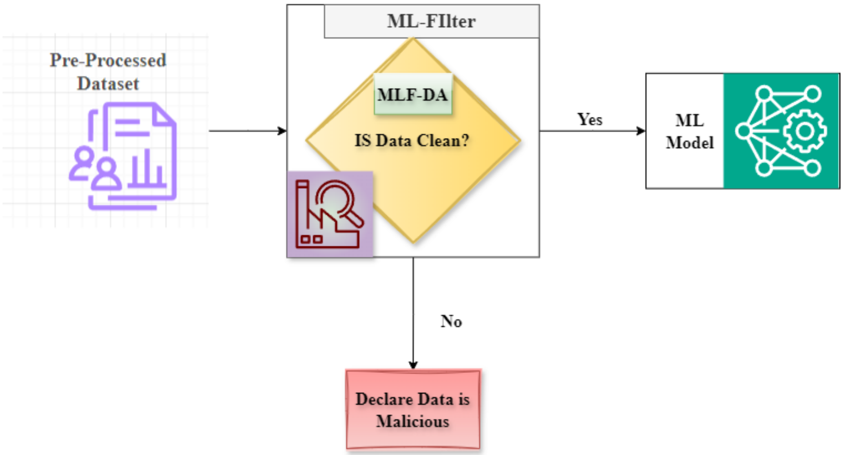


Fig 2. The design overview of ML Filter detection model

ML-Filter Detection Algorithm (MLF-DA) receives the input dataset and determines whether it is a poisoned dataset or not according to Algorithm 1. It requires a dataset to perform statistical operations on it, a parameter ϕ to store the result of statistical operations, and a clustering algorithm to divide the input dataset into smaller subsets.

ALGORITHM 1: Machine Learning Filter - Detection Algorithm (MLF-DA)

```
Let ID(S)new denotes the input sample dataset fed to the ML-Filter.
Input:
  New dataset ID(S)new
Begin:
Divide the input dataset into clusters using DBSCAN
n = number of classes output by DBSCAN
Loop
  i= 1
$\phi$ = Compute LDM (Ci, Cn)
# [LDM (C1, C2), LDM (C2, C3), LDM (C1, C3)]
$\phi$ = append $\phi$ [vector of LDM deviations measures.]
Until n
End Loop
Res = SPBIA($\phi$)
if Res = True then Determine as Poisonous
else Pass the sample ID(S)new to the ML model.
End
Output: Returns Decision Poisonous or Non-Poisonous
```

First, the DBSCAN algorithm divides the input dataset into several subsets based on their similarity scores. Then, a unique pair of the subsets are chosen at random and given as inputs to the Laplace Pairwise Deviation Metric (LPM) function to analyze the statistical characteristics of the dataset. For example, assume there are 'n' classes in the dataset, C_1, C_2, C_3…., C_n. The elements of C_1, C_2, ... C_n could be the group of data with similar probability distributions. (C_1, C_2), (C_2, C_k), …. (C_k, C_n). All the pairs went through the statistical process to examine anomalies in the subsets as shown in Figure 3.

The Laplacian kernel pairwise distance metric (LPDM), is a statistical distance metric, finds the probability distribution distance measures between a pair of vectors (in this case, the clean and poisoned samples) according to Equation (1)[(10)]. Given two n-dimensional vectors x (clean) and x' (poisoned), the Laplacian kernel K is defined as follows.

$$K(x,x') = e^{(-\|x-x'\|1)/\sigma} \tag{1}$$

Where, ||x-x' ||1 denotes the L1-norm between a and b, and σ is the scale parameter. The output vectors of deviations in the given pair of datasets are required to determine the perturbation bounds. The original Laplacian distance metric is modified to the Laplace Deviation Metric (LDM) to compute the deviations defined in the Equation (2).

$$LDM\left(x,x'\right) = 1 - K\left(x,x'\right) \tag{2}$$

```
# Modified Laplace function to calculate deviation
    def Laplace_deviation(x, y, gamma):
      dist = np.linalg.norm(x-y)
      deviation = 1-np.exp(-gamma*dist)
    return deviation.
    # End of the function
```

This code snippet calculates the deviation of the poisoned data set from the non-poisoned dataset. The resulting deviation measures of the LDM test are stored in the vector space ϕ.

Now, the Detection algorithm calls the SPBIA function (detailed in Algorithm 3) to determine if any of the values of the vector space ϕ matches the values of the threshold interval derived by the SPBIA. The SPBIA returns a Boolean value, True or False. If it returns True, the MLF determines that the dataset is poisonous or non-poisonous otherwise. The complete workflow of MLF-DA is shown in Figure 5.

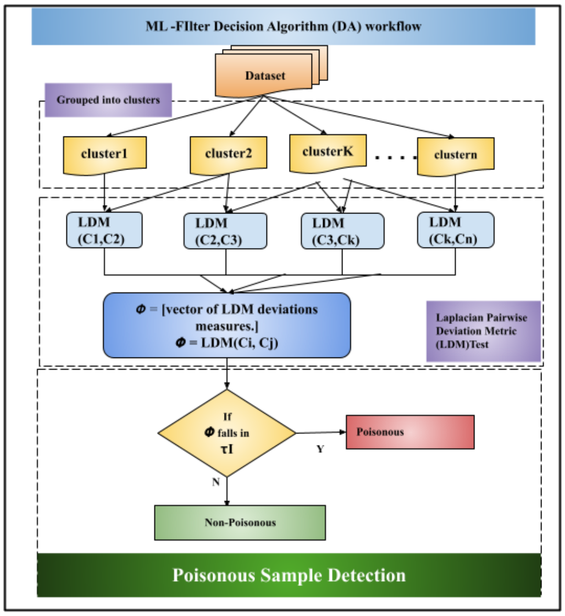


Fig 3. The detailed view of ML-Filter workflow

The SPBIA algorithm derives the deviation's threshold interval τI that helps determine the ML-Filter to determine the state (Poisoned or Non-Poisoned) of the input dataset. The SPBIA entirely depends on the prior knowledge (ED) to derive τI .

