# Classification Auto-Encoder Based Detector Against Diverse Data Poisoning Attacks

Fereshteh Razmi and Li Xiong(✉)

Emory University, Atlanta, GA 30322, USA
{frazmim,lxiong}@emory.edu

**Abstract.** Poisoning attacks are a category of adversarial machine learning threats in which an adversary attempts to subvert the outcome of the machine learning systems by injecting crafted data into training data set, thus increasing the resulting model's test error. The adversary can tamper with the data feature space, data labels, or both, each leading to a different attack strategy with different strengths. Various detection approaches have recently emerged, each focusing on one attack strategy. The Achilles heel of many of these detection approaches is their dependence on having access to a clean, untampered data set. In this paper, we propose CAE, a Classification Auto-Encoder based detector against diverse poisoned data. CAE can detect all forms of poisoning attacks using a combination of reconstruction and classification errors without having any prior knowledge of the attack strategy. We show that an enhanced version of CAE (called CAE+) does not have to rely on a clean data set to train the defense model. The experimental results on three real datasets (MNIST, Fashion-MNIST and CIFAR-10) demonstrate that our defense model can be trained using contaminated data with up to 30% poisoned data and provides a significantly stronger defense than existing outlier detection methods. The code is available at https://github.com/Emory-AIMS/CAE

**Keywords:** Data Poisoning · Anomaly Detection · Autoencoder

## 1 Introduction

Poisoning attacks are attacks at *training time* [6] in which an attacker manipulates a small fraction of the training data in order to corrupt the model. Consequently, the model may learn a significantly different decision boundary, resulting in drastic test error. Poisoning attacks are acquiring increasing importance in emerging crowd-based systems that collect data from outside sources [12,13,37]. In crowd-sourcing platforms, the attacker can cause massive damages without having a direct access to the system, but rather by poisoning the collected data from her. A few examples are autonomous driving cars, health systems, online

This work was funded by National Science Foundation (NSF) CNS-2124104.

Published by Springer Nature Switzerland AG 2023
V. Atluri and A. L. Ferrara (Eds.): DBSec 2023, LNCS 13942, pp. 263–281, 2023.
https://doi.org/10.1007/978-3-031-37586-6_16

review systems, and malware/spam detection systems. Also, recently poisoning attacks have been widespread in federated learning systems [32,33,36].

The most recognized poisoning attacks are *label flipping* and *optimal attacks* [7,41]. In these types of attacks, according to the attacker's goal and his accessibility to the data, he may change the labels of some training samples or distort the feature space of the samples, usually in an optimal way to diverge the training from its regular path. Another class of poisoning attacks is backdoor attacks in which the attacker only targets a group of test data that include a specific backdoor trigger [8,16]. Backdoor attacks are not explored in this paper, but they can be considered as future work.

Several defense methods have been recently developed to address flipping or optimal poisoning attacks. Most of them consider poisoned points as outliers and utilize outlier detection techniques. They can be based on k-Nearest Neighbor (kNN) algorithms that consider a point with contrasting label with nearby samples as a poison [30]. They can determine whether a point is poisoned by comparing its distance to a nearby point or other data points in its cluster [22,29]. However, they have several limitations. First, they may only work for a particular type of attacks (optimal or flipping) as the detection is based on the change of either labels or features. Second, they rely on purely clean data to learn the patterns of normal points. Training on tainted data is plausible only when the fraction of the anomalous data is negligible. Also they usually rely on a threshold to determine an outlier.

A potential solution for outlier detection that has not been explored for data poisoning attacks is auto-encoder based method [4,38] which learns the data representation in an unsupervised way. It has been utilized for generating poisoning attacks [10,14,45], anomaly detection [31,49], and adversarial example detection [26]. While promising, utilizing auto-encoders for detecting poisoned points under poisoning attacks present several challenges. First, existing methods train auto-encoders using clean data while there is no guarantee of purely clean data under poisoning attacks [11,24]. Second, existing methods typically select a threshold by allowing certain percentage of clean points to pass (e.g., 90% clean data) but there is no access to such clean data under poisoning attacks. Finally, existing methods for detecting adversarial examples during inference time only utilize feature space (adversarial examples do not have labels). Thus, if directly applied for detecting poisoning attacks, they overlook some essential aspects of the attacks, i.e., the labels of the poisoned data (they may be flipped).

**Contributions**. In this paper, we develop a Classification Auto-Encoder based detector (CAE) that utilizes both feature space and label (class) information to defend against diverse poisoned data. We use a Gaussian Mixture Model for discriminating poisoned points from clean data so that it does not require any explicit threshold. We further propose an enhanced version of our method (CAE+) which does not require purely clean data for training. We elaborate our contributions as follows:

- We develop a classification auto-encoder based detector (CAE) to defend against diverse data poisoning attacks, including flipping and optimal attacks.

The key idea is to utilize two components, an auto-encoder based reconstruction for learning the representation of the data from the feature space and an auxiliary classifier for incorporating the label (class) information into the data representation so it can better detect the poison points.

- We further propose an enhanced model CAE+ so that it can be trained even on partially poisoned data. The key idea is to add a reconstruction auto-encoder (RAE) with CAE to form a joint auto-encoder architecture combined with early stopping of CAE so that it does not overfit the poisoned data while still learning useful representations of the clean data.
- We evaluate our method using three large and popular image datasets and show its resilience to poisoned data and advantage compared to existing state-of-the-art methods. Our defense model can be trained using contaminated data with up to 30% poisoned data and still works significantly better than existing outlier detection methods.

## 2 Background and Related Work

### 2.1 Poisoning Attacks

Assume distribution $R$ on $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{Y} = \{-1, 1\}$. For a clean training dataset $D_{tr} = \{(x_i, y_i) \subseteq R\}_{i=0}^{n_{tr}}$, the goal of a binary classification task $\mathcal{M}$ parameterized by $\boldsymbol{w}$ is to minimize objective loss function $\mathcal{L}(D_{tr}, \boldsymbol{w})$, w.r.t its parameters $\boldsymbol{w}$. In a poisoning attack, the attacker's goal is to produce $n_p$ poisoned data points $D_p = \{(x_i, y_i) \subseteq R\}_{i=0}^{p}$ so that using new training data $D'_{tr} = D_{tr} \cup D_p$ by the learner results in attacker's goal or objective function. This goal can be maximizing the loss on the entire clean test dataset (untargeted attacks) or on a subset or class of them (targeted attacks). As a result the classification accuracy of the entire clean test data or a subset of them will drop drastically.

