Distribution Agreement

Signature:

____ ______________

Fereshteh Razmi Marani Date

Defensive Machine Learning Techniques for Countering Adversarial Attacks

By

Fereshteh Razmi Marani
Doctor of Philosophy

Computer Science and Informatics

Li Xiong, Ph.D.
Advisor

Joyce Ho, Ph.D.
Committee Member

Yuan Hong, Ph.D.
Committee Member

Vaidy Sunderam, Ph.D.
Committee Member

Accepted:

Kimberly J. Arriola, Ph.D., M.P.H.
Dean of the James T. Laney School of Graduate Studies

Date

Defensive Machine Learning Techniques for Countering Adversarial Attacks

By

Fereshteh Razmi Marani
B.Sc., Sharif University of Technology, Iran 2011
M.Sc., Sharif University of Technology, Iran, 2014
M.Sc., Emory University, GA, 2020

Advisor: Li Xiong, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2023

Abstract

Defensive Machine Learning Techniques for Countering Adversarial Attacks
By Fereshteh Razmi Marani

The increasing reliance on machine learning algorithms has made them a target for exploiting vulnerabilities in these systems and launching adversarial attacks. The attacker in these attacks manipulates either the training data or test data, or both, known as a poisoning attack, adversarial example, or backdoor attack, respectively. They primarily aim to disrupt the model's classification task. In cases where the model is interpretable, the attacker may target the interpretation of the model's output.

These attacks can have significant negative impacts; therefore, it is crucial to develop effective defense methods to protect against them. Current defense methods have limitations. Outlier detectors, used to identify and mitigate poisoning attacks, require prior knowledge of the attack and clean data to train the detector. Robust defense methods show promising results in mitigating backdoor attacks, but their effectiveness comes at the cost of decreased model utility. Furthermore, few defense methods have addressed adversarial examples that target the interpretation of the model's output.

To address these limitations, we propose defense methods that protect machine learning models from adversarial attacks. Our methods include an autoencoder-based detection approach to identify various untargeted poisoning attacks. We also provide a comprehensive comparative study of differential privacy approaches and suggest new approaches based on label differential privacy to defend against backdoor attacks. Lastly, we propose a novel attack and defense method to protect the interpretation of a healthcare-related machine learning model. These approaches represent significant progress in the field of machine learning security and have the potential to protect against a wide range of adversarial attacks.

Defensive Machine Learning Techniques for Countering Adversarial Attacks

By

Fereshteh Razmi Marani
B.Sc., Sharif University of Technology, Iran 2011
M.Sc., Sharif University of Technology, Iran, 2014
M.Sc., Emory University, GA, 2020

Advisor: Li Xiong, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2023

Acknowledgments

First and foremost, I would like to express my heartfelt gratitude to my advisor, Dr. Li Xiong, for her guidance, support, and mentorship throughout my graduate studies. Her kindness, consideration, understanding, and care for her students have always been a source of inspiration and motivation for me. I am deeply grateful for the knowledge and wisdom that I have gained from her, and I will always cherish the impact that she has had on my academic and personal development.

I would also like to thank my dissertation committee members Dr. Ho, Dr. Hong and Dr. Sunderam for their guidance, constructive criticism and insightful suggestions that helped me to improve the quality of my research. I am also grateful to my other mentor, Jian Lou who guided me and kept me motivated throughout my research.

Furthermore, I would like to express my appreciation to my friends and fellow graduate students at AIMS, who have been very encouraging and supportive. Special thanks go to Farnaz, Yifei, Wendy, Pengfei, Rongmei, Hong Kyu, and Mark. I would also like to extend my gratitude to my dear friends Sahar, Safoora, Azade, Samaneh, Zahra, Rezvan, Fatemeh, and Somayyeh, whose unwavering support, friendship and kindness have been a source of comfort and joy.

Moreover, I owe a debt of gratitude to my parents, Mahin and Naser, who always inspired me and helped me to overcome my challenges with their sacrifices, unconditional love and support. I would also like to thank my dear brother, sister and nephew Ali, Sara and Arman who always supported me and wished me the best in my life.

Finally, I would like to express my deepest appreciation and love to my beloved husband, whose kindness, patience, and support have been my constant companions throughout this journey. He is my life partner, best friend and mentor, and I cannot describe how blessed I am to have him by my side.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

1.1.1 Adversarial Attacks

Adversarial attacks against machine learning are a type of cyber attack that uses data to manipulate a machine learning model. The adversarial attacks are mainly divided into two classes, based on the time that the attack is applied. The attackers can insert malicious data into the training process of model (poisoning attacks and backdoor attacks), or at the inference time, after the model is deployed (adversarial examples).

Poisoning Attack

Poisoning attacks which are a type of train-time attacks involve injecting malicious data into the training set in order to manipulate the model's behavior. Poisoning attacks are particularly effective in crowd-based systems, where the attacker can cause damage by poisoning the data collected from outside sources [27, 28, 75, 76, 82, 84]. Some examples of systems that are vulnerable to poisoning attacks include autonomous driving cars, health systems, online review systems, and malware/spam detection systems. The most common types of poisoning attacks are label flipping and optimal attacks, which involve changing the labels of training samples or distorting the feature space of the samples in order to divert the training process from its usual course [9, 89].

Backdoor Attacks

Backdoor attacks are a class of training-time attacks in which the attacker only targets a group of test data that include a specific backdoor trigger [10, 33]. In contrast to conventional poisoning attacks, a backdoor attack retains the inference accuracy for benign samples and only activates in the presence of the trigger.

Adversarial Example Attacks

Adversarial examples are considered inference-time attacks. These attacks involve perturbing a data point in a way that causes the model to make a incorrect prediction, even though the perturbed data point is visually similar to the original data point [32, 43, 51, 60]. Adversarial examples can be difficult to defend against because they often involve small, subtle changes to the data that are difficult for a human to detect.

1.1.2 Defense Approaches

There are two major categories of defense methods: 1. reactive defense (detection) 2. proactive defense (robust training and inference).

Reactive defense

Reactive defense is a category of defense approaches which detects and removes the anomalies existing in the data. Reactive defense methods against poisoned data typically involve the use of outlier detection or contribution-based techniques. Outlier detection are usually distance-based or cluster-based methods that identify and remove potentially malicious data points from the training dataset. These techniques determine whether a data point is poisoned by comparing its distance to nearby points or checking if it shares a common label with the rest of its cluster members [15, 45, 66, 67, 77]. Contribution-based methods, on the other hand, measure the influence of each data point on the final model accuracy, allowing for the identification and removal of data points that have a disproportionate effect on the model's performance [7, 37, 39, 62]. Reactive defenses against adversarial examples typically involve the use of some filters, such as an autoencoder which learns to approximate the manifold of normal examples, or a classifier that distinguishes between normal and adversarial examples [58, 59].

Proactive defense

Proactive defense is another category of defense techniques which aims to create the robust machine learning models or techniques so that they become less vulnerable and less sensitive to encountering malicious data. Proactive defense methods, such as regularization, adversarial training, and distillation, are the most well-known techniques for defending against adversarial examples [32, 64]. These methods aim to provide robustness to a model by adding or modifying the components of the model, making it less sensitive to the adversarial examples. Additionally, augmenting the data and using randomness, such as differential privacy, can make the model more robust against both training and test time attacks [10, 11, 23, 26, 54]. In addition to these best-effort defenses, a new class of certified defenses has been developed that can provide theoretical guarantees of robustness to norm-bounded attacks at both training and inference time. These defenses can be both reactive and proactive, depending on how they are implemented [20, 38, 48, 81]. However, they often have limitations in terms of scalability and model support, making them less practical for use with large datasets or complex model architectures [48].

1.1.3 Gaps and Challenges

There are several limitations to the current defense methods available. Reactive methods, such as outlier detectors, tend to focus on specific parts of the data to detect attacks, such as distortion in feature space or anomalous changes in labels. As a result, these methods require prior knowledge of which parts of the data are anomalous. These detectors also rely on clean data to train the detector on a clean manifold. In addition, they require a pre-defined threshold to distinguish outliers based on their impact on the detector. These factors make these detectors less effective in practice. Furthermore, some of these detectors are computationally expensive due to enumerative retraining.

On the other hand, while proactive defense methods can often offer theoretical or certified results, their effectiveness may be limited to specific categories of attacks. For instance, while adversarial training or randomized smoothing may be effective at defending against adversarial examples, they may not provide [high] robustness against training-time attacks. To address this issue, differential privacy, which aims to protect the privacy of data, has recently been suggested as a proactive defense method against training-time attacks. However, these theoretical results have not been thoroughly evaluated in practice using state-of-the-art differentially private approaches.

Moreover, current defense methods in the literature tend to focus on specific attack obejctives which are based on misclassification. However, some of the latest deep learning models also provide explanations and reasoning for their output class [97]. As a result, attacks that target the interpretability of these models have recently been proposed. Although these attacks are gaining popularity, there are still very few reactive or proactive defense methods in the literature designed to address this new category of attacks. Only a few works in the image domain have proposed inference-time attacks and defenses for interpretable models.

1.2 Research Contributions

In this dissertation, we propose defense approaches that protect machine learning models from the negative impact of adversarial attacks. Our goal is to broaden the defense methods to provide protection against a wider range of adversarial attacks, including diverse types of poisoning attacks that distort the features and/or labels space of the training data. We also aim to find a robust model against the hardest class of training-time attacks, namely backdoor attacks, in which not only the training data is affected but also the feature space of the test data is influenced by the attack. Finally, we extend our proposed defense methods to protect against a new class of attacks that aim to harm the interpretability of the model, rather than the classification accuracy.

In Chapter 2, a detection method based on autoencoders is proposed for various untargeted poisoning attacks, without the need for access to a trusted dataset. In Chapter 3, practical differentially private algorithms, including DP-SGD, PATE, and Label-DP approaches, are explored to provide robustness against backdoor attacks. In Chapter 4, an attack against the interpretability of a model is proposed and a defense method is developed to protect against it. The summarized defense methods we suggest and the corresponding attacks they tackle are presented in Table 1.1. The details of these contributions are discussed in more depth in the rest of this chapter.

Table 1.1: The category of the proposed defense approaches and the corresponding attack's objective they address in each chapter.

	Chapter 2	Chapter 3	Chapter 4
Attack time	Training-time	Training-time	Inference-time
Attack objective	Dropping accuracy	Evasion of targets	Distorting interpretations
Defense category	Reactive	Proactive	Proactive
Defense summary	Auto-encoder	Differential privacy	Sequential auto-encoder

1.2.1 Classification Auto-Encoder based Detector against Diverse Data Poisoning Attacks (Chapter 2)

In this chapter, 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:

1. 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, a reconstruction part for learning the representation of the data from the feature space and a classification part for incorporating classification information into the data representation so it can better detect the poison points.