The ED is acquired from the extensive experiments of LDM computed for poisoned and non-poisoned datasets of the dataset considered for deriving τI as discussed in Algorithm 2, section 2.1.1.

2.1.1 Prior knowledge acquisition process

The probability distribution of a poisoned dataset differs from the original dataset. These differences are captured as prior knowledge/Experimental Data (ED) with the help of statistical pairwise deviation metric Laplace deviation metric (LDM) defined in Equation (2). For conducting statistical experimental analysis, the original dataset as well as poisoned datasets are required. Because of unavailability of poisoned datasets, three different attack poisoned samples FGSM, PGD, and CW, are synthetically generated for the chosen dataset with perturbation size denoted by epsilon (ε) values of v_i*e^{-1}, v_i*e^{-2}, v_i*e^{-3}, v_i*e-^{4}, v_i*e^{-5}, v_i*e^{-6}, where v_i = {0.1,0.5,1.0,2.0, 3.0}.

Let $\boldsymbol{P}_f$, $\boldsymbol{P}_g$, and $\boldsymbol{P}_c$ denoted by the corresponding set notations S($\boldsymbol{P}_f$) = {[x_{f1}', x_{f2}', x_{f3}', …. x_{fn}'], ∀ v_i*e^{-k}}, S($\boldsymbol{P}_g$) = {[x_{g1}', x_{g2}', x_{g3}', …. x_{gn}'], v_i*e^{-k}}, S($\boldsymbol{P}_c$) = {[x_{c1}', x_{c2}', x_{c3}', …. x_{cn}'], v_i*e^{-k}}. The values xi', i=1, 2, ..., n are the natural values of poisoned image samples generated by each attack algorithm, k = 1,2,3,4,5, and 6. S(C) is the set of non-poisoned image samples. These two sets were given as input to the Laplacian deviation metric test as defined in Equation (2). The outcomes are the probability distribution deviation measures computed between the poisoned and non-poisoned datasets stored as Experimental Data (ED).

```
ALGORITHM 2: Prior knowledge acquisition
Input:
S($\boldsymbol{P}_f$ ) = {[$x_{f1}'$, $x_{f2}'$, $x_{f3}'$, …. $x_{fn}'$ ]}, #Poisoned set of FGSM
S($\boldsymbol{P}_g$ ) = {[$x_{g1}'$, $x_{g2}'$, $x_{g3}'$, …. $x_{gn}'$ ]}, #Poisoned set of PGD
S($\boldsymbol{P}_c$ ) = {[$x_{c1}'$, $x_{c2}'$, $x_{c3}'$, …. $x_{cn}'$ ]}, #Poisoned set of CW
S(C) = {$x_1$, $x_2$, $x_3$, …. $x_n$}.
Begin:
Loop until LDM is computed ∀ the poisoned sets
ED = LDM( (S($\boldsymbol{P}_i$ ), S(C) )
#Populate deviation measures according to Equation (2).
ED = $\bigcup_i (LDM(S(Pi), S(C))$
# append ED with deviation measures ∀ the statistical tests ((S($\boldsymbol{P}_i$ ), S(C))
# i = f, g, and c where f = fgsm, g=pgd, and c = CW attack samples
return ED
End
Output: Prior Knowledge ED.
```

The prior knowledge data has been utilized by the SPBIA algorithm in deriving the statistical perturbation bounds of the poisoning attacks according to Algorithm 3. The algorithm takes the ED values as input. First, the lower and upper bounds of the deviation measures of each poisoned sets ED(S($\boldsymbol{P}_f$)), ED(S($\boldsymbol{P}_g$)), and ED(S($\boldsymbol{P}_c$)) are determined as discussed in following sections 5.12 and 5.13. Secondly, the Maximum Likelihood Estimation (MLE) is applied to get the point estimates of the lower and upper bounds, finally serving as the statistical perturbation bounds, which form the threshold interval τI .

2.1.2 Lower Bound

Let ***L*** denote the lower bound of the deviation measures observed from the statistical test. Laplacian pairwise Deviation Metric (LDM) is the minimum possible deviation observed between the poisonous and clean sample datasets. Given a set of experimental outcomes S(ED), upon each poisoned dataset represented as S(ED) = {d_1, d_2, d_3, ……, d_n}. Then, the lower bound '***L***' is the smallest value in the set. Mathematically the lower bound ***L*** is defined as

$$\begin{aligned} L &= min(S(ED)) \\ &= min(d_1, d_2, d_3, \ldots\ldots, d_p) \end{aligned} \tag{3}$$

2.1.3 Upper Bound

The maximum deviation between the poisonous and clean sample datasets is the upper bound ***U*** of the deviations observed from the statistical test. The ***U*** value has been derived as follows. Given a set of experimental outcomes S(ED), where S(ED) = {d_1, d_2, d_3, ……, d_n}, the upper bound '***U***' is the maximum value in the set. Mathematically the lower bound ***U*** is defined as

$$\begin{aligned} U &= max(S(ED)) \\ &= max(d_1, d_2, d_3, \ldots\ldots, d_p) \end{aligned} \tag{4}$$

2.1.4 Maximum Likelihood Estimation

A statistical technique for determining the parameters of a probability distribution that has been conjectured based on the results of some observed data is called maximum likelihood estimation (MLE). By maximizing a likelihood function, the observed facts are rendered as probable as feasible in light of the underlying statistical model. The likelihood function's maximum point in the parameter space (***Z***) is known as the Max_Likelihood_Estimate, is defined in the Equation (5)[11].

$$\widehat{\mu} = \frac{1}{N}\sum_{i=1}^{N} E_i \tag{5}$$

In this study, the MLE function is computed to pick the point estimates of Lower and upper bounds for τI .

```
ALGORITHM 3: Statistical Perturbation Bounds Identification Algorithm
Input:
    $\phi$ = Result vector of LDM test on input dataset to ML-Filter
    ED = Prior experimental data
Begin:
Find L as per Equation (3)
Find U as per Equation (4)
LB = Compute MLE (L) for determining the lower bound for ED as per Equation (5)
UB = Compute MLE (U) for determining the upper bound for ED as per Equation (5)
$\tau I = [LB, UB]$
if LB <= $\phi$ <= UB then   # if $\phi$ falls in $\tau I$
        Res =  True #boolean value (Poisoned dataset)
            else
        Res = False #boolean value (Non-Poisoned dataset)
end
Output: Res   #boolean value (True/False)
```

3 Results and Discussion

In this section, we briefly describe the experimental setup to implement the ML-Filter and the metrics used to evaluate the MLF-DA performance. We report the findings of the experimental results and discussion on the outcomes.