Poisoning attacks have different manifestations depending on which part of the data is manipulated during the attack [39]. Each of them can have a different impact on attacker's objective function and different attack strength. In *Label flipping attacks* or in short flipping attacks, only class labels of poisoned data are flipped, and the adversary usually has a limited budget for the number of samples it is allowed to change their labels [30,41,43,46].

*Optimal attacks* are based on optimizing the poisons to maximizing the degradation of the model's performance. These attacks are stronger compared to other poisoning attacks, since both feature space and labels can be changed. For classification problems [7,27,42], the rule of thumb is to initialize poisons with real samples from training data set and flip their labels. Since labels are not differentiable, they only optimize the feature space. For image datasets, the input space refers to the pixels comprising the image.

In this paper, we also introduce *Semi-optimal attacks* which keep the original labels of the points without flipping them and only optimize the feature space. This attack can be realistic when the attacker has no control over the labeling process. The distinction between the different attacks are shown in Table 1.

**Table 1.** Various types of poisoning attacks based on tampering different input domains of the initial candidate poisoning points.

| Domain | Attack | | |
|---|---|---|---|
| | Flipping | Semi-optimal | Optimal |
| $\mathcal{X}$ | - | ✓ | ✓ |
| $\mathcal{Y}$ | ✓ | - | ✓ |

### 2.2 Defense Against Poisoning Attacks

**Outlier Detectors**. Outlier detection methods are common to defend against poisoning attacks. These methods are based on the fact that poisoned data deviates from normal points or underlying data generation mechanism. Paudice et al. [29] suggest distanced-based outlier detection methods to mitigate the effect of optimal attacks, assuming they have access to a trusted dataset to train the outlier detector. Steinhardt et al. [35] use a centroid-based outlier detector and estimates a data-dependent upper-bound on the objective loss for poisoning attacks, offering a certified defense. Chen et al. [11] benefit from combining Generative Adversarial Network (GAN) based models, namely cGAN and WGAN-GP, to create augmented clean samples from a trusted dataset and mimic the original model. It then compares the training data against a threshold calculated by the augmented clean data to detect the poisoned samples. Outliers can also be detected by clustering based methods [34]. Laishram and Phoha [22] consider clustering in the combination of feature and label space to defend against optimal attacks and showed that the poisoned points are more separated from the rest of their cluster compared to using features alone. K-nearest neighbor algorithm is proposed to combat flipping attacks [30]. They assume samples close together share a common label; otherwise, the sample's label is highly likely flipped.

As we discussed, the existing outlier detection based methods have several limitations. They typically focus on one type of attacks and rely on purely clean data to train the detector and a threshold to determine outliers and hence are not very effective or robust. Furthermore, a recent work [20] considers a new attack method that generates adjacent poisoned samples. In this case, proximity-based outlier detection algorithms such as K-nearest neighbor fail to recognize poisoned data.

**Contribution-Based Methods**. Another type of defense methods are based on how and to what extent each point contributes to or influences the resulting model. In the context of regression problems, Jagielski et al. [18] retrain the model multiple times and removes points with high residuals as poisonous points. RONI [28] is another defense method that tests the impact of each data point on training performance and discards those points that have a negative contribution. Baracaldo et al. [5] takes a similar approach but reduces computational expenses by examining the impact of entire group of points on the model in order to find manipulated groups. [19] is another work that attempts to find the high impact samples in a less costly process using influence functions. Hong et al. [17] shows poisoned data impact the magnitude and orientation of gradients during the training. They use DP-SGD [1], a training method for achieving differential privacy [48], to add noise to the gradients and limit their sensitivity to mitigate the impact of poisoned samples. In summary, while showing promising results without approximation, the contribution-based methods have the main drawback of high computation cost due to enumerative retraining, which makes it impractical, especially for settings where potentially poisonous data are being continuously acquired. Also, for the differentially private defense methods, they suffer from a decrease in model utility [17].

### 2.3 Auto-Encoders in Anomaly Detection

Auto-encoders [4,38] are neural networks that learn data representation in an unsupervised way. Auto-encoders reconstruct input $x$ into output $x'$ by minimizing reconstruction error usually on an $L_p$-norm distance:

$$RE(x) = \| x - x' \|_p \tag{1}$$

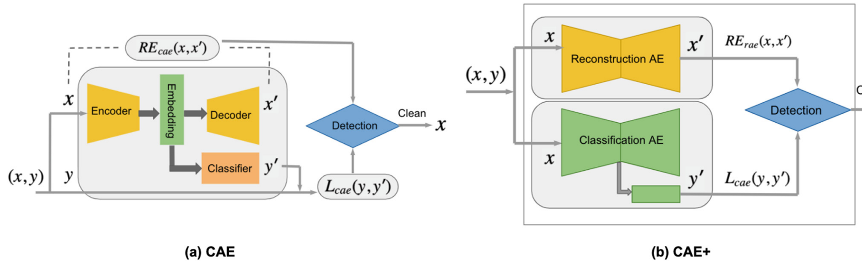
If auto-encoders are trained with only benign data, they learn to capture only the main characteristics of these data. So when the reconstruction error of a sample exceeds a threshold, it is considered an anomaly [2,31]. Nevertheless most anomalies are recognized as samples with observable differences from the real data [3], which are not effective for poisoned data that have very small perturbations.

Auto-encoders have been proposed to detect adversarial examples at inference time by Magnet [26]. In addition to considering reconstruction error between the input and output, they also feed them to the target classifier and compare the corresponding softmax layer outputs to boost the detection power. However, in the context of poisoning attacks, a pre-trained trusted classifier does not exist. Instead the defender has access to an extra piece of information which is the associated label of the poisoned point.

When used for both outlier detection and adversarial example detection, the auto-encoders need to be trained with pure clean data to capture shared properties amongst normal data [2,26]. Even in some works [3,47] that considered anomalous data in the training process of the auto-encoders, the percentage of anomalies in the dataset is insignificant. In the setting of poisoning attacks, the assumption of having a clean dataset for training the defense method is not realistic. By utilizing a joint architecture, we show that our defensive model can remain resilient to poisoning attacks even up to 30% poisoned points.