2. 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.

3. 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.

1.2.2 Prevention of Backdoor Attacks through Differential Privacy in Practice (Chapter 3)

This chapter aims to bridge the theory and practice and provide a comprehensive and in-depth understanding of whether and, more importantly, how various DP models and methods defend against backdoor attacks in practice given the theoretical promise and preliminary evidence in the literature. We study both the standard DP class of algorithms and the Label-DP variant for the first time against backdoor attacks and compare four representative DP and Label-DP algorithms in their defense power. We evaluate their performance empirically on two widely used datasets in the domain of backdoor attacks and differential privacy, namely MNIST and CIFAR-10. To summarize, we make the following contributions:

1. We conduct a comparative study of DP approaches against backdoor attacks, including standard DP-SGD approach and the less-studied PATE approach. Some studies use DP-SGD for training DP models, to defend against poisoning or backdoor attacks. In this chapter, we explore another well-known DP algorithm, PATE (Private Aggregation of Teacher Ensembles), and test it against backdoor attacks.

   We compare PATE to DP-SGD and show that these classical DP approaches can provide robust models for backdoor attacks. Also, we will demonstrate that the bagging structure of the PATE inherently makes it suitable against backdoors.

2. We investigate for a deeper understanding of the impact of noise and other parameters of DP approaches on backdoor attacks. The effectiveness of DP approaches is affected by other parameters besides noise. We explore the origin of these algorithms' resilience by examining whether randomness is the sole player or if the other parameters have an impact.

   We empirically show that the randomness (privacy budget) contributes to mitigating the backdoor attack success rate which is compatible with the theoretical results in the literature. However, we demonstrate that the impact of other parameters can be significant on the outcome, especially for PATE, e.g. the threshold utilized to aggregate the teacher models' outputs.

3. We study the Label-DP class of algorithms for the first time against backdoor attacks including two algorithms ALIBI [56] and LP-2ST [30]. Label-DP protects the privacy of the labels of the training data by ensuring the output model is indistinguishable with respect to the label of a training sample. We have two incentives for this study. First, based on the definition of label-DP, we expect it to break the tight association between the backdoor triggers and their assigned target class. Second, Label-DP methods usually converge faster than regular DP algorithms with a higher model utility. It is because the indistinguishability is only required on the labels, hence less noise can achieve the same level of privacy.

   Our evaluations confirm that Label-DP makes the model more immune to backdoor attacks while maintaining model accuracy. We show that Label-DP is superior to DP approaches in terms of convergence speed. Furthermore, we demonstrate they can achieve better robustness accuracy trade-offs under certain settings. For instance, for a lower percentage of backdoors, ALIBI can eradicate the negative impact of the attack while achieving the highest accuracy among all the other approaches. For stronger attacks with higher backdoors, LP-2ST outperforms other approaches when the privacy budget is low.

1.2.3 Interpretation Attacks on Interpretable Models with Electronic Health Records (Chapter 4)

In this chapter, we study and propose an interpretation attack on temporal Electronic Health Record (EHR) data for the first time, utilizing specific metrics suitable for this data type. An interpretation attack is a kind of adversarial example attack targeting interpretable models for temporal EHR data that aims to change the interpretation (importance vectors of the feature attributes) of the model output while keeping the classification the same. We evaluate our attack against a powerful existing detection technique designed for conventional adversarial examples on EHR data and demonstrate that the attack is not detectable. Furthermore, we aim to make the EHR interpretations robust against the proposed attack. We show that using an auto-encoder to de-noise the data at inference time is significantly more effective than using noisy input, as in the state-of-the-art method SmoothGrad. We summarize our contributions as follows:

1. We propose an interpretation attack on EHR data. This attack is created on top of an interpretable model, so the interpretations are closely tied to the model's predictions. It differs from previous attacks in the image domain, which rely on gradient-based and post-hoc interpretation methods.

2. We propose three metrics to assess the EHR interpretation attack. In the previous works, top-K salient explanations between the clean and adversarial images were used for evaluation. However, it is not suitable for EHR data. Two of our evaluation metrics are alternatives to the top-K criteria, and the third metric is

based on the Wasserstein distance which better captures the similarity between temporal data.

3. We conduct experiments showing that the state-of-the-art detector RADAR, which was designed to detect conventional EHR adversarial examples, is not successful in detecting the proposed attack. We then explore the factors that contribute to this attack evasion.

4. Finally, we present a method to enhance the interpretations' robustness and reduce the attack strength. We employ an auto-encoder to boost the robustness of our interpretations through a de-noising process. We show that out approach outperforms SmoothGrad, which is commonly used in gradient-based methods by averaging noisy data.

Chapter 2

Classification Auto-Encoder based Detector against Diverse Data Poisoning Attacks

2.1 Problem Definition

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 chapter, 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.

2.2 Preliminaries and Backgrounds

2.2.1 Untargeted 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).

Poisoning attacks have different manifestations depending on which part of the data is manipulated during the attack. 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 [67, 89, 92, 98]. *Optimal attacks* are based on optimizing the poisons to drastically degrade 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 [9, 61, 91], the rule of thumb is to initialize poisons

Table 2.1: Various types of poisoning attacks based on tampering different input domains of the initial candidate poisoning points.

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

with real samples from training data set and flip their labels. Since labels are not differentiable, they only optimize the feature space. In this chapter, 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 2.1.

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 [67]. They can determine whether a point is poisoned by comparing its distance to a nearby point or other data points in its cluster [45, 66]. 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.

2.2.2 Auto-encoders in Anomaly Detection

Auto-encoders [6, 85] 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:

$$RE(x) = \parallel x - x' \parallel_p \tag{2.1}$$

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, 71]. Nevertheless most anomalies are recognized as samples with observable differences from the real data [5], 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 [58]. 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.

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 [2, 15, 55]. Even in some works [5, 100] that considered anomalous data in the training process of the auto-encoders, the percentage of anomalies in the dataset is insignificant. 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) [58]. Thus, if they are leveraged in the context of poisoning attacks, they overlook some essential

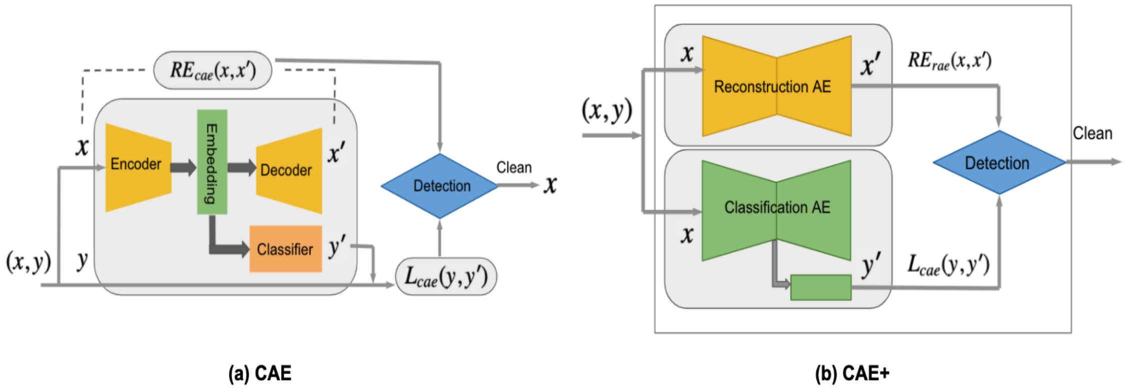


Figure 2.1: **Auto-encoders Structure: (a) The structure of Classification Auto-encoder (CAE).** If trained on pure clean dataset it provides a high success defense against all poisoning attacks. (b) The structure of CAE+. Both Reconstruction Auto-encoder (RAE) and Classification Auto-encoder (CAE) work together to combat against poisons. This joint structure makes the defense method more robust even if trained on a contaminated dataset.

aspects of the attacks, i.e., the labels of the poisoned data (they may be flipped). In this chapter we use auto-encoders for defending agasint poisoning attacks. For training our defense model, we assume 10% of poisoned points. By utilizing a joint architecture, we show that our defensive model can remain resilient to poisoning attacks.

2.3 Proposed Approach

As a baseline solution to untargeted poisoning attacks, 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 sample $s_p = (x_p, y_p)$, $RE(x_p)$ can be used to discriminate x_c and x_p:

$$RE(x_c) << RE(x_p) \tag{2.2}$$

According to (2.2) any data point with significantly large reconstruction error 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. 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. We propose Classification Auto-Encoder (CAE) which has an auxiliary classifier fed by the latent representation z of the encoder (Figure 2.1.a).

2.3.1 Classification Auto-Encoder(CAE)

If RE_{cae} indicates the reconstruction error, and L_{cae} indicates the auxiliary classifier's loss on representation layer z, while training the CAE, it tries 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; first, they train the pair of encoder-decoder and then use the low-dimensional representation for training the classifier [29, 93].

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, since it considers deviations in both feature space and label space.

$$Error(x) = \alpha.RE_{cae}(x) + (1 - \alpha).L_{cae}(x) \tag{2.3}$$

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. α 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 [37]. 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 components 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.

2.3.2 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 method 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 classification layer. 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:

$$Error(x) = \alpha.RE_{rae}(x) + (1-\alpha).L_{cae}(x) \tag{2.4}$$

This is a modification to (2.3), 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 is not overfitting the poisonous data using early stopping. 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 (Figure 2.1.b).

In practice, the training data may be poisoned, so using CAE+ and Equation 2.4 is required. In Section 2.4, 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 on 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.

2.4 Evaluation Results

In Section 2.4.1, 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 offer a fourth type of attack that combines all other poisoning attacks to show the strength of CAE+ against all kinds of attacks. Furthermore, we clarify how we used the periodic update of the model to mimic real scenarios wherein poisoning attacks occur.

In the Results section 2.4.2, we depict the impact of each type of attack on the poisoned data, 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 dominance 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 CIFAR-10. Finally, we illustrate the robustness of CAE+ vs. CAE when, unlike the other experiments, we assume there is a trusted training dataset for training CAE.

2.4.1 Experimental Setup

Datasets

First, we evaluate the performance of CAE+ using the MNIST dataset [47], and more challenging Fashion-MNIST dataset [90] on binary sub-problem 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 [39, 9]. Second, we conduct experiments on a more complex dataset CIFAR-10 [42] for two randomly chosen classes Airplane vs. Automobile. All datasets are normalized within the interval [0, 1].

Attacks

Support Vector Machines (SVM) are known to be subject to strong poisoning attacks [9, 89]. In contrast to complicated models and neural networks [61], poisoning attacks can achieve a high success in dropping the accuracy of SVM. As we will show in Figure 2.6, 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 [57] with some modifications.

Setup

A common paradigm for training ML models in real world is the periodic update [45] 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 chapter is the average of these 10 test runs.