3.1 Experimental Setup and Evaluation

As discussed earlier in the methodology section, the MLF-DA decision is based on the SPBIA output based on the threshold interval τI derived from prior knowledge, i.e., known probability distribution deviations. Prior knowledge is acquired from the MNIST dataset. It is a benchmark dataset for handwritten digit recognition, containing 28x28 grayscale images of numbers from 0 to 9. The training and test sets have 60,000 and 10,000 images, respectively[12].

To analyze the statistical deviations between the non-poisoned and poisoned datasets of MNIST, we require the poisoned dataset of MNIST. The poisoned MNIST sample of 16 sets were synthetically generated using Nicolas Carlini & Wagner et al. attack algorithms available from github source[13].

We evaluated the proposed MLF-DA detection accuracy on three benchmark datasets, namely CIFAR10, FashionMNIST, and CIFAR100[14] [15], to new attack types, namely BIM[16], MIM[17], MAD[18], FGSM[19], PGD[20] and CW[3]. Novel CNN classifiers were built for MNIST, CIFAR10, FashinMNIST, and CIFAR100, for which an accuracy of 99.20%, 92.80%, 99%, and 96%, respectively was achieved.

The ML-Filter employs an unsupervised density-based spatial clustering algorithm of applications with noise (DBSCAN), which divides the input dataset into smaller subsets, as discussed in algorithm 1. It uses statistical functions such as Principal Component Analysis (PCA) / histogram-oriented gradient (HOG) methods for feature extraction and distance-based clustering to divide the dataset into smaller groups.

Two evaluation metrics used to check the performance of the proposed statistical method and ML filter discussed here.

3.1.1 True Positive Rate

The true positive metric here calculates the percentage of poisoned image set detection with respect to the total number of image samples in the training set (clean + poisonous) for each sample set tested.

$$TPR = \frac{No.of\ P\_S}{No.of\ C\_S + T\_S} \tag{6}$$

Where P_S represents poisonous samples detected as poisonous, C_S for clean samples detected as clean and T_S for total number of samples (i.e., clean + poisonous 7000 in our case).

3.1.2 Statistical Significance Test

Conjecture (H0): The Laplacian deviation measure between C_i and C_j denoted by Θ does not belongs to SPBs if the image samples are non – poisonous.

Research hypothesis (Ha): The Laplacian deviation measure between C_i and C_j denoted by Θ belongs to SPBs if the image samples are Poisonous.

The percentage of accuracy that the SPBM detected the poisonous samples is calculated according to the Equation (7).

$$Accuracy\ = \frac{TP\ +\ TN}{TP\ +\ TN\ +\ FP\ +\ FN} \tag{7}$$

where TP, FN, FP, TN, refers to true positive, false negative, false positive and true negative rates respectively.

A high rate of accuracy to accept the hypothesis indicates that the input dataset is non-poisonous. A high rate of accuracy to accept the Alternative hypothesis indicates that the input data is poisonous.

The proposed ML-Filter Detection Algorithm is tested for its detection accuracy on 16 test (poisoned) datasets with different parameter settings as shown in Table 1. The datasets 1 to 16 have poisoned samples of different attacks. For ease of performance analysis, these datasets are categorized into two types named as known and unknown attack categories. The prior knowledge gained by the SPBIA containing the known deviation measures of MNIST dataset are categorized into known attacks. The deviation measures tested for new input datasets (CIFAR10, FashionMNIST, and CIFAR100) and attacking strategies (BIM, MIM, and MAD, FGSM, PGD, and CW) are categorized into unknown attacks. For example, the LDM deviations obtained for MNIST and their poisoned samples of FGSM, PGD, and CW are known to SPBIA and thus they come under, known attack category. But the LDM deviations of MNIST and their poisoned samples of BIM, MIM, and MAD attacks are not known by the SPBIA. Similarly, the LDM deviations of CIFAR10 and their poisoned samples of FGSM, PGD, CW, BIM, MIM, and MAD attacks are not known by the SPBIA. Thus, they fall under the unknown attack category. The ✔ mark indicate the successful detection of attack, and **X** mark indicating the detection is not possible. The earlier works[3],[4],[6],[8] achieved detectability at epsilon value greater than 2 but failed in detecting the attack samples with perturbation values < 2.0 and when encounter with new attack types. Our ML-Filter achieved the credibility for detection of those attacks samples of unknown attacks and known attacks and with the attack samples with minimal perturbations, thus overcoming the lack of generalized detection by the earlier works.

Table 1. The parameters of the Test datasets used for performance evaluation of ML-DA

Dataset Number	Target Dataset	Attack Methods used to poison the dataset	Epsilon size of poisoned image	Attack Category	
				Known	Unknown
Set-1	MNIST	FGSM, PGD, C&W	0.2*e-6	✔	-
Set-2	MNIST	FGSM, PGD, C&W	0.8*e-6	✔	-
Set-3	MNIST	FGSM, PGD, C&W	1.5*e-6	✔	-
Set-4	MNIST	FGSM, PGD, C&W	2.5*e-6	✔	-
Set-5	CIFAR 10	FGSM, PGD, C&W	0.1*e-6, 0.5*e-6, 1*e-6, 2*e-6, 3*e-6	-	✔
Set-6	FashionMNIST	FGSM, PGD, C&W	0.1*e-6, 0.5*e-6, 1*e-6, 2*e-6, 3*e-6	-	✔
Set-7	CIFAR 100	FGSM, PGD, C&W	0.1*e-6, 0.5*e-6, 1*e-6, 2*e-6, 3*e-6	-	✔
Set-8	MNIST	BIM, MIM, MAD	0.1*e-6, 0.5*e-6, 1*e-6, 2*e-6, 3*e-6	-	✔
Set-9	CIFAR 10	BIM, MIM, MAD	0.1*e-6, 0.5*e-6, 1*e-6, 2*e-6, 3*e-6	-	✔