## 3 Classification Auto-Encoder Based Detector

As a baseline solution, we can train an auto-encoder on feature space as in existing outlier detection methods. For a clean sample $s_c = (x_c, y_c)$, and a poisoned



**(a) CAE**

**(b) CAE+**

**Fig. 1.** Auto-encoders Structure: (a) The structure of Classification Auto-encoder (CAE). If trained on pure clean dataset it provides a high success defense against all poisoning attacks. (b) The structure of CAE+. Both Reconstruction Auto-encoder (RAE) and Classification Auto-encoder (CAE) work together to combat against poisons. This joint structure makes the defense method more robust even if trained on a contaminated dataset.

sample $s_p = (x_p, y_p)$, the reconstruction error $RE(x_p)$ can be used to discriminate $x_c$ and $x_p$:

$$RE(x_c) \ll RE(x_p) \tag{2}$$

According to (2) any data point with significantly large reconstruction error (greater than certain threshold) can be considered as a poison. The limitation of this approach is that it will only capture the changes in the feature space. Hence it will address only semi-optimal attacks which only change the features.

### 3.1 Classification Auto-Encoder (CAE)

To defend against all types of poisoning attacks, we need a method that incorporates both labels and features in detection process. In other words, the latent encoding of the auto-encoder needs to reflect the label information.

**Classification Auto-Encoder.** We propose Classification Auto-Encoder (CAE) which has an auxiliary classifier fed by the latent representation $z$ of the encoder (as shown in Fig. 1(a)). If $RE_{cae}$ indicates the reconstruction error, and $L_{cae}$ indicates the auxiliary classifier's loss on representation layer $z$, CAE is trained to minimize $\sum_{x_i}(RE_{cae}(x_i) + L_{cae}(x_i)))$ on training dataset $D_c = \{x_i, y_i\}_{i=0}^{n}$. As a result, $z$ is learned in such a way that the classifier is able to predict the label, and the decoder can reconstruct the associated input. To boost the connection between these two tasks, we train the auxiliary classifier and the decoder simultaneously. It contrasts with previous works that utilize classification auto-encoders for predictive or classification objectives. They employ a two-stage training process which trains the pair of encoder-decoder and then uses the low-dimensional representation for training the classifier [15,44].

**Detection Criteria.** Once the CAE is trained, given a data point, we can use the combined reconstruction error and classification loss as a detection criteria for poisoned data (e.g., if it is greater than certain threshold), since it considers deviations in both feature space and label space.

$$Error(x) = \alpha.RE_{cae}(x) + (1 - \alpha).L_{cae}(x) \tag{3}$$

The first term $RE_{cae}(x)$ is the reconstruction error of CAE and the second term $L_{cae}(x)$ is the loss of the CAE auxiliary classifier. $\alpha$ and $1 - \alpha$ are weights to control the effect of each term. Since $RE(x)$ is indicative of changes in $x$, and $L(x)$ reflects the classification loss, the combined metric $Error(x)$ can detect both changes in feature space and labels and hence defend against the different types of attacks.

In general, a threshold can be defined based on *a guess on the number of possible poisoned points* $K$ [18]. Tuning $K$ is a difficult job that makes the detector very sensitive to the actual fraction of poisoned data. Instead, we use a clustering approach and cluster all points based on $Error(x)$ into two clusters using a Gaussian Mixture Model (GMM). We show that the error is so distinct between clean and poisoned points that GMM can separate it very well into two clusters, each representing clean or poisoned data.

### 3.2 Enhanced Classification Auto-Encoder (CAE+)

CAE requires clean data for training the auto-encoder so it can learn the structure of the normal data and detect any deviation from that. Since we assume the training data is poisoned, we need to add a mechanism that is robust to contaminated data. We do so by leveraging a combination of early stopping and a replicate reconstruction auto-encoder.

**Early Stopping**. Since we assume there is no access to purely clean data for training the detector, to prevent CAE to learn patterns from poisoned data, we use the early stopping method. Early stopping leads the auto-encoder to focus on reconstructing the pattern of the majority of data, and avoids overfitting on anomalies. The auxiliary classifier is a single dense layer and can usually catch all the class information quickly, especially in binary-class problems. Selecting a small number of neurons in this layer does not provide sufficient parameters for the classification task, and leads to missing even the general patterns of the training dataset. On the other hand, large number of neurons makes the classifier more complex and may overfit the poisonous data. To capture all the information and avoid underfitting, we can select a fairly large number of neurons and address the overfitting problem using early stopping.

By using this approach, CAE can be very robust to the poisoned data. However, at the stop point of the training process, $z$ has captured those patterns of the data that help mostly with classification, but not the reconstruction (which takes longer to learn). Hence we propose a joint auto-encoder architecture to address this challenge by using a parallel reconstruction auto-encoder (RAE).

**Reconstruction Auto-encoder**. The Reconstruction Auto-Encoder (RAE) is a replicate of the encoder-decoder part of CAE without the auxiliary classifier (as shown in Fig. 1(b)). RAE is trained to minimize the reconstruction error only. By having these two auto-encoders, for an input $\{x, y\}$ we calculate the following combined error:

$$Error(x) = \alpha.RE_{rae}(x) + (1 - \alpha).L_{cae}(x) \tag{4}$$

This is a modification to (3), in which the reconstruction error has been replaced with reconstruction error of RAE ($RE_{rae}(x)$).

This extra auto-encoder helps us adjust the training process for RAE separately so that while RAE can be trained to full capacity, CAE with early stopping is not overfitting the poisonous data. In comparison to the classifier of CAE, RAE with high capacity (especially with convolutional layers) can be trained with a high number of epochs without overfitting the poisoned data. We call this joint structure of CAE and RAE, CAE+, since it is enhancing the CAE functionality.

In practice, the training data may be poisoned, so using CAE+ and Eq. 4 is required. In Sect. 4, we investigate potential scenario of having a clean training dataset $D_c$ and compare CAE vs. CAE+. In the case of clean training data, since the concern of overfitting poisoned data does not exist, CAE can be trained until both the classification layer and decoder converge. We show that CAE can be effective under this circumstance. In contrast, when training data is poisoned, we show that CAE+ is much more robust.