Note that for each of the attacks unless otherwise specified, up to 10% of the clean data are poisoned. Exceeding 10% may not be realistic in practice [41, 14, 9, 45]. 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 [58] with some modifications. Our reconstruction auto-encoders, for MNIST and Fashion-MNIST dataset, consist of 3x3 convolutional layers in the encoder, each composed of 3 filters of size 3x3 with 1x1 strides and sigmoid activations. Between these two convolutional layers a MaxPooling 2x2 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 3x3 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 [58] 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, 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 2.4 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 [41, 66]. One of their interesting properties is that they are very robust against poisoned data and do not require to be trained on a purely clean dataset. So, similar to [66], 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. 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 Magnet original paper in which the authors train the Magnet on a thoroughly clean dataset. It is based on the assumption of access to such a clean dataset, which is valid under evasion attack (against adversarial examples) at inference time, but not the poisoning attacks during training time. We use the same

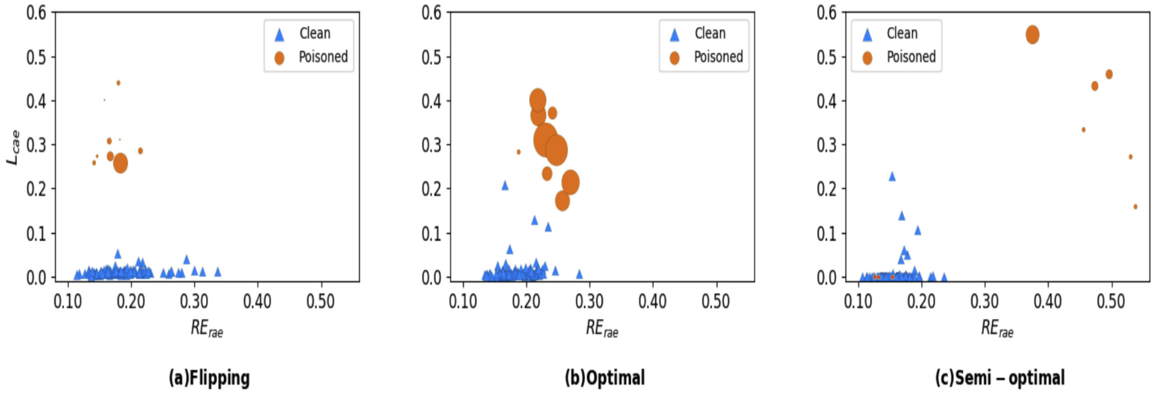


Figure 2.2: **The effect of different attack types on the reconstruction error and auxiliary classification loss for poisoned MNIST-4-0 dataset.** Triangles and circles represent clean and poisoned points, respectively. The poisons' size represents their impact on degrading the SVM accuracy (larger circles indicate higher impact).

structure as the original paper suggests [58], 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 [58]. In this case, after Magnet detector filters out poisons, it passes the remaining data through the reformer, which is another autoencoder. The reformer's reconstructed output will replace the original input and then be fed to the classifier.

2.4.2 Results

Effect of Different Attacks

As we discussed in Section 2.3.1, each type of poisoned data can have a different impact on CAE+ components. Figure 2.2 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 four plots. For the poisoned data, the size of circles indicates their importance in degrading the SVM classification results.

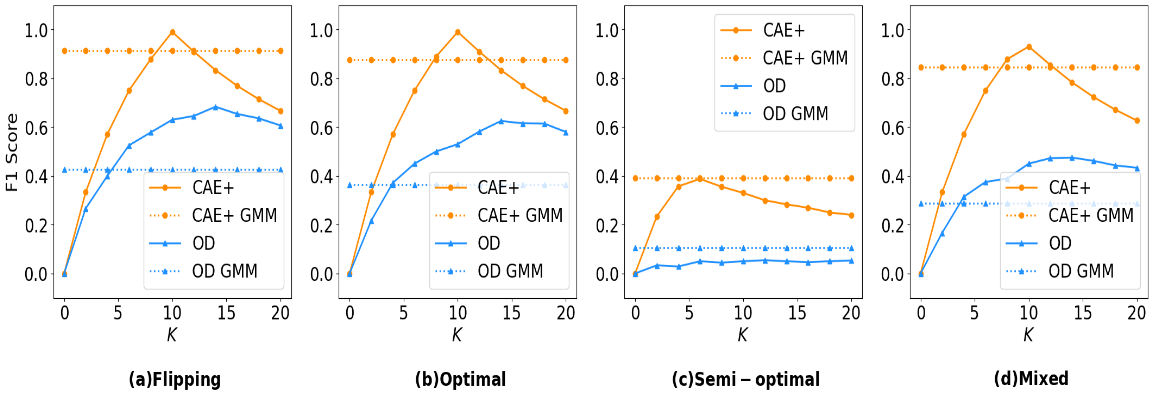


Figure 2.3: **Changes on MNIST-4-0 F1-score over different thresholds for CAE+ and OD.** Thresholds are guesses on the probable number of poisoned data within the training dataset.

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 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 Equation 2.4 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 Section 2.3.1, 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 2.3 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 Figure 2.2. 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 Figure 2.3, 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

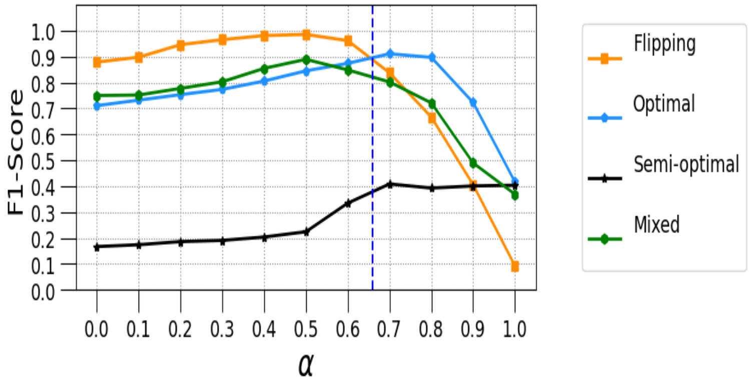
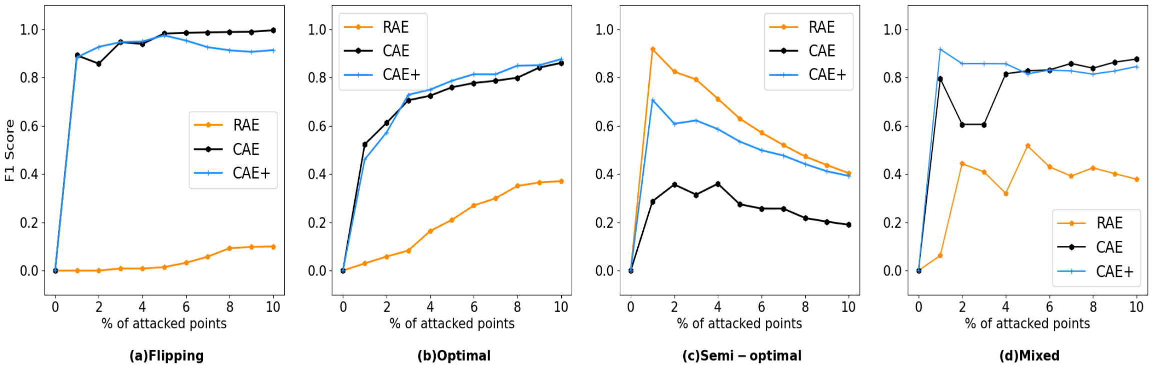


Figure 2.4: CAE+ F1-score for different values of α (Equation 2.4).

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 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 α in Equation 2.4 can be adjusted to meet this goal. Since the attacker's attack type is not known to the defender, α should be pre-adjusted considering all the attack types. Figure 2.4 demonstrates how different values of α affect F1-score. Reconstruction error has a significant impact on semi-optimal attacks, and as a result, higher α 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 α=0.66 in which every attack sustains high F1-socre. According to Equation 2.4, at this value the coefficient of $RE(x)$ is twice as the coefficient of $L(x)$.



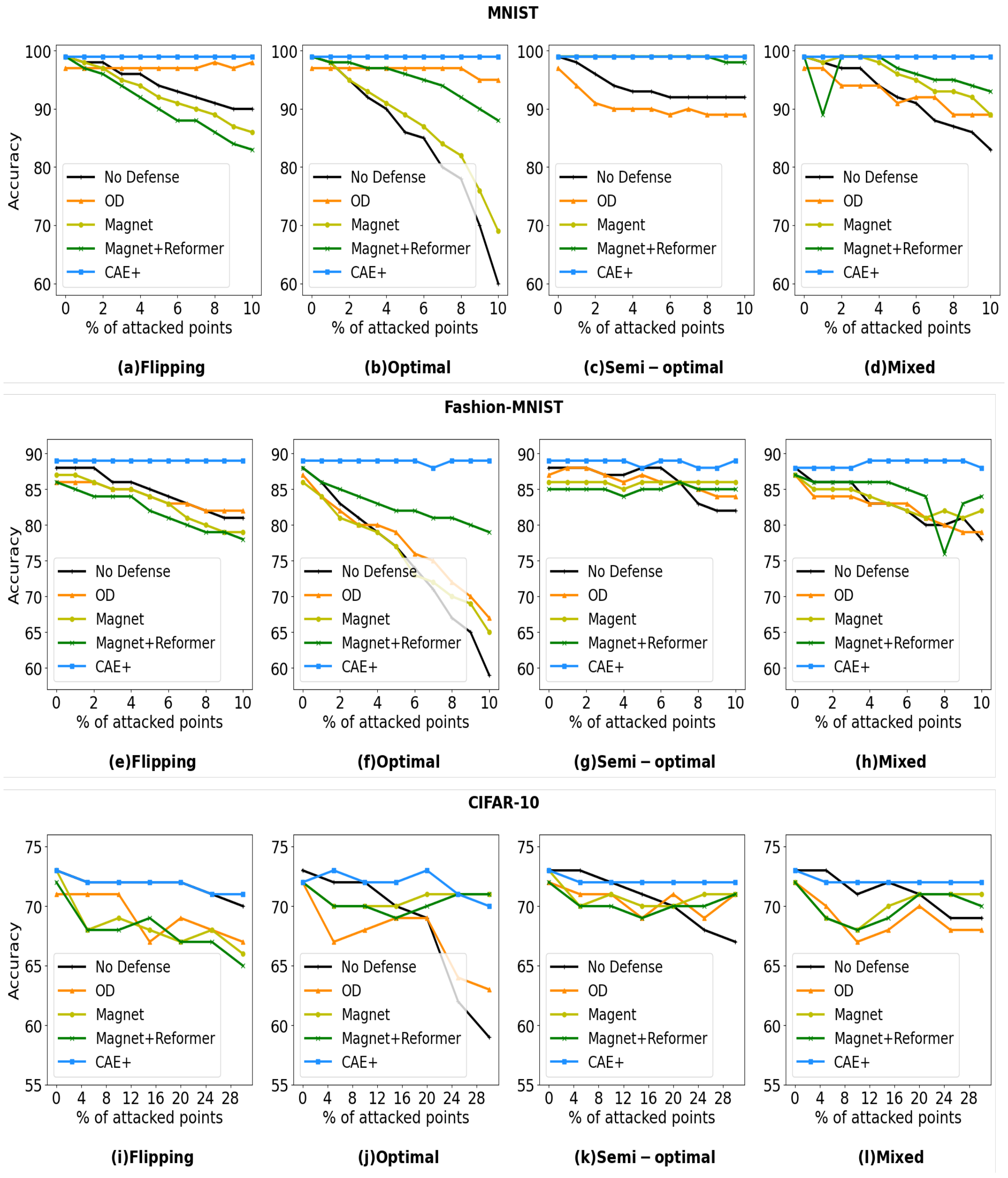
(a)Flipping (b)Optimal (c)Semi – optimal (d)Mixed

Figure 2.5: Ablation study between CAE+, CAE and RAE on MNIST 4-0.