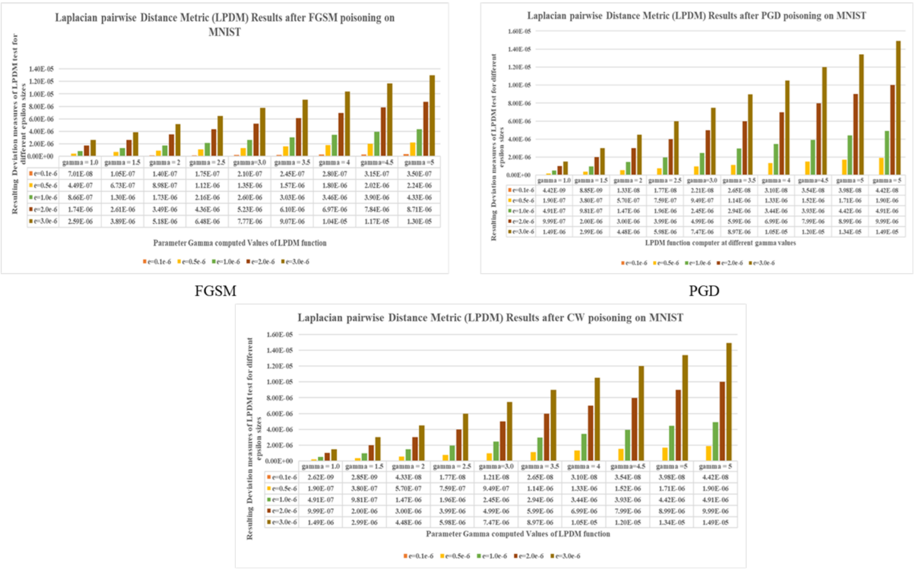
Continued on next page

Table 1 continued

Set-10	FashionMNIST	BIM, MIM, MAD	0.1*e-6, 0.5*e-6, 1*e-6, 2*e-6, 3*e-6	-	✔
Set-11	CIFAR 100	BIM, MIM, MAD	0.1*e-6, 0.5*e-6, 1*e-6, 2*e-6, 3*e-6	-	✔
Set-12	MNIST	FGSM, PGD, C&W, MIM, BIM, MAD	1	-	✔
Set-13	MNIST	FGSM, PGD, C&W, MIM, BIM, MAD	2	-	✔
Set-15	FashionMNIST	FGSM, PGD, C&W, MIM, BIM, MAD	1	-	✔
Set-16	FashionMNIST	FGSM, PGD, C&W, MIM, BIM, MAD	2	-	✔

3.2 Results of Laplacian Pairwise Deviation Metric computed for prior knowledge acquisition

This section reports the deviation measures obtained after LDM test computed between the original MNIST dataset and synthetically generated poisoned datasets of FGSM, PGD and CW are presented in Figure 4. The graph clearly indicates that the deviation measure increases with the increase in the epsilon size and gamma parameter. So, there is a significant relation between the epsilon size and the corresponding deviation measure. These results are used for prior knowledge acquisition and determine the statistical differences its range for MNIST dataset.



	gamma = 1.0	gamma = 1.5	gamma = 2	gamma = 2.5	gamma=3.0	gamma = 3.5	gamma = 4	gamma=4.5	gamma =5
e=0.1e-6	7.01E-08	1.05E-07	1.40E-07	1.75E-07	2.10E-07	2.45E-07	2.80E-07	3.15E-07	3.50E-07
e=0.5e-6	4.49E-07	6.73E-07	8.98E-07	1.12E-06	1.35E-06	1.57E-06	1.80E-06	2.02E-06	2.24E-06
e=1.0e-6	8.66E-07	1.30E-06	1.73E-06	2.16E-06	2.60E-06	3.03E-06	3.46E-06	3.90E-06	4.33E-06
e=2.0e-6	1.74E-06	2.61E-06	3.49E-06	4.36E-06	5.23E-06	6.10E-06	6.97E-06	7.84E-06	8.71E-06
e=3.0e-6	2.59E-06	3.89E-06	5.18E-06	6.48E-06	7.77E-06	9.07E-06	1.04E-05	1.17E-05	1.30E-05

FGSM

Laplacian pairwise Distance Metric (LPDM) Results after PGD poisoning on MNIST
Resulting Deviation measures of LPDM test for different epsilon sizes
1.60E-05
1.40E-05
1.20E-05
1.00E-05
8.00E-06
6.00E-06
4.00E-06
2.00E-06
0.00E+00
LPDM function computer at different gamma values
e=0.1e-6 e=0.5e-6 e=1.0e-6 e=2.0e-6 e=3.0e-6

	gamma = 1.0	gamma = 1.5	gamma = 2	gamma = 2.5	gamma=3.0	gamma = 3.5	gamma = 4	gamma=4.5	gamma =5	gamma = 5
e=0.1e-6	4.42E-09	8.85E-09	1.33E-08	1.77E-08	2.21E-08	2.65E-08	3.10E-08	3.54E-08	3.98E-08	4.42E-08
e=0.5e-6	1.90E-07	3.80E-07	5.70E-07	7.59E-07	9.49E-07	1.14E-06	1.33E-06	1.52E-06	1.71E-06	1.90E-06
e=1.0e-6	4.91E-07	9.81E-07	1.47E-06	1.96E-06	2.45E-06	2.94E-06	3.44E-06	3.93E-06	4.42E-06	4.91E-06
e=2.0e-6	9.99E-07	2.00E-06	3.00E-06	3.99E-06	4.99E-06	5.99E-06	6.99E-06	7.99E-06	8.99E-06	9.99E-06
e=3.0e-6	1.49E-06	2.99E-06	4.48E-06	5.98E-06	7.47E-06	8.97E-06	1.05E-05	1.20E-05	1.34E-05	1.49E-05