**Overview**. Combining all the previously mentioned information, the overview of the detection mechanism can be summarized as follows: Initially, we train the CAE+ detector using the available poisoned data. Subsequently, we utilize the output of this detector to create two separate clusters for GMM: one for normal data and the other for poisoned data. Afterwards, we input the suspected points to the combined models of the trained CAE+ and GMM. If a point falls within the clean cluster, it is utilized for training the target classifier.

## 4 Experiments

In Sect. 4.1, we describe the details of our experimental settings, including the datasets, the attacker's target model, the architecture of our detectors, the comparison methods, and the attributes of the attacks. We also evaluate mixed attacks that combine all types of poisoning attacks to show the strength of CAE+ against diverse attacks. Furthermore, we clarify how we used the periodic update of the model to mimic real scenarios wherein poisoning attacks occur.

In Sect. 4.2, we depict the impact of each type of attack on the poisoned data under different defense methods, the prominence of the Gaussian Mixture Model (GMM) over threshold selection, and the effect of the different auto-encoders employed in the CAE+. Then an ablation study reveals the benefit of CAE+ over CAE and RAE. To confirm the superiority of CAE+, it is compared to the other state-of-the-art detectors in the literature on multiple datasets, including MNIST, Fashion-MNIST, and CIFAR-10. Finally, we evaluate the robustness of CAE+ and CAE under a hypothetical setting in contrast to the previous experiments, when there is a trusted training dataset for training CAE.

### 4.1 Experimental Setup

**Datasets**. First, we evaluate the performance of CAE+ using the MNIST dataset [23], and more challenging Fashion-MNIST dataset [40] on binary subproblem classes: MNIST 9 vs. 8 and 4 vs. 0, and Fashion-MNIST Sandal vs. Sneaker and Top vs. Trouser. It is common practice to apply binary setting for data poisoning attacks [7,19]. Second, we conduct experiments on a more complex dataset CIFAR-10 [21] for two randomly chosen classes Airplane vs. Automobile. All datasets are normalized within the interval [0, 1]. These three datasets are frequently employed in the literature to evaluate poisoning attacks [7,9] and offer varying levels of complexity.

**Attacks**. Support Vector Machines (SVM) are known to be subject to strong poisoning attacks [7,41]. In contrast to complicated models and neural networks [27], poisoning attacks can achieve a high success in dropping the accuracy of SVM. As we will show in Fig. 6, the accuracy of optimal attacks on the SVM model drops to 60% with 10% of poisons. Hence, we use poisoning attacks against SVMs in the experiments to better demonstrate and evaluate the effectiveness of different defense methods. We use linear kernel for MNIST and Fashion-MNIST and RBF kernel for CIFAR-10. We note that our methods work on poisoning attacks against any target models such as neural networks.

We compare four types of attacks; flipping, optimal, semi-optimal, and mixed attacks, then assess our defense model against them. In a mixed attack, the attacker selects 1/3 of the poisons from each of the aforementioned attack types. This way, we can challenge the defender's ability to detect diverse poison simultaneously, despite their different characteristics. The optimal attack is conducted based on [25] with some modifications.

**Setup**. A common paradigm for training machine learning models in real world is the periodic update [22] in which the data is acquired continuously. In this scenario, data is provided by users and buffered until sufficient data is obtained to retrain the model. To implement such a periodic update setting for SVM classifier, we consider 60 rounds of SVM updates. Each round represents a new batch of data which consists of 500 data points divided into a training set, a validation set, and a test set of 100, 200, and 200 samples, respectively. Based on different attack types, the attacker generates poisoned points for each round and adds them into the training data for that round. At the next step, we assume that the defender has access to the recent 50 rounds of buffered data. By aggregating the contaminated buffered data of those 50 runs, we train our defense model. Then for evaluation purposes, we use the remaining 10 rounds of updates for testing the defense methods, namely 10 times the buffered data is fed to the detector and the data passing through it is used for model assessment. Every result reported in this paper is the average of these 10 test runs. Maintaining a ratio of 5/1 between training and testing the defense method guarantees a sufficient amount of data is available to evaluate the effectiveness of the defense method.

Note that for each of the attacks unless otherwise specified, up to 10% of the clean data are poisoned. In practical scenarios, attackers typically have limited access to data, usually restricted to a small percentage. Therefore, exceeding 10% may not be realistic [7,20,22]. We believe this is high enough to validate the robustness of CAE+ against poisonous data. To further show the impact of the percentage of the poisoned data, we conduct the experiment on CIFAR-10 with a higher poisoning rate (up to 30%).

**Implementation Details**. The structure of CAE reconstruction component and RAE is inspired by the auto-encoders introduced in Magnet [26] with some modifications. Our reconstruction auto-encoders, for MNIST and Fashion-MNIST dataset, consist of $3 \times 3$ convolutional layers in the encoder, each composed of 3 filters of size $3 \times 3$ with $1 \times 1$ strides and sigmoid activations. Between these two convolutional layers a MaxPooling $2 \times 2$ is located. At the decoder, the structure of convolutional layers are the same as the encoder. The only difference is that the MaxPooling layer is replaced with a 2D UpSampling layer. As the last layer of the decoder we have a third $3 \times 3$ convolutional layer with only one filter (compatible to number of channels in MNIST and Fashion-MNIST) to reconstruct an output image with the same size as the input image. Also, as [26] suggests, we use a slightly different architecture for CIFAR-10, by utilizing only one convolutional layer in the encoder and one in the decoder with the mentioned parameters. For the auxiliary classifier, the encoder's output is flattened and fed to a dense classification layer with size 128. We experimentally found out that dropping out the data with rates 0.25 and 0.5 before and after the dense layer serves the best in training the model and reduces the overfitting. For each dataset, we train CAE for 100 epochs and the RAE for 300 epochs with a batch size of 256 using the Adam optimizer. The aggregated error $Error(x)$ is calculated based on Equation (4) on weighted sum of the normalized $L_1$-norm reconstruction error and the auxiliary classifier's cross entropy loss.

**Comparison Methods**. Distance-based outlier detectors are state-of-the-art methods in defending against poisoning attacks [20,29]. One of their interesting properties is that they are more robust against poisoned data and do not require to be trained on a purely clean dataset compared to other outlier detection based methods. So, similar to [29], we select **centroid-based Outlier Detectors (OD)** as the baseline. It first finds the centroids of each class in the training dataset and then discards the points that are distant from their respective class centroid.