Ablation Study

In this section we show the contribution of each component in CAE+ (recall Figure 2.1). We train two additional models for comparison: 1) CAE that is not combined with the RAE (the bottom auto-encoder in Figure 2.1.b) and has the error function in Equation 2.3; 2) RAE that is a stand-alone reconstruction auto-encoder (the top auto-encoder in Figure 2.1.b) and uses reconstruction error as defined in Equation 2.1.

The error for CAE is calculated based on Equation 2.3, 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 2.5 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. 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.



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

(e)Flipping (f)Optimal (g)Semi – optimal (h)Mixed

(i)Flipping (j)Optimal (k)Semi – optimal (l)Mixed

Figure 2.6: **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.

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 2.6 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 [58] for evasion attacks under multiple adversarial example distortion rate ϵ. Adding the reformer enhances Magnet's results significantly. It gives us the insight that using the reformer along with CAE+ can boost its performance. We did not experiment on CAE+reformer, but it can be a direction for future work. The Fashion-MNIST and CIFAR-10 results are similar to MNIST.

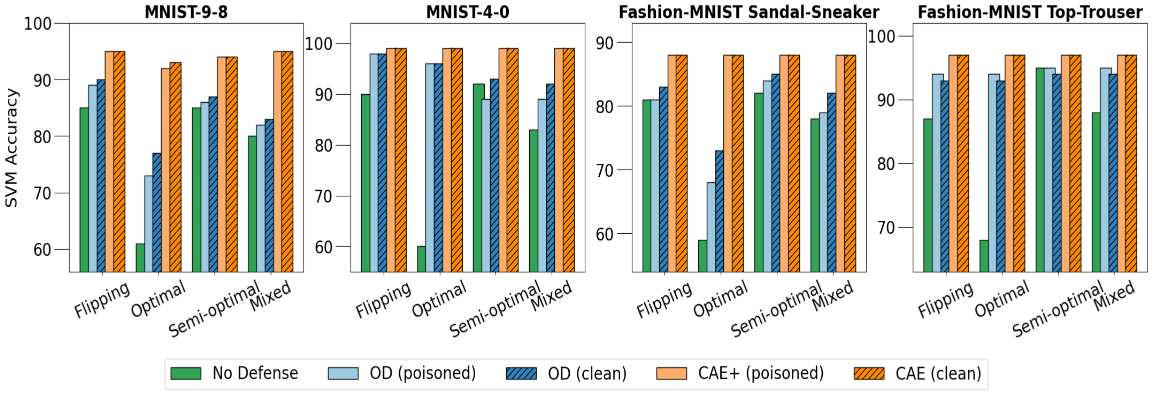


Figure 2.7: Comparison of SVM accuracy using detectors trained on clean vs. poisoned data.

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, but 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 stand-alone CAE on clean data, utilizing (2.3) 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 Figure 2.7. 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+/CAE always outperforms OD, especially in optimal attacks.

Chapter 3

Prevention of Backdoor Attacks through Differential Privacy in Practice

3.1 Problem Definition

Differential privacy (DP) was initially proposed for privacy protection purposes. Recently, it has been widely used for protecting machine learning (ML) models against poisoning and backdoor attacks. Several studies utilize DP-SGD against backdoor attacks. However, there needs to be a comprehensive and in-depth understanding of whether and how well different DP techniques can defuse backdoor attacks in practice. In this chapter, we consider two classes of DP techniques, namely standard DP methods and Label Differential Privacy (Label-DP), and comprehensively investigate their effectiveness against backdoor attacks. We consider PATE and DP-SGD as the primary standard representatives in the first class of DP methods. For the first time in the literature, we investigate PATE in the context of backdoor attacks and compare its effectiveness to that of DP-SGD. We then explore the role of various components of DP algorithms in defending against backdoor attacks. We show that although PATE is very powerful in this context, its power lies not in its privacy preservation characteristics but in the bagging structure of the teacher models it employs.

One of the main issues of DP-SGD and PATE is their prohibitive cost of training. As an alternative solution, we study Label-DP. For the first time in the literature, we use it against backdoor attacks. These classes of algorithms are faster and provide privacy on the labels with less loss of accuracy when compared with classical DP algorithms. We consider two state-of-the-art Label-DP approaches, ALIBI and LP-2ST. We show in our experiments that hyper-parameters of DP algorithms -which are hidden parts of the algorithm and do not directly impact privacy - and also the number of backdoors in the training dataset affect the DP algorithm's success against backdoor attacks. We conclude that, in general, Label-DP algorithms are weaker in the privacy they provide. However, if the hyper-parameters are tuned accurately, they can outperform DP algorithms in some circumstances.

3.2 Preliminaries and Backgrounds

3.2.1 Backdoor Attacks

Backdoor attacks are a category of attacks that involve attaching a small patch to a portion of a base class of the training dataset along with flipping their labels to a specified target class. After the model has been trained using these backdoor samples, it would be vulnerable to the presence of the patch in the inputs. So as the next step of the attack, the attacker attach the same patch to some desired test samples of the base class and pass it to the backdoored model, so that this combination of base class pattern plus the patch pattern mislead the model to misclassify the sample as the target class. This form of backdoor attacks initially introduced by Gu et al. [33] are powerful attacks and have gain many attentions. Some other works tried to make

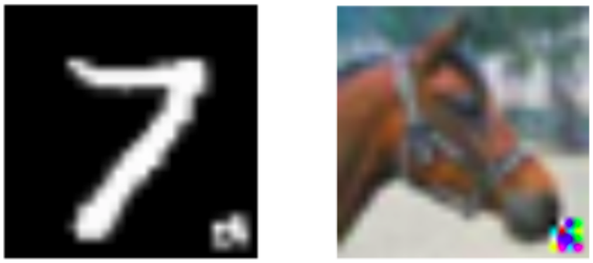


Figure 3.1: Two samples of backdoor input data from MNIST and CIFAR-10 datasets with a 4×4 trigger patch attached to their bottom right corner

some other type of backdoor attacks that are less detectable or employ them in other domains including videos [70, 99].

3.2.2 Differential Privacy and Label Differential Privacy

Differential Privacy (DP) is a privacy-preserving method that makes an observer unable to tell if particular information contributes to the computation. [25] defines DP as a quantifiable notion of individual privacy for statistical algorithms. In the context of machine learning, the model's outcome should not be sensitive to a specific training sample, so it does not reveal whether the training sample has been utilized in the training process.

Differential Privacy.

Let X and Y be the feature and label domain, respectively. Also, let the training dataset consists of n samples from a domain $U = (X \times Y)_n$. Given sample x, we have a classification task for the model M to predict y. A randomized training algorithm $\mathcal{M} : U \to R$ is (ε, δ)-DP if for any two adjacent datasets $D, D' \in U$ differing on at most one sample, it holds that:

$$\forall S \subset R, P[M(D) \in S] \leq e^{\varepsilon} P[M(D') \in S] + \delta. \tag{3.1}$$

A smaller ε guarantees stronger privacy but leads to a lower utility. Using a DP property called **group privacy**, this definition can be extended to two datasets differing in k examples where k denotes more than one data point [24]. It is achievable by a linear increase in the privacy cost.

Label Differential Privacy.

In contrast to the previous definition of DP notion, label differential privacy (Label-DP) considers the labels as the only sensitive part of the training data that requires to be kept secret. A randomized training algorithm $\mathcal{M} : U \to R$ is (ε, δ)-Label-DP if for any two adjacent datasets $D, D' \in U$ that differ on **the label** of at most one sample, it holds that:

$$\forall S \subset R, P[M(D) \in S] \leq e^{\varepsilon}.P[M(D') \in S] + \delta. \tag{3.2}$$

Based on the definitions of (ε, δ)-DP and (ε, δ)-Label-DP, the only difference between these two randomized algorithms is in how the neighborhood in training datasets D and D' are defined. Therefore, Label-DP can be seen as a relaxation of DP algorithms that guarantees only the privacy of the labels. One of the applications of Label-DP is recommendation systems where the user's profile or search queries are public, but the history of the user rating is sensitive.

3.2.3 DP and Label-DP in Deep Learning

In this section, we explore the main methods for achieving DP (DP-SGD, PATE) and Label-DP (LP-MST and ALIBI) respectively.

DP-SGD

Abadi et al. [1] introduce the most widely used algorithm for building DP models. DP-SGD restricts the privacy loss in each iteration of SGD (Stochastic Gradient Descent), by updating model in two steps: 1) clipping the L2 norm of the gradients, and 2) inserting calibrated Gaussian noise into those clipped gradients.

PATE

Papernot et al. [63] provides privacy through a teacher-student structure. First, an ensemble of teachers is trained on disjoint subsets of the private data. Then, given an unlabeled public dataset, a student model queries the teacher ensemble and uses their noisy aggregated vote as the label. The number of queries is restricted. Plus, their response is based on a noisy aggregation without access to any specific private data point. However, access to a public dataset forces a strong assumption on PATE compared to DP-SGD.

PATE was originally introduced with Laplacian noise [63]. Then it was revised to improve the utility and privacy trade-off through a more confident aggregated teacher consensus, called Confident-GNMax [65]. In this work, we adopt the Confident-GNMax version of the PATE framework, which is based on Gaussian noise.

Label Private Multi-Stage Training (LP-MST)

Ghazi et al. [30], as a recent work, achieve Label-DP for deep learning. It leverages a modified version of the Randomized Response (RR) algorithm to add noise to the labels [88]. RR outputs the actual class of a sample or randomly replaces it with one of the other classes. However, the randomness deteriorates the utility.