PGD

Laplacian pairwise Distance Metric (LPDM) Results after CW poisoning on MNIST
Resulting Deviation measures of LPDM test for different epsilon sizes
1.60E-05
1.40E-05
1.20E-05
1.00E-05
8.00E-06
6.00E-06
4.00E-06
2.00E-06
0.00E+00
Parameter Gamma computed Values of LPDM function
e=0.1e-6 e=0.5e-6 e=1.0e-6 e=2.0e-6 e=3.0e-6

	gamma = 1.0	gamma = 1.5	gamma = 2	gamma = 2.5	gamma=3.0	gamma = 3.5	gamma = 4	gamma=4.5	gamma =5	gamma = 5
e=0.1e-6	2.62E-09	2.85E-09	4.33E-08	1.77E-08	1.21E-08	2.65E-08	3.10E-08	3.54E-08	3.98E-08	4.42E-08
e=0.5e-6	1.90E-07	3.80E-07	5.70E-07	7.59E-07	9.49E-07	1.14E-06	1.33E-06	1.52E-06	1.71E-06	1.90E-06
e=1.0e-6	4.91E-07	9.81E-07	1.47E-06	1.96E-06	2.45E-06	2.94E-06	3.44E-06	3.93E-06	4.42E-06	4.91E-06
e=2.0e-6	9.99E-07	2.00E-06	3.00E-06	3.99E-06	4.99E-06	5.99E-06	6.99E-06	7.99E-06	8.99E-06	9.99E-06
e=3.0e-6	1.49E-06	2.99E-06	4.48E-06	5.98E-06	7.47E-06	8.97E-06	1.05E-05	1.20E-05	1.34E-05	1.49E-05

CW

Fig 4. The deviation measures resulting from LDM test on MNIST's clean and its FGSM, PGD, and CW poisoned dataset pairs respectively

3.3 Results for the lower and upper bounds determined by SPBIA

To estimate the range of deviations resulting for FGSM, PGS and CW attacks, the lower bounds (minimum) and upper bounds (maximum) are determined by SPBIA from the acquired ED values for the MNIST original and poisoned datasets. The minimum deviation observed for FGSM, PGD and CW attacks are 3.20e^{-08}, 8.45e^{-09}, and 2.12e^{-09} respectively. Similarly, the maximum deviations observed for MNIST clean and Poisoned datasets of FGSM, PGD and CW are 1.35e^{-05}, 1.20e^{-05}, and 1.55e^{-05} respectively.

3.4 Results of the Maximum Likelihood Estimation for (τI) derived by SPBIA

The MLE function is applied on the SPBIA outcomes to find the point estimates of lower bound and upper bounds. The outcomes of MLE function are given as follows:

MLE (Lower Bound) (***LB***) = $8.45e^{-9} \pm 0.5e^{-01}$.

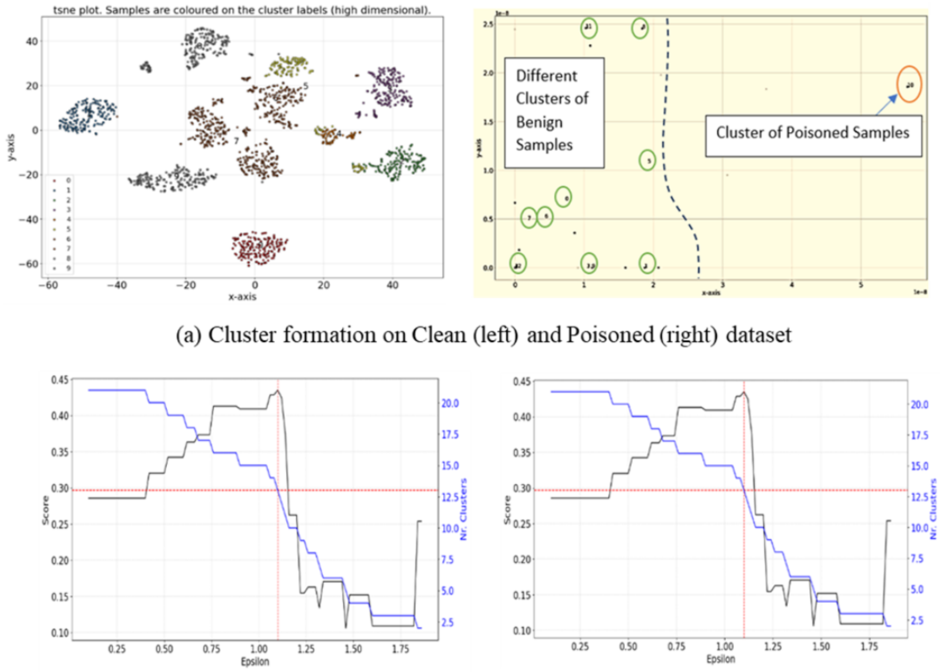
MLE (Upper Bound) (***UB***) = $1.55\ e^{-5} \pm 0.5e^{-01}$.

τI = $[8.45e^{-9} \pm 0.5e^{-01}, 1.55\ e^{-5} \pm 0.5e^{-01}]$.

The τI is the decision factor of MLF-DA which helps determining whether the input dataset is poisoned or not.

3.5 Results of DBSCAN algorithm in splitting the input dataset into similar groups

The cluster formation of the input (non-poisoned and poisoned) dataset to the ML-Filter is shown in Figure 5(**a**). The DBSCAN algorithm employed by the ML-Filter divided the input dataset into similar groups based on the similarity score. 1 to 10 legends in the left plot of Figure 5(**a**) indicating each class of the non-poisoned MNIST dataset. The right plot of DBSCAN clusters formed showing one particular cluster 10 is too far from the other clusters clearly indicate the poisoned dataset's persistence in the input dataset. The silhouette scores of both non-poisoned and poisoned datasets are evaluated to test the efficiency of the DBSCAN algorithm in forming clusters. Figure 5 (**b**) shows silhouette score for DBSCAN performed on non-poisoned and poisoned dataset.