Furthermore, we compare our method to a modified Magnet, a state-of-the-art auto-encoder based detector designed for adversarial examples [26]. We note that there are more recent detection approaches against adversarial examples, however, our goal is not to defend against adversarial examples, but rather

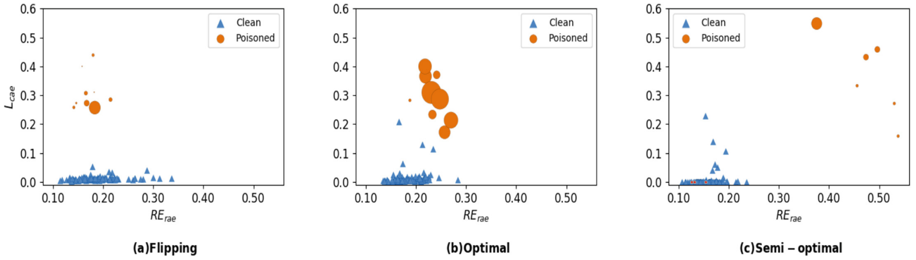


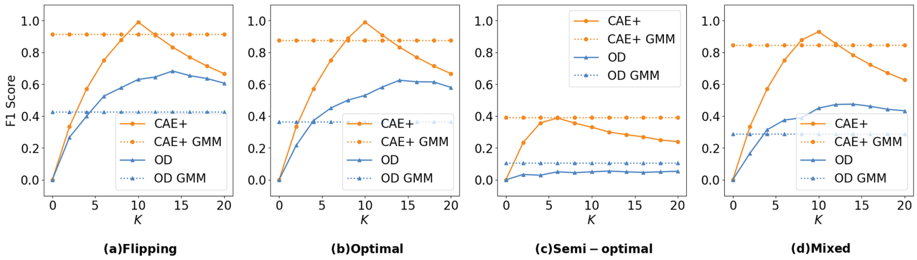
**Fig. 2.** The effect of different attack types on the reconstruction error and auxiliary classification loss for poisoned MNIST-4-0 dataset. Triangles and circles represent clean and poisoned points, respectively. The poisons' size represents their impact on degrading the SVM accuracy (larger circles indicate higher impact).

to adapt the auto-encoder based detector as a potential solution for poisoning attacks as a comparison. We make the following modifications in order to make it compatible with poisoning attacks under our setting. We train Magnet on the same poisonous data as the other defense methods. It contrasts with the original paper in which the authors train Magnet on a thoroughly clean dataset. The assumption of access to a clean dataset is valid under adversarial example attacks at inference time, but not under the poisoning attacks during training time. We use the same structure as the original paper suggests [26], the only hyperparameter we change is the number of epochs for a better adaptation to poisoning attacks (from 100 epochs to 300 epochs). In addition to the detector, we also evaluate the performance of Magnet detector paired with a reformer [26]. In this case, after Magnet detector filters out poisons, it passes the remaining data through the reformer, which is another auto-encoder. The reformer's reconstructed output will replace the original input and then be fed to the classifier.

### 4.2 Results

**Effect of Different Attacks**. As we discussed in Sect. 3, each type of poisoned data can have a different impact on CAE+ components. Figure 2 illustrates this fact by showing the classification error $L_{cae}$ and reconstruction error $RE_{rae}$ of the different poisoning attacks on MNIST-4-0. Blue triangles and orange circles represent the clean and poisoned points, respectively. Clean data is the same for all three plots. For the poisoned data, the size of circles indicates their importance in degrading the SVM classification results. Larger circles imply that the insertion of those poisons to the SVM clean training dataset drops more accuracy.

For the flipping attack, the reconstruction error $RE_{rae}$ cannot differentiate the poisoned samples from the rest of the data since the feature space of the poisons is intact, while the classification loss $L_{cae}$ is much larger for the poisoned data. Under the optimal and semi-optimal attacks, the transformations that occur in the feature space discriminate the clean data and the poisons through $RE_{rae}$. It is more noticeable for the semi-optimal attack because the features



**(a)Flipping** **(b)Optimal** **(c)Semi – optimal** **(d)Mixed**

**Fig. 3.** Changes on MNIST-4-0 F1-score over different thresholds for CAE+ and OD. Thresholds are guesses on the probable number of poisoned data within the training dataset.

alter more drastically than in the optimal attack. This discrepancy between the poisons' features and the clean space impacts their classification results and increases the loss $L_{cae}$. Therefore, as Eq. 4 suggests, a mixture of both reconstruction and classification errors is required to detect diverse attacks in the context of an attack-agnostic defense.

**Threshold vs. GMM**. According to Sect. 3.1, we pass the detectors' output to a GMM for clustering the data into poisoned and clean data, so that we do not need to specify a threshold of possible poisoned points $K$ for filtering poisons. We compare our GMM-based approach with the baseline threshold approach when a fixed number of training data is poisoned (about 10% of the training data, i.e., 10 poisons). We report **F1-score** for the detection, which is the harmonic mean of the precision and recall with the best value at 1. F1-score is indicative of how successful a detector is in filtering poisons and passing clean data. An ideal detection algorithm can identify all and only poison data, which means a perfect F1-score.

Figure 3 depicts how the detectors' F1-scores change with different threshold of $K$ for MNIST-4-0 (solid lines). For flipping, optimal and mixed attacks, the F1-score of CAE+ hits almost 1 at $K = 10$. In other words, it can accurately detect all ten poisoned points with very few false positives. The V shape of CAE+ plots depicts its sensitivity to an accurate threshold $K$. Before threshold 10 there are naturally some false negatives, and after that point, false positives are emerging. In contrast, we do not need to specify any threshold in the unsupervised GMM method (dashed line) for CAE+. We can see that it competes very closely with the best guess on $K$ in the threshold-based method.