Ghazi et al. [30] alter the RR algorithm to compensate for the utility, by iteratively training the model on disjoint subsets of the dataset. Then they use the trained model from the previous stage to get the top-K predictions and limit the RR algorithm to

Table 3.1: Parameters of the DP and Label-DP algorithms.

Method	Parameters
DP-SGD	1. **Noise multiplier** : Added randomness to the model's clipped gradients to provide DP 2. **Upper bound of the clipping norm** ($Cnorm$) : Bound to clip the L2-norm of the gradients to control their sensitivity to the noise
PATE	1. **Threshold** T : Queries exceeding this minimum teachers' aggregation are selected for training the student model 2. **Selection noise with variance** σ_1 : Gaussian noise added to the aggregator's votes before applying threshold to enforce privacy 3. **Result noise with variance** σ_2 : Noise added to the selected queries after applying threshold to guarantee DP 4. **Number of teacher models** 5. **Number of queries**
LP-2ST	1. **1. Data split ratio** : The portion the training dataset split between two training stages (more in the first stage helps with accurate prior but causes underfit in the second stage) 2. **2. Temperature T** : For logit z_i and calculation of prior p_i of class i, a small T in $p_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$ boosts the confidence of the top classes and a large T makes the priors more uniform 3. **3. Epsilon** ε : Randomness parameter that is equivalent to the privacy budget
ALIBI	1. **1. noise of soft training labels** : Laplacian noise with $\delta = 0$ which is applied once and determines the privacy budget

those predictions. Similar to the main paper, we report our results on LP-2ST with two training stages.

Additive Laplace Noise Coupled with Bayesian Inference (ALIBI)

Malek et al. [56] propose another Label-DP method in ML recently. It first adds a Laplacian noise to one-hot labels, then uses these soft new labels to train the model while preserving Label-DP. Since the post-processing does not affect differential privacy, Bayesian post-processing de-noises the soft labels iteratively during each step of SGD. The combination of additive Laplacian noise and iterative Bayesian inference increases the utility.

3.2.4 Robustness of DP

DP has recently been highlighted for providing robust models to alleviate the negative impact of backdoor attacks. The rationale is that according to the definition of DP and group privacy, DP models are less sensitive to the impact of one or a group of poisoned data. In this section, we go through the literature to investigate where and how differentially private approaches used to defend against backdoor and poisoning attacks. We then find the gaps in the literature, formulate those as research questions, and try to answer them and assess the results empirically.

There are two lines of work in the literature that considered the defensive power of DP methods on poisoning attacks; theoretical and practical studies.

Ma et al. [54] theoretically prove the robustness of DP models and provide a theoretical bound. They assume a training dataset D and an attacker with full knowledge creates some poisoned dataset $\tilde{D}$ from D. The poisoned model $\theta_{\tilde{D},b}$ is parameterized through the poisoned data $\tilde{D}$ and noise parameter b of the DP model. The attacker's objective loss $C : \Theta \to R$ usually misclassifies some targets or disrupts the overall classifier's functionality. Assuming the attacker does not know the exact realization of the noise, then the attack is reduced to :

$$\min_{\tilde{D}} \quad J(\tilde{D}) = E_b \left[C(\theta_{\tilde{D},b}) \right] \tag{3.3}$$

Given k poisoned data, the authors utilize the property of differential privacy in Equation 3.1 and conclude:

$$J(\tilde{D}) \geq e^{-sign(C).k\varepsilon} J(D) \tag{3.4}$$

According to Equation 3.4 the attacker is unable to change $J(\tilde{D})$ arbitrarily because it is lower bounded by 0 if C is positive (for example, in case of Mean Squared Error) or it is unbounded from below if C is negative.

This chapter provides insight into how DP methods may provide a natural immunity against data poisoning attacks. However, it has two limitations. First, the lower bound of $J(\tilde{D})$ is loose. Second, this work expands its findings on general attack loss functions and DP frameworks. Thus the specific impact of Equation 3.4 on SOTA deep learning models (e.g. DP-SGD) and practical attacks (e.g. backdoor attacks) remains neglected.

To overcome the second limitation, a parallel set of works have employed DP-SGD as a practical usage of differential privacy in deep learning to achieve protection against poisoning attacks [10, 23, 94]. Hong et al. [35] was one of the first works that considered DP-SGD against backdoor and other poisoning attacks. However, their primary motive was not originated from the fact that DP-SGD is a private algorithm and Equation 3.1. Instead, they observed that during the training on a poisoned dataset, the gradients computed on poisoned samples have a higher magnitude and different orientation than those computed on clean samples. Hence they leveraged DP-SGD to offset the behavior of the model's gradients on both clean and poisoned data through the randomness of the gradients. Their results show some degree of protection against specific poisoning attacks, but their outcome is not promising on backdoor (insertion) attacks. Later, Jagielski and Oprea claimed that differential privacy itself can not serve as a defense against poisoning attacks [36]. They argued that it is possible that the robustness of DP-SGD stems from some parameters other than noise.

3.3 Proposed Approach

The existing studies on DP-SGD are inconclusive, and there are no studies on other state-of-the-art DP approaches as a potential defense. It motivates us to extend current works by conducting more comprehensive experiments on DP-SGD and introducing other DP methods as a defense. Based on this primary motivation, in this section, we pose some research questions and elaborate their significance. Then in the following sections, we will try to address them empirically.

Question 1. *Is DP-SGD a successful protective algorithm against backdoor attacks? Can PATE, as another main DP approach, mitigate backdoor attacks?*

Current studies have differing views on whether DP, particularly DP-SGD, can defend against backdoor attacks. It opens the door for a more comprehensive study of DP-SGD. It's not clear whether the robustness is achieved by the randomization by DP methods in general or other algorithmic specific parameters of DP-SGD. Additionally, this outcome can emphasize the gap between DP's theoretical and practical results against poisoning data.

So in this chapter, we first explore DP-SGD to understand why there is no consensus in the literature on DP-SGD as a defensive algorithm. Then for the first time, we explore PATE as a DP method against backdoor attacks to demonstrate if it confirms DP models' robustness. We examine the effectiveness of these algorithms by analyzing their hyperparameters, even those that do not contribute to the randomness for DP. With this investigation, we hope to determine whether these algorithms are effective defense mechanism solely because they are DP.

Question 2. *Can other DP notions, such as Label-DP, also provide robustness and even better accuracy and robustness trade-off? How do different DP notions and algorithms compare in the trade-off?*

Answering the research question 1, leads us to two other major challenges with regard to DP-SGD and PATE. The first challenge is their prohibitive training time. Training an ensemble of teachers in PATE is heavily costly. Also, DP-SGD requires computation of per-sample gradient norms, which is extremely slow. The other issue with the DP algorithms is the trade-off between the privacy budget and the utility, which means decreasing the privacy budget (i.e., achieving stronger DP) is accompanied by a drop in models' accuracy. We will show that lower privacy budgets usually lead to a lower attack success rate (ASR), which is necessary to defeat attacks. We call this simultaneous reduction in accuracy and ASR the *Accuracy-ASR trade-off*. We will define the criteria for attack success rate in Section 3.4.1. To address these challenges, we conduct a comparison between Label-DP and other DP algorithms by varying DP budgets and attack strengths.

3.4 Experiments and Results

3.4.1 Experimental Setup

Datasets and Models

We evaluate each DP model on two datasets: MNIST [46] and CIFAR-10 [42]. We study end-to-end training and fine-tuning since both are common practices in modern machine learning. We use the same CNN architecture as [4] with two convolutional layers for MNIST and train it from scratch. Also, for CIFAR-10, as [86] suggests, we use ResNet50 [34] pretrained on ImageNet as a feature extractor and fine-tune its classification head. Table 3.2 elaborates the details of the architecture of neural networks utilized for the MNIST and CIFAR-10 datasets in Chapter 3.

Table 3.2: Model structures used for MNIST and CIFAR-10.

MNIST ::	$Conv(8 \times 8, 16) \rightarrow ReLU \rightarrow MaxPool$ $\rightarrow Conv(4 \times 4, 32) \rightarrow ReLU \rightarrow$ $MaxPool \rightarrow Linear(32) \rightarrow Linear(10)$
CIFAR-10 ::	$ResNet50 \rightarrow AvgPool \rightarrow$ $Linear(256) \rightarrow ReLU \rightarrow Linear(10)$

Training Configuration.

For each DP algorithm, we use a different training configuration. Corresponding to each DP algorithm's specification, we find an optimizer and a learning rate with a grid search algorithm so that the training process achieves the highest accuracy. In addition, data augmentation reduces the effectiveness of all of the attacks [40, 72]. It leads to a bias in our results. So we skip the data augmentation in our experiments. Table 3.3 lists the training setup for each DP algorithm on CIFAR-10. For MNIST the setup had just some minor differences so we did not list them here. For ALIBI in CIFAR-10 experiments, we have a learning rate that is scheduled according to the piecewise constant with linear ramp-up scheme, previously used by [30]. It increases from 0.003 to 0.1 in the first 30 epochs and then remains piecewise constant at 0.01 and 0.001 in epochs 30 and 40, respectively. We noticed increasing the number of epochs beyond 50 while using various learning rates did not enhance the outcome.

Table 3.3: **Training configurations of four DP algorithms for CIFAR-10.** The physical batch size of DP-SGD is set to 128.

	DP-SGD	**PATE**	**LP-2ST**	**ALIBI**
Optimizer	RMSProp	Adam	SGD	SGD
Learning rate	0.001	0.001	0.001	0.003
Epochs	50	50	50	50
Batch size	512*	32	128	128

To conduct the experiments of Section 3.4.5, we pick the best parameters obtained based on the results of the experiments in Sections 3.4.3 and 3.4.4. The best parameters are those that produce both high accuracy and an ASR as low as possible. Wherever there is a trade-off between accuracy and success rate, we prioritize high accuracy. Table 3.4 and Table 3.5 list these parameters for MNIST and CIFAR-10, respectively.

Table 3.4: **Optimal hyperparamters of different DP algorithms for MNIST.** PATE has two different version, based on what parameter used to change the privacy budget of the algorithm; the noises or the number of queries.

MNIST	
DP-SGD	Optimizer: SGD (lr=0.1) , Cnorm=2
PATE (Noise-based)	#Teachers:200 , #Queries:10000 , Threshold:150
PATE (Query-based)	#Teachers:200 , Threshold:150 , Selection noise:120, Result noise:50
LP-2ST	Temperature:0.5, Data split ratio: 50/50

Table 3.5: **Optimal hyperparamters of different DP algorithms for CIFAR-10.**

CIFAR-10	
DP-SGD	Optimizer: SGD (lr=0.1) , Cnorm=2
PATE (Noise-based)	#Teachers:200 , #Queries:10000 , Threshold:180
PATE (Query-based)	#Teachers:200 , Threshold:180 , Selection noise:100, Result noise:25
LP-2ST	Temperature:0.1, Data split ratio: 40/60

Averaging the Results.