(a) Cluster formation on Clean (left) and Poisoned (right) dataset

(b) Silhouette score plots of Clean (left) and Poisoned (right) datasets

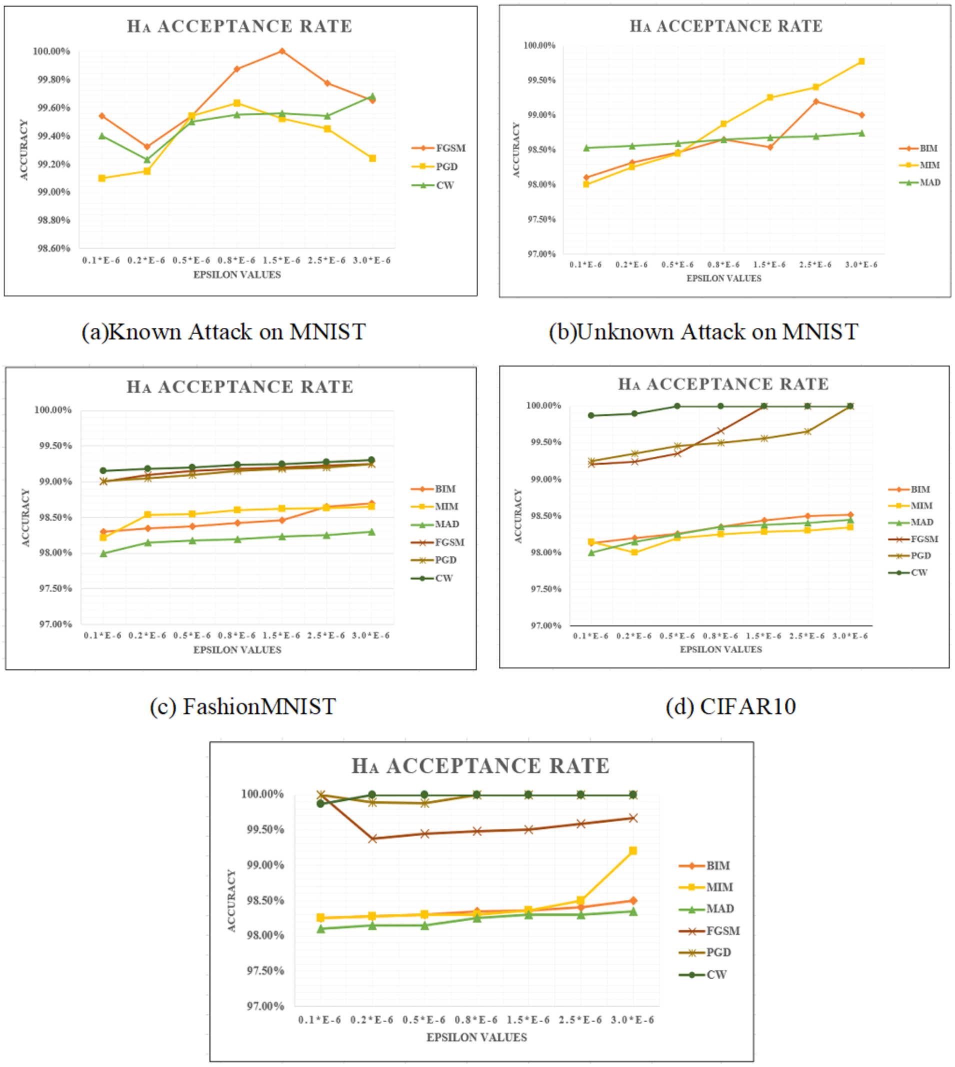
Fig 5. DBSCAN results on non-poisoned and poisoned MNIST datasets (a) cluster formation of non-poisoned and Poisoned dataset (b) Silhouette score for Poisoned dataset

3.6 Results of the Significance test

The significance test mentioned in 3.1.2 is conducted for detection of the known and unknown attack category of datasets listed in the Table 1. The high acceptance rate accuracy of alternative hypotheses indicates that the poisoned samples exist in the input dataset. The test results show that the poisoned datasets were successfully detected with high accuracy of 99.52% on an average for known attacks and 99.22% for unknown attacks as well.

The τI bounds include the deviations of poisoned samples with epsilon sizes 0.1*e^{-6}, 0.5*e^{-6}, 1.0*e^{-6}, 2.0*e^{-6}, and 3.0*e^{-6} respectively. The SPBIA is aware of the deviation measures of the above-mentioned epsilon size perturbations only. But we have tested for the epsilon sizes of 0.2*e^{-6}, 0.8*e^{-6}, and 1.5*e^{-6} which is not known by the SPBIA. Still, most of their deviation measures fall in the τI bounds and thus they are detected by the MLF-DA, substantiating the claim we made for generalized detection.

Figure 6 shows the detention accuracy of MLF-DA detection to the known as well as to unknown attacks of the poisoned datasets. The known attack detection average accuracy for FGSM is 99.48%, 99.47% for PGD, and 99.61% for CW attacks. The unknown attacks' accuracy on MNIST poisoned dataset obtained for BIM is 98.55%, MIM with 99.27%, and 99.14% for MAD attacks. The unknown attack average detection accuracies achieved for the poisoned datasets of FashionMNIST are 99.41%, 99.23%, 99.33%, 98.59%, 98.42%, and 98.34%, CIFAR10 are 99.83%, 99.74%, 99.90%, 98.63%, 99.26%, and 99.12% and CIFAR100 are 99.77%, 99.95%, 99.99%, 99.08%, 99.08%, and 98.95 % for FGSM, PGD, CW, BIM, MIM and MAD attacks respectively.



(a)Known Attack on MNIST

(b)Unknown Attack on MNIST

(c) FashionMNIST

(d) CIFAR10

(e) CIFAR100

Fig 6. Ha acceptance Rates of (a) Known attacks (FGSM, PGD and CW) and (b) Unknown attacks (BIM, MIM and MAD) on MNIST dataset. Unknown attacks of (FGSM, PGD, CW, BIM, MIM, and MAD) on (c) FashionMNIST (d) CIFAR10 (e) CIFAR 100 datasets respectively

3.7 True Positive Rate Evaluation Results

The results of TPR accuracy obtained for BIM, MIM, MAD, FGSM, PGD and CW on MNIST, Fashion MNIST, CIFAR10 and CIFAR100 poisoned sets are shown in Figure 7. The detection algorithm achieves high true positive rates of 99% accuracy for known attack detection and 98.96% for unknown attack detection. The TPR accuracy of the true positives are high to FGSM attack and comparatively less values for CW for on MNIST.