For the semi-optimal attacks, the scenario is slightly different. The majority of the poisoned points in semi-optimal attacks get stuck in local maxima and do not change their feature space; hence they have little impact on the attack. For the same reason, they do not harm the accuracy even though they can not be filtered out. This fact is illustrated in Fig. 2. Some of the low-impact attacked points (shown with small circles) are placed at the bottom left corner of the plot, where the majority of the clean data points are located. As a result, in Fig. 3,

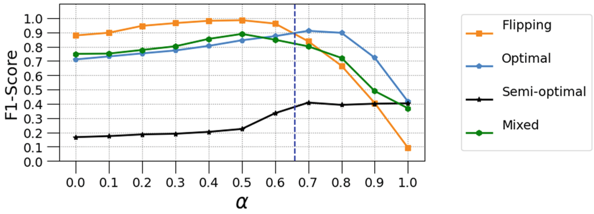


**Fig. 4.** CAE+ F1-score for different values of $\alpha$ (Eq. 4)

F1-score for semi-optimal attacks is not high; but we show later that CAE+ can detect all the high impact attack points and achieve the original SVM's accuracy.

In all the attacks, for both threshold-based and GMM methods, CAE+ yields significantly better F1-scores than OD. For linear SVM, overlooking poisoned points can be much more harmful than filtering out clean data. So despite the high false-positive rate, OD can still partially enhance the SVM accuracy. OD completely fails to operate as a detector if the system is sensitive to clean data removal. In the remaining experiments, we leverage GMM for all the detection approaches (including the baseline OD) to have a fair comparison of how they boost SVM accuracy.

**Impact of Alpha**. There are four types of attacks. Each of the CAE+ reconstruction or classification auto-encoders is suitable to address different attack types. Coefficient $\alpha$ in Eq. 4 can be adjusted to meet this goal. Since the attacker's attack type is not known to the defender, $\alpha$ should be pre-adjusted considering all the attack types. Figure 4 demonstrates how different values of $\alpha$ affect F1-score. Reconstruction error has a significant impact on semi-optimal attacks, and as a result, higher $\alpha$ boosts the F1-score. In flipping attacks and optimal attacks, classification error gains more importance. In particular, in optimal attacks, there is a trade-off between reconstruction error and classification error. The vertical dashed line shows $\alpha$=0.66 in which every attack sustains high F1-socre. According to Eq. 4, at this value the coefficient of $RE(x)$ is twice as the coefficient of $L(x)$.

**Ablation Study**. In this section we show the contribution of each component in CAE+ (recall Fig. 1). We train two additional models for comparison: 1) CAE that is not combined with the RAE (the bottom auto-encoder in Fig. 1(b)) and has the error function in Eq. 3; 2) RAE that is a stand-alone reconstruction auto-encoder (the top auto-encoder in Fig. 1(b)) and uses reconstruction error as defined in Eq. 1.

The error for CAE is calculated based on Eq. 3, and for RAE, it is limited to just reconstruction error. Note that all these methods are trained with 10% contaminated data and paired with GMM. Figure 5 shows the effectiveness of these detectors based on F1-score. Since RAE considers only feature space, it is effective on semi-optimal attacks and, to a less extent, on optimal attacks.

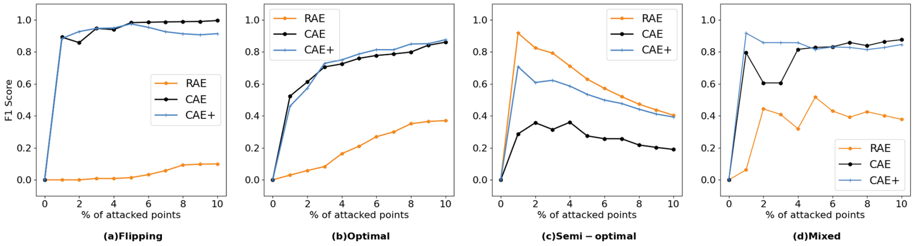


**Fig. 5.** Ablation study between CAE+, CAE and RAE on MNIST 4-0

However, flipping attacks can evade it. On the other hand, CAE relies on classification and reconstruction errors with more emphasis on classification loss. So it fails on semi-optimal attacks. CAE+ has the advantage of using both CAE classification error and RAE reconstruction error, and as a result, it gains a better F1-score on average. Since the attack is not known in advance, CAE+ is the best detector among these three.

**Comparison**. In this experiment, we compare the performance of CAE+ in terms of accuracy of the resulting model with state-of-the-art defense methods. We feed the learner's training data into detectors and filter suspicious poisoned points using GMM. The rest of the points are used to retrain the SVM classifier. A perfect filter leaves us with the entire clean data, excluding all poisons, which results in a high SVM accuracy.

Figure 6 illustrates the resulting accuracy on different percentages of poisoned training datasets. The plots on the first row (a to d), second row (e to h) and third row (i to l) belong to MNSIT-4-0, Fashion-MNIST-Sandal-Snkear and CIFAR-10 Airplane-Automobile, respectively, with original accuracies of 99%, 88% and 73% on clean unpoisoned datasets. In each row, all plots have the same scale. Each plot indicates one type of attack and corresponding detection methods.

In each plot, we show the accuracy without any detection (attack), and the accuracy with CAE+, in comparison with other three detection methods (OD, Magnet, and Magnet+reformer). We first elaborate on the results of the first row, for MNIST-4-0 dataset. Considering each plot individually, for all the attack types, CAE+ constantly achieves almost the original accuracy (blue lines), and outperforms other detectors. As expected, optimal attacks are the strongest among all four types of attacks.

Magnet does not consider label flipping, so it fails on flipping attack scenarios. When the feature space changes are significant (mostly semi-optimal attacks), its performance is comparable to CAE+. Magnet's sensitivity to perturbation size has been explored in [26] for evasion attacks under multiple adversarial example distortion rate $\epsilon$. Adding the reformer enhances Magnet's results significantly. It gives us the insight that using the reformer along with CAE+ can boost its performance, which can be a direction for future work. The Fashion-MNIST and CIFAR-10 results are similar to MNIST.