We discovered that Label-DP algorithms are less stable than the DP algorithms. Therefore, to make the results more unbiased, we repeat their training process 10 times, using different random seeds for the noise, and report the average accuracy and ASR. The query-based PATE is the PATE model in which noises are constant, and the number of queries is changed to achieve different privacy budgets. In our experiments, we noticed that the results vary among multiple runs. The reason is that the backdoor samples change in different subsets of the queries. Thus in one query subset, the backdoors can be stronger than the other subset. So we repeat the training of query-based PATE 10 times with a random subset of queries selected from 10000 public data points. Finally, we report the average outcome.

Attack and Threat Model

All the DP models are in white-box settings. The backdoors are made based on the triggers introduced in BadNets [33]. To generate backdoors, we first randomly select two classes as base and target class. Then, according to Figure 3.1 we randomly select half of the samples from the base class, attach a 4 × 4 trigger patch to their bottom right corner and assign the target class as their labels [10]. We poison 50% base class to ensure the number of backdoors is high enough, and sufficient clean samples are left in the base class. Under this condition, the model learns both clean and backdoor data points.

Evaluation Metrics

Attack success rate **(ASR)** is the metric to evaluate the success of the backdoor attacks. According to the definition of the backdoor attacks in Section 3.2.1, ASR indicates the number of test samples from base class that are patched with the backdoor trigger and misclassified as the target class. Thus, a defense method is considered more successful if it leads to a lower ASR.

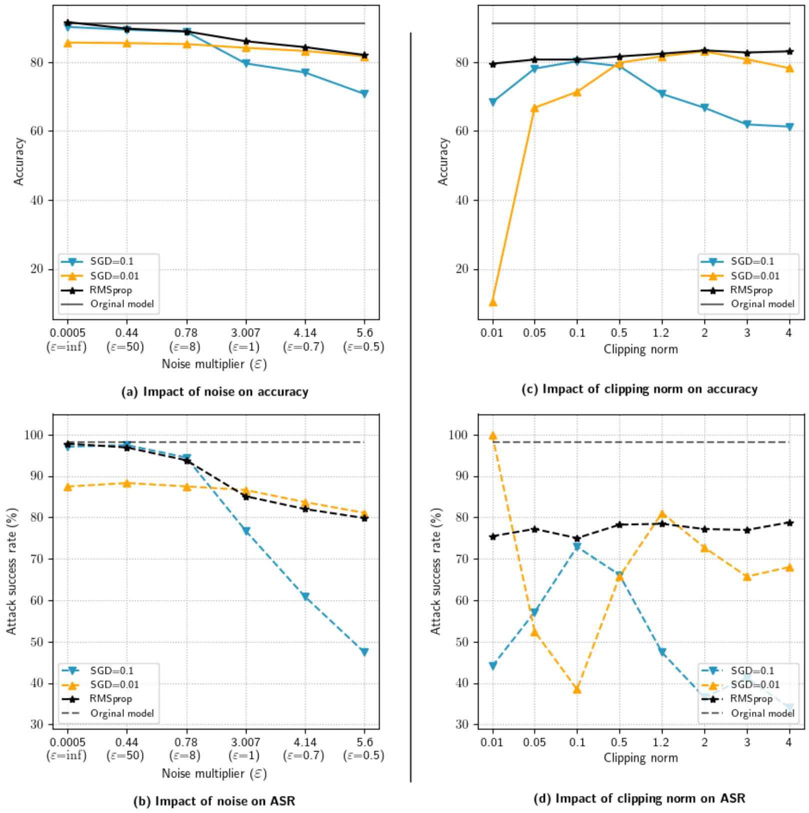
The second defensive purpose is to maintain high **accuracy** for the clean test data. The original accuracy of our CIFAR-10 vanilla model over the clean test data is 91.24% and the backdoor ASR is 98.1%. The MNIST model's initial accuracy and ASR are 98.92% and 100%, respectively.

3.4.2 Experimental Roadmap

This section provides an overview of the experiments in the forthcoming sections. In Section 3.4.3, we analyze two DP algorithms, DP-SGD and PATE, by assessing the impact of their privacy budget and other hyperparameters on the attack success rate. It helps us clarify the underlying reason for their defensive power. At the same time, we will show their resulting accuracy and attack success rate. We will repeat these sets of experiments for Label-DP algorithms in Section 3.4.4. Then, in Section 3.4.5, we compare all the DP and Label-DP algorithms in various circumstances to witness which one is prominent and whether the outcome alters in a different situation.

3.4.3 DP against Backdoors

This section investigates DP-SGD and PATE, against backdoor attacks. For each algorithm, we will evaluate their key hyperparameters (introduced in Table 3.1) on CIFAR-10 dataset and show that some of them have a critical impact on the accuracy and ASR. The results of the MNIST dataset are very similar. So to be concise, we



(a) Impact of noise on accuracy

(c) Impact of clipping norm on accuracy

(b) Impact of noise on ASR

(d) Impact of clipping norm on ASR

Figure 3.2: **Effectiveness of DP-SGD against backdoor attacks, w.r.t the noise multiplier, clipping norm, and the optimizer.** The solid lines on the top figures show the accuracy, and the dashed lines show the percentage of attack success rate (ASR). Higher noises can reduce the ASR, but it costs some accuracy reduction as well (Left). Although clipping norms change the accuracy and ASR, this change does not follow a straightforward pattern (Right). For both noise multiplier and clipping norm, the type of optimizer impacts the results significantly.

skip their reports here but use them to conduct the experiments in the subsequent sections.

DP-SGD vs. Backdoors

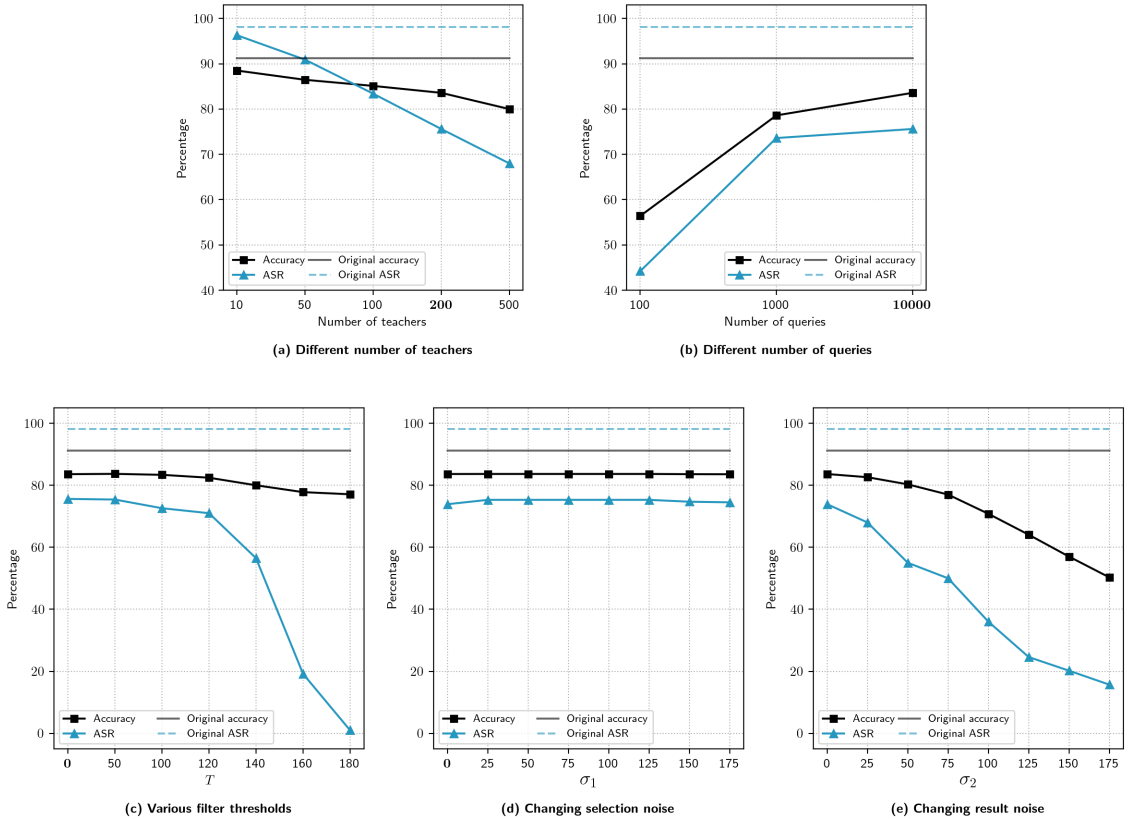
SGD is the dominant optimizer in practice paired with the DP-SGD algorithm, especially in defeating poisoning attacks [1, 11, 35, 36]. So we consider different optimizers and learning rates to depict the sensitivity of DP-SGD performance to these factors: RMSProp, SGD with a learning rate of 0.1, and SGD with a learning rate of 0.01. Based on the size of the dataset, we set the DP-SGD algorithm as $(\varepsilon, 10^{-5})$-DP and report ε as the privacy budget [65].

Figures 3.2.a and 3.2.b show the impact of the noise multiplier by fixing the clipping norm to 1.2 (typical for CIFAR-10). Interestingly, the rate of the accuracy drop to the ASR drop differs for each optimizer. However, in general, the higher noises reduce the accuracy and ASR simultaneously. It suggests that SGD can resist the backdoor attacks more significantly by paying slightly more utility cost.

Figures 3.2.c and 3.2.d illustrate the impact of different clipping norms on the accuracy (top) and ASR (bottom) using a fixed noise of 5.6. In contrast to RMSProp, for SGD optimizers, the choice of learning rate makes two different patterns of ASR w.r.t the clipping norm, which reveals how SGD training without an adaptive learning rate can be affected by the norm of the gradients. So while the clipping norm significantly impacts the model utility and robustness, it is difficult to optimally adjust it when the defender is agnostic to the attack specifications.

According to [12], the impact of the clipping norm on accuracy is not monotonic, which is manifested as a non-monotonic pattern of accuracy and ASR in Figures 3.2.c and 3.2.d. For the reason of the different pattern of ASR in the left side of Figure 3.2.d with SGD-0.01, we speculate that the small learning rate accompanied by a high noise and small clipping norm can hardly learn the normal images' manifold, and instead it retains the repetitive and striking patterns of the backdoor triggers.

Conclusion (Q1): In our evaluations, DP-SGD was successful in mitigating the impact of backdoor attacks. However noise multiplier, clipping norm and training parameters determine the extent of this success. As a result, differences in these parameters contribute to the varying results reported in previous studies on the effectiveness of DP-SGD as a defense mechanism.