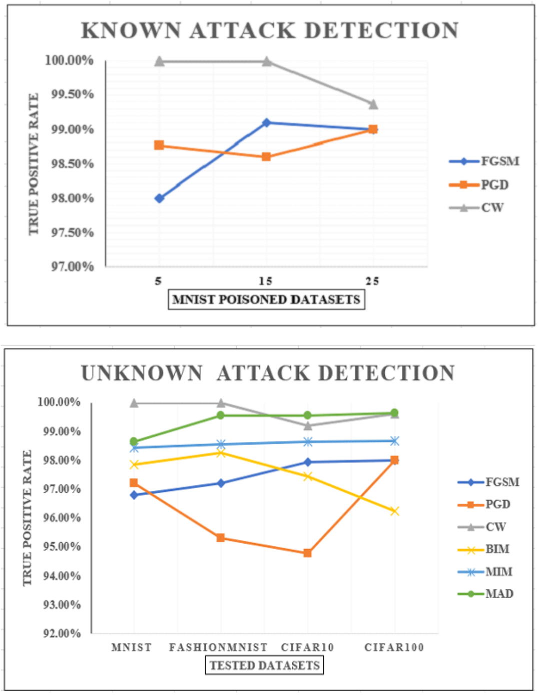


Fig 7. The true positive rate exhibited by the MLF-DA to Known attacks and Unknown attacks

3.8 Performance comparison of MLF-DA with existing methods

In this section, the results of our proposed method are compared with the existing works in the literature. Table 2 presents the result comparison of the existing methods and their detection to different perturbation values (epsilon sizes). The FGSM, PGD and CW attacks when perturbed with epsilon size > 2.0, the methods ED, MMD, ECMF and Mahalanobis distance-based methods successfully identify the attack as these methods learn the attack patterns as a result of AT. But it is clear that the existing methods cannot detect when the epsilon size is less than 2.0. The proposed method is successful in detecting the attacks with all epsilon sizes mentioned in the table except for 3. The reason behind this is due to the SPBIA not considering the epsilon size 3 for deriving bounds. Table 3 shows the comparison of the Mahalanobis distance method with the proposed method MLF-DA. The results show that MLF-DA can successfully detect all the tested poisoned datasets with 98% for unknown attacks and 99.63% to known attacks. The existing methods have the overhead of Adversarial Training whereas the MLF-DA does not have this overhead by excluding the training time.

Table 2. Epsilon values comparison to known and unknown distributions of the proposed and existing methods

Method [Reference]	Type of attack	Epsilon sizes	Detection to Known Attack	Detection to Unknown Attack
ED [8]	FGSM, PGD	2.41,3	✔	X
	FGSM, PGD	0.1 *e-6 to 3.0*e-6	X	X
MMD [8]	FGSM, PGD	2.41,3	✔	X
	FGSM, PGD	0.1 *e-6 to 3.0*e-6	X	X

Continued on next page

Table 2 continued

ECMF [6]	FGSM, PGD	2.11	✔	X
	FGSM, PGD	0.1 *e-6 to 3.0*e-6	X	X
Threshold [3]	FGSM, PGD	2.0, 3.0	✔	X
	FGSM, PGD	0.1 *e-6 to 3.0*e-6	X	X
Mahalanobis [4]	FGSM, CW	2, 3	✔	X
	FGSM, CW	0.1 *e-6 to 3.0*e-6	X	X
MLF-DA (Proposed Method)	FGSM, PGD, CW, BIM, MIM, MAD	1, 2, 0.1 *e-6 to 3.0*e-6, 1, 2	✔	✔

Table 3. Poisonous images detection rate comparison between proposed and Mahalanobis methods

Method	Dataset used for training Model	Datasets	Poisoning Attacks detected	Detection Accuracy	AT Required? Yes/No	GD Possible?
Mahalanobis + GAN [4]	MNIST, CIFAR-10, ImageNet	MNIST, CIFAR-10, ImageNet	FGSM C&W	75.68%	Yes	No
Mahalanobis + ResNet [5]	SVHN, MNIST, CIFAR-10	SVHN, MNIST, CIFAR-10	FGSM	99.32%	Yes	No
Proposed (MLF-DA)	● *	MNIST, CIFAR-10, FashionM-NIST, CIFAR-100	**FGSM, PGD, C&W, BIM, MIM, MAD**	**99.63% (known attack), 98% (unknown attack)**	**No**	**Yes**

*The datasets are not used for training. Instead, only MNIST dataset used for deriving perturbation Bounds.

The earlier Mahalanobis distance-based methods cannot determine whether or not the dataset is poisonous without adversarial training. This is because, the data poisoning attacks create poisoned images with a specific perturbation value called epsilon size. The resulting poisoned image features vary with the epsilon size, the attack type, and the target dataset (in this case, images), and the model used for training them. Due this reason, when there is a variation in the epsilon size, attack type, the target dataset, the resulting image patterns vary accordingly. Hence, the earlier Mahalanobis detection methods are ineffective for malicious instances with distinct characteristics adapted by new attacks and lack generalized detection. Our ML-Filter is based on the statistical deviations caused by the gradient-based poisoning attacks with a wide range of epsilon perturbations. This is the reason why our method is free from probability distributions of datasets, adaptable to new attacks and ML Models as well. Thus, the proposed method ML-Filter achieved generalized detection of known and unknown attacks which substantiates our claim made earlier in this paper. Also, the proposed Secured ML Architecture integrate and leverage the capabilities of ML-Filter to safeguard against gradient-based data poisoning attacks effectively.