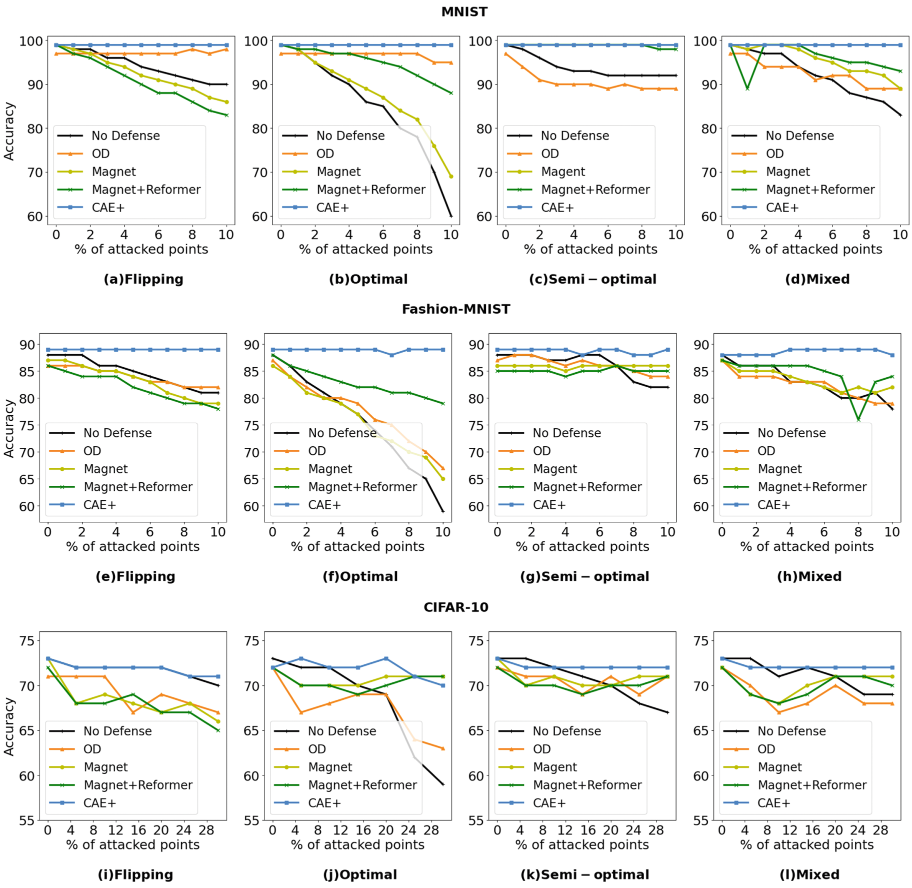


**Fig. 6.** Comparison of SVM accuracy after filtering suspicious points by CAE+, OD, and Magnet over different percentages of poisons. The first row represents MNIST-4-0, the second row is Fashion-MNIST Sandal-Sneaker and the third row belongs to CIFAR-10 Airplane-Automobile.

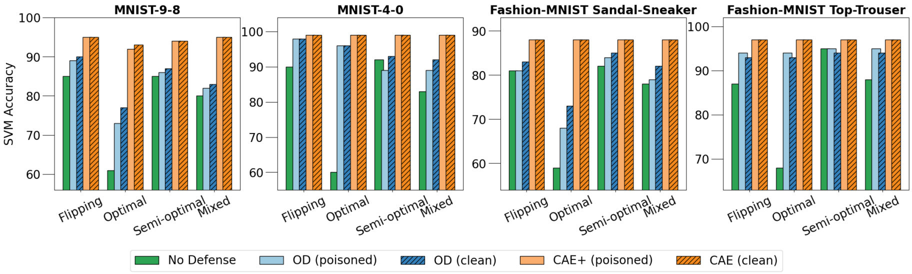


**Fig. 7.** Comparison of SVM accuracy using detectors trained on clean vs. poisoned data

Note that MNIST and Fashion-MNIST were tested for up to 10% of poisoned data, and CIFAR-10 is tested for up to 30% of poisons. Although it is not practical for an attacker to inject this high number of poisons into the system in real world, this is a good stress test to show CAE+ is robust to even higher poison rates.

**Robustness**. Given the assumption of having access to only an untrusted (contaminated) dataset, CAE+ was chosen over CAE in all the previous experiments. However, if clean data is available, we can simply use CAE. Therefore, to verify the impact of this assumption on the detectors' performance, we train a standalone CAE on clean data, utilizing Equation (3) and on a large number of epochs (300). In this experiment, we end up with two new detectors; a clean CAE and a clean OD.

We use a training dataset with 10% poisoned data to train SVM, then apply both clean and poisoned versions of CAE(+) and OD on this data to see how they filter poisoned points and recover SVM accuracy. The result of this comparison on four datasets MNIST 9-8, 4-0, Fashion-MNIST Sandal-Snkear, and Top-Trousers are represented in Fig. 7. The original SVM accuracies on trusted data for these datasets are 95%, 99%, 88%, and 97%, respectively. We observe that OD is susceptible to contaminated data as clean OD usually surpasses its contaminated version. We note that when the defender has access to a clean dataset, it is adequate to train CAE directly without CAE+. Also, CAE+ and CAE always outperform OD, especially in optimal attacks.

## 5 Conclusion

This paper utilized auto-encoders to defend against various types of poisoning attacks for the first time. We proposed CAE, a novel two-component auto-encoder that utilizes an auxiliary classifier to boost detection performance. We enhanced the structure of CAE by introducing CAE+. The enhanced version is a joint auto-encoder detector that has a high robustness against contaminated data. Experiments demonstrated the detection power of CAE+ against diverse poisoning attacks including optimal, semi-optimal and label-flipping attacks and showed that it surpasses the state-of-the-art distance-based outlier detector and Magnet detector. In all these cases, CAE+ is trained on a dataset that is corrupted with a high rate of poisoned data and still preserved its performance.

Directions for future work include demonstrating the results of such detectors on non-convex target models. Studies can also explore the influence of these defensive approaches on multi-classification problems. Besides, it is worth extending current work on backdoor attacks since they are carried out during training and share many characteristics with poisoning attacks.