(a) Different number of teachers

(b) Different number of queries

(c) Various filter thresholds

(d) Changing selection noise

(e) Changing result noise

Figure 3.3: **The impact of number of teachers, number of queries, threshold, selection noise and result noise on the student model's accuracy and ASR from left to right and top to bottom, respectively).** For figures (a) to (e), we select the optimized value of the discussed metric (highlighted on x-axis) and use it in the next figures. The blue lines represent ASR and the black lines are accuracy. The flat lines are calculated on vanilla model. The number of public queries is bounded to 10000 (b).

PATE vs. Backdoors

In this section, we evaluate the robustness of PATE against backdoor attacks and the impact of different parameters including number of teachers, number of queries, threshold, selection noise, and result noise. The result is shown in Figure 3.3. Whenever noises or threshold are not evaluated, we fix their values equal to 0. In the case of number of queries and number of teachers, the default values are fixed to 10000 and 200, respectively. For training PATE, we assume 1/5 (i.e. 10000 samples) of the training data is publicly available for training the student model, and the rest is private. In the original PATE paper [63], the number of queries is set to as low as 1000. However by doing so, we naturally remove a large fraction of poisoned data and make the comparison between different DP methods unfair. So we keep the default number of queries 10000 and in the next sections to compare the models, we analyze the impact of both noise and the number of queries on the PATE's utility and privacy budget.

Figures 3.3.a and 3.3.b show the number of teachers and the number of queries impact the accuracy and ASR in opposite ways. A higher number of teachers means fewer training data and lower accuracy for each teacher, hence less accurate consensus from the aggregator. It also compromises the consensus on assigning the target class to the backdoor samples and decreases the ASR, which aligns with the literature finding that bagging can hinder the success of the backdoor attacks [8, 18, 38]. Furthermore, in Figure 3.3.b, lower number of queries are associated with less training data for the student model and fewer backdoors, hence lower accuracy and ASR.

Figure 3.3c illustrates that aggregation threshold is crucial in defeating backdoors and has minimal impact on utility loss. This finding complements previous results to use bagging against poisoning attacks. The threshold forces the aggregation process to filter out uncertain data and backdoors, resulting in higher accuracy and lower ASR in the student model. To the best of our knowledge, this factor has not been considered in previous works as a major contributor to the effectiveness of bagging.

Figures 3.3.d and 3.3.e demonstrate the effect of selection noise and result noise used in selecting and randomizing queries which form the basis of DP for PATE. We found the same trends when one of the noises is fixed to a random positive value. Based on these results, to defeat ASR we need a high result noise which leads to a dropped accuracy. Since we fixed the number of queries and only varied the noise values to control privacy, the privacy budget still remains as large as $\varepsilon = 4$ at a high noise level of 175.

Conclusion (Q1): PATE is very successful in defeating backdoor attacks. It can be more successful than DP-SGD but it is highly sensitive to the algorithm parameters. Result noise (σ_2) and number of queries which are the most influential parameters on the privacy budget (ε) decrease the ASR but they also cause a drastic decrease in the accuracy at the same time. On the contrary, the best result is achieved through tuning the threshold, although it cannot provide any DP by thresholding alone.

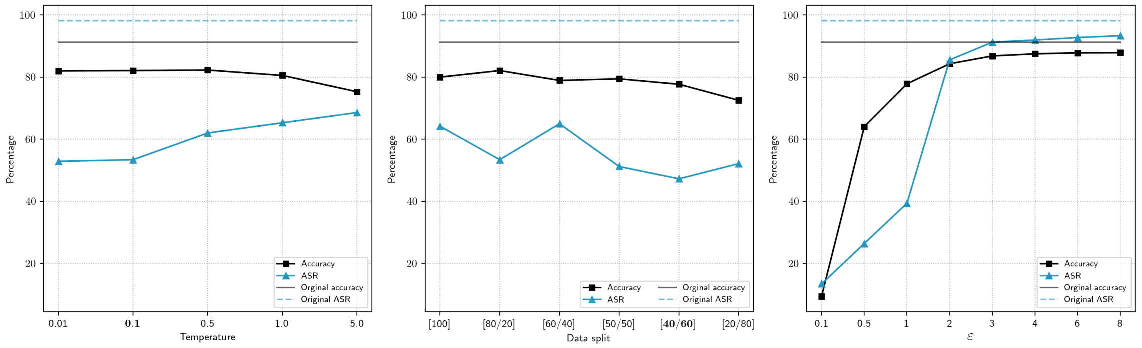
3.4.4 Label DP against Backdoors

Label-DP add randomness only to the labels, and keep the feature space intact. Thus it raises this concern that the impact of the Label-DP approaches can be limited on the backdoor attacks in comparison to the regular DP approaches which adding randomness to the input space. We will show that despite this fact, Label-DP methods still can disentangle the association between the trigger patch in backdoor samples and their associated wrong labels, hence mitigate the backdoor attacks. In this section, we evaluate LP-2ST, and ALIBI as two Label-DP models. We investigate if their randomness or other related parameters can help to mitigate the backdoor attacks. To this end, Table 3.1 presents the various parameters involved in these algorithms.

LP-2ST vs. Backdoors

For each figure from left to right, we pick a parameter bold on the x-axis (which are chosen randomly) and apply it for the experiments in the succeeding figure. For the first two figures, we set $\varepsilon = 1$.

Figure 3.4.a demonstrates the effect of temperature with a random data split of [80/20]. Compatible to [30], sparsifying the priors helps to improve the utility, but to our surprise, it decreases ASR. We speculate the reason is that the backdoor still has a touch of the base class. Thus the first round of LP-2ST predicts target and base



(a) Impact of temperature

(b) Impact of data split

(c) Impact of epsilon

Figure 3.4: **The impact of temperature, data split between two stages and epsilon on LP-2ST (from left to right).** Epsilon, the factor of privacy-preserving in LP-2ST, can drastically deteriorate the ASR with an acceptable utility cost (c).

classes as the backdoors' top-2 classes. The sparsified prior shifts the probabilities of these two classes far away from zero, so the algorithm selects the base class more confidently.

In Figure 3.4.b the training data has been partitioned for two stages. [p1/p2] on the x-axis indicates the percentage of the data in stage 1 and stage 2 of LP-2ST, respectively. When 100% of data is allocated to the first stage, it means that we are using LP-1ST with RR. There is not a clear pattern between ASR and data split. But an LP-2ST model with more data in the first stage has more enhanced priors and higher accuracy.

Figure 3.4.c compares different privacy budgets ε, which is the random factor of the RR algorithm. Naturally, more randomness helps to decrease the ASR. Especially the results for $\varepsilon = 1$ are impressive since it drops the ASR to less than 40%, while the accuracy is still roughly 80%.

Conclusion (Q2): To our surprise, eventhough Label-DP only randomize the labels, but it is still successful against backdoor attacks. In this success all parameters are involved but noise has the major impact. LP-2ST vividly can mitigate the attack but it is very important what ε is selected to obtain a reasonable accuracy-ASR trade-off.

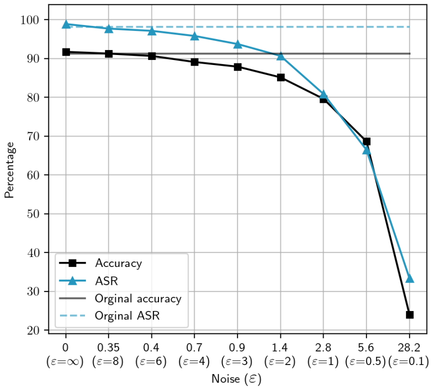


Figure 3.5: **Effectiveness of randomizing labels on reducing ASR in ALIBI**. The noise added to one-hot labels in ALIBI impacts both accuracy and ASR proportionally.

ALIBI vs. Backdoors

According to Figure 3.5, ALIBI with higher noise drops both accuracy and ASR proportionally. It can be justified by the fact that all the labels randomly change just once at the beginning of the training.

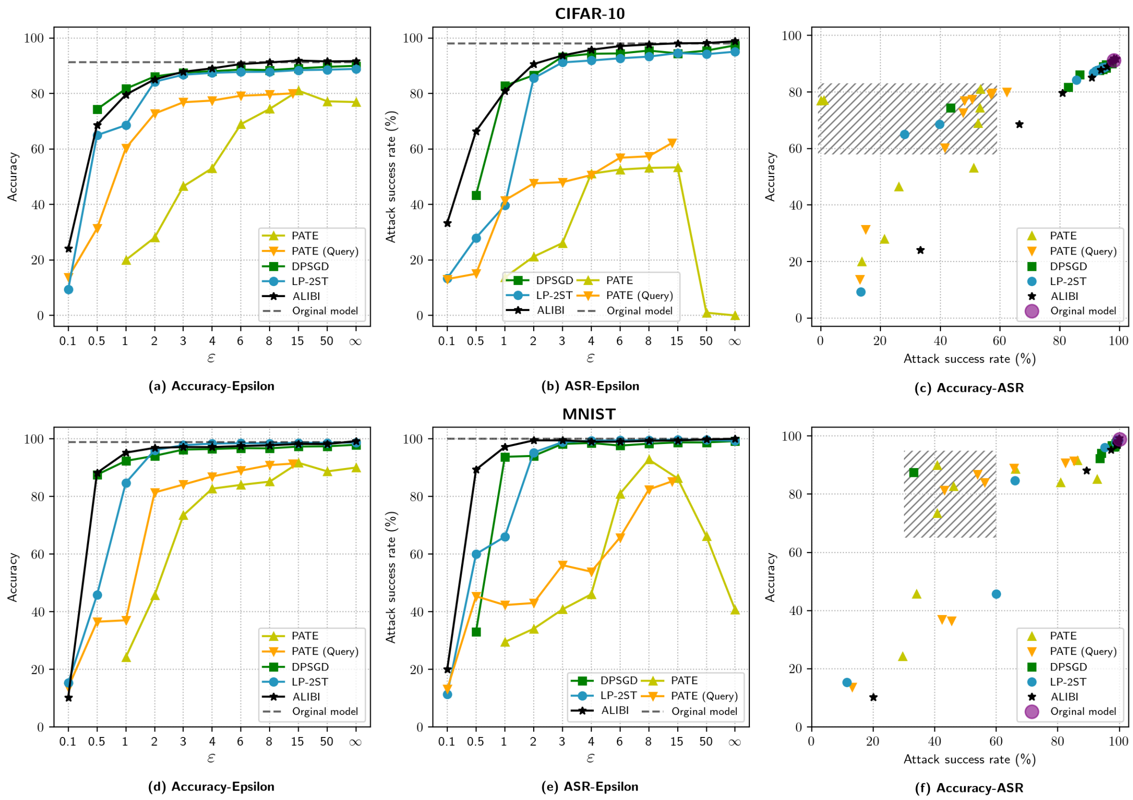
Conclusion (Q2): On average, ALIBI can mitigate the effect of backdoor attacks but with reduced utility costs.