4 Conclusion

Securing the Machine Learning models is necessary in the context of adversarial machine learning. The Mahalanobis distance-based methods are dependent on adversarial training and lack generality in detecting new attacks. The proposed method ML-Filter is independent of dataset's probability distribution, type of attack and ML models. The secured architecture, leveraging the attack-agnostic detection capabilities of proposed ML-Filter method successfully identifies the data poisoning attack with 99% TPR to known attacks and 98.96% to unknown attacks. Thus, it achieves generalized detection of unknown attacks without need for adversarial training, substantiating the claims we made earlier in this paper.

In this study only image datasets and CNN architecture were considered. The decision factor of ML-Filter is based on the range of epsilon deviations derived by SPBM and need to refine its bounds to accommodate more detection features to ML-Filter. Also, usage of DBSCAN for splitting the dataset is a time-consuming task. In future more time-efficient method for anomaly detection from the dataset is needed.

References

1) Yerlikaya FA, Bahtiyar Ş. Data poisoning attacks against machine learning algorithms. *Expert Systems with Applications*. 2022;208. Available from: https://dx.doi.org/10.1016/j.eswa.2022.118101.

2) Kumar RSS, Nyström M, Lambert J, Marshall A, Goertzel M, Comissoneru A, et al. Adversarial Machine Learning-Industry Perspectives. In: 2020 IEEE Security and Privacy Workshops (SPW). IEEE. 2020. Available from: https://doi.org/10.1109/SPW50608.2020.00028.

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

4) Uwimana A, Senanayake R. Out of Distribution Detection and Adversarial Attacks on Deep Neural Networks for Robust Medical Image Analysis. 2021. Available from: https://doi.org/10.48550/arXiv.2107.04882.

5) Hou Y, Chen Z, Wu M, Foo CS, Li X, Shubair RM. Mahalanobis Distance Based Adversarial Network for Anomaly Detection. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. 2020;p. 3192–3196. Available from: https://doi.org/10.1109/ICASSP40776.2020.9053206.

6) Macedo D, Ren TI, Zanchettin C, Oliveira ALI, Ludermir T. Entropic Out-of-Distribution Detection: Seamless Detection of Unknown Examples. *IEEE Transactions on Neural Networks and Learning Systems*. 2022;33(6):2350–2364. Available from: https://dx.doi.org/10.1109/tnnls.2021.3112897.

7) Craighero F, Angaroni F, Stella F, Damiani C, Antoniotti M, Graudenzi A. Unity is strength: Improving the detection of adversarial examples with ensemble approaches. *Neurocomputing*. 2023;554:1–14. Available from: https://dx.doi.org/10.1016/j.neucom.2023.126576.

8) Aliyu I, Van Engelenburg S, Mu'Azu MB, Kim J, Lim CG. Statistical Detection of Adversarial Examples in Blockchain-Based Federated Forest In-Vehicle Network Intrusion Detection Systems. *IEEE Access*. 2022;10:109366–109384. Available from: https://dx.doi.org/10.1109/access.2022.3212412.

9) Cinà AE, Grosse K, Demontis A, Biggio B, Roli F, Pelillo M. Machine Learning Security Against Data Poisoning: Are We There Yet? *Computer*. 2024;57(3):26–34. Available from: https://dx.doi.org/10.1109/mc.2023.3299572.

10) Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. Author Correction: SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*. 2020;17:1–1. Available from: https://dx.doi.org/10.1038/s41592-020-0772-5.

11) Lio W, Liu B. Uncertain maximum likelihood estimation with application to uncertain regression analysis. *Soft Computing*. 2020;24(13):9351–9360. Available from: https://dx.doi.org/10.1007/s00500-020-04951-3.

12) Wang Y, Li F, Sun H, Li W, Zhong C, Wu X, et al. Improvement of MNIST Image Recognition Based on CNN. In: 7th Annual International Conference on Geo-Spatial Knowledge and Intelligence ;vol. 428 of IOP Conference Series: Earth and Environmental Science. IOP Publishing. 2020;p. 1–8. Available from: https://dx.doi.org/10.1088/1755-1315/428/1/012097.

13) Zhang X, Tan H, Huang X, Zhang D, Tang K, Gu Z. Adversarial Attacks on ASR Systems: An Overview. 2022. Available from: https://doi.org/10.48550/arXiv.2208.02250.

14) Xhaferra E, Cina E, Toti L. Classification of Standard FASHION MNIST Dataset Using Deep Learning Based CNN Algorithms. In: 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE. 2022;p. 494–498. Available from: https://doi.org/10.1109/ISMSIT56059.2022.9932737.

15) Barz B, Denzler J. Do We Train on Test Data? Purging CIFAR of Near-Duplicates. *Journal of Imaging*. 2020;6(6):1–8. Available from: https://dx.doi.org/10.3390/jimaging6060041.

16) Hirano H, Takemoto K. Simple Iterative Method for Generating Targeted Universal Adversarial Perturbations. *Algorithms*. 2020;13(11):1–10. Available from: https://dx.doi.org/10.3390/a13110268.

17) Mohandas S, Manwani N, Dhulipudi DP. Momentum Iterative Gradient Sign Method Outperforms PGD Attacks. In: Proceedings of the 14th International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and Technology Publications. 2022;p. 913–916. Available from: https://www.scitepress.org/Papers/2022/109384/109384.pdf.

18) Goldblum M, Tsipras D, Xie C, Chen X, Schwarzschild A, Song D, et al. Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023;45(2):1563–1580. Available from: https://dx.doi.org/10.1109/tpami.2022.3162397.

19) Gao L, Zhang Q, Song J, Liu X, Shen HT. Patch-Wise Attack for Fooling Deep Neural Network. In: European Conference on Computer Vision – ECCV 2020;vol. 12373 of Lecture Notes in Computer Science. Springer, Cham. 2020;p. 307–322. Available from: https://doi.org/10.1007/978-3-030-58604-1_19.

20) Chiang PY, Geiping J, Goldblum M, Goldstein T, Ni R, Reich S, et al. Witchcraft: Efficient PGD Attacks with Random Step Size. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. 2020;p. 3747–3751. Available from: https://doi.org/10.1109/ICASSP40776.2020.9052930.