## References

1. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)
2. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE 2(1) (2015)
3. Aytekin, C., Ni, X., Cricri, F., Aksu, E.: Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2018)
4. Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, pp. 37–49 (2012)
5. Baracaldo, N., Chen, B., Ludwig, H., Safavi, J.A.: Mitigating poisoning attacks on machine learning models: a data provenance based approach. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 103–110 (2017)
6. Biggio, B., Fumera, G., Roli, F.: Security evaluation of pattern classifiers under attack. IEEE Trans. Knowl. Data Eng. **26**(4), 984–996 (2013)
7. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
8. Borgnia, E., et al.: Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3855–3859. IEEE (2021)
9. Carnerero-Cano, J., Muñoz-González, L., Spencer, P., Lupu, E.C.: Regularisation can mitigate poisoning attacks: a novel analysis based on multiobjective bilevel optimisation. arXiv preprint arXiv:2003.00040 (2020)
10. Chan, A., Tay, Y., Ong, Y.S., Zhang, A.: Poison attacks against text datasets with conditional adversarially regularized autoencoder. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 4175–4189 (2020)
11. Chen, J., Zhang, X., Zhang, R., Wang, C., Liu, L.: De-Pois: an attack-agnostic defense against data poisoning attacks. IEEE Trans. Inf. Forensics Security **16**, 3412–3425 (2021)
12. Estellés-Arolas, E., González-Ladrón-de Guevara, F.: Towards an integrated crowdsourcing definition. J. Inf. Sci. **38**(2), 189–200 (2012)
13. Fang, M., Sun, M., Li, Q., Gong, N.Z., Tian, J., Liu, J.: Data poisoning attacks and defenses to crowdsourcing systems. In: Proceedings of the Web Conference 2021, pp. 969–980 (2021)
14. Feng, J., Cai, Q.Z., Zhou, Z.H.: Learning to confuse: generating training time adversarial data with auto-encoder. Adv. Neural. Inf. Process. Syst. **32**, 11994–12004 (2019)
15. Geng, J., Fan, J., Wang, H., Ma, X., Li, B., Chen, F.: High-resolution SAR image classification via deep convolutional autoencoders. IEEE Geosci. Remote Sens. Lett. **12**(11), 2351–2355 (2015)
16. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: Badnets: evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)
17. Hong, S., Chandrasekaran, V., Kaya, Y., Dumitraş, T., Papernot, N.: On the effectiveness of mitigating data poisoning attacks with gradient shaping. arXiv preprint arXiv:2002.11497 (2020)

18. Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., Li, B.: Manipulating machine learning: poisoning attacks and countermeasures for regression learning. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 19–35. IEEE (2018)
19. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1885–1894. JMLR. org (2017)
20. Koh, P.W., Steinhardt, J., Liang, P.: Stronger data poisoning attacks break data sanitization defenses. Mach. Learn., 1–47 (2022)
21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
22. Laishram, R., Phoha, V.V.: Curie: a method for protecting SVM classifier from poisoning attack. arXiv preprint arXiv:1606.01584 (2016)
23. LeCun, Y., Haffner, P., Bottou, L., Bengio, Y.: Object recognition with gradient-based learning. In: Shape, Contour and Grouping in Computer Vision. LNCS, vol. 1681, pp. 319–345. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-46805-6_19
24. Madani, P., Vlajic, N.: Robustness of deep autoencoder in intrusion detection under adversarial contamination. In: Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security, pp. 1–8 (2018)
25. Melis, M., Demontis, A., Pintor, M., Sotgiu, A., Biggio, B.: SECML: a python library for secure and explainable machine learning (2019). arXiv preprint arXiv:1912.10013
26. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 135–147 (2017)
27. Muñoz-González, L., et al.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 27–38 (2017)
28. Nelson, B., et al.: Exploiting machine learning to subvert your spam filter. LEET **8**, 1–9 (2008)
29. Paudice, A., Muñoz-González, L., Gyorgy, A., Lupu, E.C.: Detection of adversarial training examples in poisoning attacks through anomaly detection. arXiv preprint arXiv:1802.03041 (2018)
30. Paudice, A., Muñoz-González, L., Lupu, E.C.: Label sanitization against label flipping poisoning attacks. In: Alzate, C., et al. (eds.) ECML PKDD 2018. LNCS (LNAI), vol. 11329, pp. 5–15. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-13453-2_1
31. Sakurada, M., Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, pp. 4–11 (2014)
32. Shejwalkar, V., Houmansadr, A.: Manipulating the byzantine: optimizing model poisoning attacks and defenses for federated learning. In: NDSS (2021)
33. Shejwalkar, V., Houmansadr, A., Kairouz, P., Ramage, D.: Back to the drawing board: a critical evaluation of poisoning attacks on production federated learning. In: IEEE Symposium on Security and Privacy (2022)
34. Shen, S., Tople, S., Saxena, P.: Auror: defending against poisoning attacks in collaborative deep learning systems. In: Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 508–519 (2016)
35. Steinhardt, J., Koh, P.W.W., Liang, P.S.: Certified defenses for data poisoning attacks. In: Advances in Neural Information Processing Systems, pp. 3517–3529 (2017)

36. Sun, J., Li, A., DiValentin, L., Hassanzadeh, A., Chen, Y., Li, H.: FL-WBC: enhancing robustness against model poisoning attacks in federated learning from a client perspective. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
37. Tahmasebian, F., Xiong, L., Sotoodeh, M., Sunderam, V.: Crowdsourcing under data poisoning attacks: a comparative study. In: Singhal, A., Vaidya, J. (eds.) DBSec 2020. LNCS, vol. 12122, pp. 310–332. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49669-2_18
38. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 11(Dec), 3371–3408 (2010)
39. Wang, Z., Ma, J., Wang, X., Hu, J., Qin, Z., Ren, K.: Threats to training: a survey of poisoning attacks and defenses on machine learning systems. ACM Comput. Surv. **55**(7), 1–36 (2022)
40. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
41. Xiao, H., Xiao, H., Eckert, C.: Adversarial label flips attack on support vector machines. In: ECAI, pp. 870–875 (2012)
42. Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C., Roli, F.: Is feature selection secure against training data poisoning? In: International Conference on Machine Learning, pp. 1689–1698 (2015)
43. Xiao, H., Biggio, B., Nelson, B., Xiao, H., Eckert, C., Roli, F.: Support vector machines under adversarial label contamination. Neurocomputing **160**, 53–62 (2015)
44. Xing, C., Ma, L., Yang, X.: Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. J. Sens. 2016 (2016)
45. Yang, C., Wu, Q., Li, H., Chen, Y.: Generative poisoning attack method against neural networks. arXiv preprint arXiv:1703.01340 (2017)
46. Zhao, M., An, B., Gao, W., Zhang, T.: Efficient label contamination attacks against black-box learning models. In: IJCAI, pp. 3945–3951 (2017)
47. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 665–674 (2017)
48. Zhu, T., Li, G., Zhou, W., Yu, P.S.: Differential Privacy and Applications. AIS, vol. 69. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62004-6
49. Zong, B., et al.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection (2018)