3.4.5 Comparison of DP and Label-DP Methods

In this section, we compare all the DP and Label-DP algorithms to discover which one and under what conditions are more successful.

Privacy Budget Analysis

The ϵ in DP and Label-DP serves two different goals. So we do not directly compare the ϵ values of the two methods even though both can be reduced to label DP [30]. Instead, what we care is the trade-off between accuracy and ASR provided by varying ϵ of the two methods. We pick the best parameters from the results in the previous section to conduct the current experiment. The best parameters lead to high accuracy



(a) Accuracy-Epsilon

(b) ASR-Epsilon

(c) Accuracy-ASR

(d) Accuracy-Epsilon

(e) ASR-Epsilon

(f) Accuracy-ASR

Figure 3.6: **The impact of epsilon on DP and Label-DP methods using CIFAR-10 (top) and MNIST dataset(bottom).** For PATE, the impact of changing noises and the number of queries are investigated.

and a low ASR. Wherever there is a trade-off between accuracy and ASR, we prioritize accuracy. For MNIST, we do not present those parameter selections due to the similar outcomes.

Figures 3.6.a and 3.6.b compare the accuracy and ASR of the different methods for CIFAR-10 with varying ϵ while 3.6.c shows the trade-off of accuracy and ASR of different methods (the ideal case correspond to 100%accuracy and 0% ASR). PATE can achieve different levels of privacy by varying two factors: 1) noises (lime green plots), and 2) number of queries (orange plots). The first observation is that non-DP PATE outperforms all other results and methods (the rightmost point of the lime green plot). It indicates the power of bagging with a threshold against backdoor attacks. LP-2ST for some ϵ values works well. For instance, $\varepsilon = 1$ has high accuracy

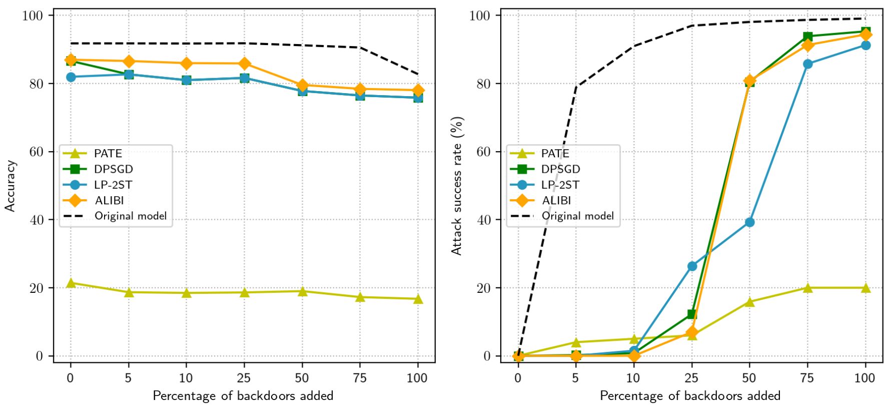
(78%) and a significantly decreased ASR (39%). However DP-SGD gives the best results when $\varepsilon = 0.5$. For ALIBI, both accuracy and ASR drop proportionally.

Figures 3.6.d, 3.6.e and 3.6.f show similar trends for MNIST. Figure 3.6.f combines the results of the two other columns by directly comparing the accuracy and corresponding ASR. The rectangular areas with the hatched pattern in the last column consist of the most desired results with high accuracy and dropped ASR regardless of their privacy budget. It includes different private algorithms, but mostly PATE, which indicates the dominance of PATE.

Conclusion (Q2): The DP and Label-DP techniques effectively reduce the vulnerability of backdoor attacks, albeit at the cost of decreased accuracy. If the optimal approach is determined by the accuracy-to-ASR ratio, then the superiority of each DP or Label-DP model depends on the allocated privacy budget. Label-DP approaches have this advantage that they do not need any extra information such as access to a public dataset as PATE does. However since Label-DP only provides differential privacy for labels, it is weaker than DP approaches in terms of achieving differential privacy. Therefore, if we aim to achieve both privacy and robustness, Label-DP may not be sufficient.

Attack Strength Analysis

We discussed the hyperparameters and the privacy budget of the algorithm as two factors that impact the immunity of the DP approaches against backdoor attacks. A third factor that should be considered when assessing the level of immunity is the strength of the attack itself. So far, we have synthesized powerful attacks by poisoning 50% of the data with backdoors. However, in practice, the attacker conceals her malicious activity by limiting the percentage of poisoned data introduced into the pipeline. Therefore we change the percentage of the backdoors in the base class to develop a range of more realistic and more powerful (but less realistic) attacks.



(a) Accuracy-Backdoor%

(b) ASR-Backdoor%

Figure 3.7: **The significant impact of poisoned data on DP-based defense methods.** The epsilon is fixed to 1 and then all the methods are compared by varying the percentage of the training data that has been poisoned. The accuracy does not show a drastic change (Left). However the ASR is very dependent on the number of poisoned data (Right).

Figure 3.7 shows the accuracy and ASR w.r.t number of backdoors, when the privacy budget for all DP algorithms has been fixed to $\varepsilon = 1$. We observe that the accuracy does not drastically change w.r.t number of backdoors, yet the ASR increases as the attack becomes more powerful. Looking at the pattern, we can see that the DP algorithms almost entirely diffuse the attack when the percentage of backdoors is sufficiently small. It should be noted that the low accuracy of PATE is a result of controlling its privacy budget by adding noise, rather than limiting the number of queries according to the reasoning we had in Section 3.4.3.

Conclusion (Q2): These results illustrate the effectiveness of DP-SGD, LP-2ST, and ALIBI against more realistic backdoor attacks (with backdoor% ≤ 10). For such attacks, the accuracy drops by 10%, and the attack achieves no success. This is compatible with Equation 3.4 that shows that the attacker's loss limit in DP models is theoretically linked to the number of poisoned data.

Table 3.6: Comparison of the highest accuracy and epsilon that DP methods can achieve while ASR=0.

	DP-SGD	**PATE**	**ALIBI**	**LP-2ST**
Accuracy	88.67	85.02	**89.53**	79.9
Epsilon	2	**inf**	2	0.9
Time	140s	220	**59s**	**58s**

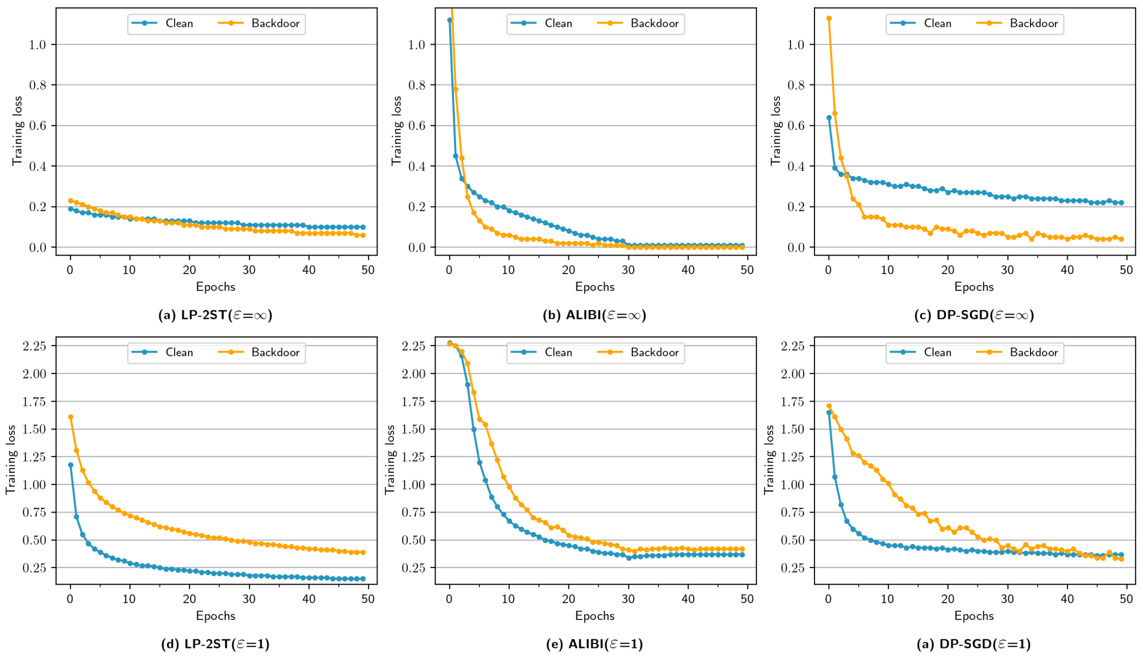
Accuracy-Privacy Trade-off

To see the accuracy when a perfect defense is desired (close to 0 ASR), we have analyzed different privacy budgets for each DP method and found the greatest ε where the ASR does not exceed 1%. This small ASR is achievable when the number of backdoors is insignificant (we set it to 10%). By doing so, we achieve the least randomness that leads to a successful defense. After removing the impact of the attack, we can have a fair comparison of accuracy and training time.

Table 3.6 highlights the best values of accuracy, privacy budget, and training time in each row. The previous findings indicate that a deterministic version of PATE, with noise removed, is the most resilient against attacks. However, when the goal is to simultaneously defend against backdoors and protect privacy, this result is not favorable for PATE. DP-SGD and ALIBI, with the same privacy budget, can achieve better accuracy than PATE.

Finally, with respect to training time, two Label-DP methods demonstrate a considerable reduction in training time, surpassing other DP techniques. It is important to note that this experiment was conducted on a CIFAR-10 fine-tuning task, where training time is negligible. However, in more complex architectures with end-to-end settings, time may become a bottleneck for PATE and DP-SGD. Please note that the inference times are very similar since the model architecture is the same for all the approaches.

Conclusion (Q2): When a perfect defense is desired, Label-DP methods offers best efficiency and comparable or better accuracy trade-off to DP approaches.



(a) LP-2ST(ε=∞)

(b) ALIBI(ε=∞)

(c) DP-SGD(ε=∞)

(d) LP-2ST(ε=1)

(e) ALIBI(ε=1)

(a) DP-SGD(ε=1)

Figure 3.8: An overview of the training process of LP-2ST, ALIBI and DP-SGD using $\varepsilon = \infty$ (upper) and $\varepsilon = 1$ (lower).

Training Process

In this section, we compare the training process of DP-SGD, LP-2ST, and ALIBI on CIFAR-10. These comparisons are based on two privacy budgets $\varepsilon = \infty$ and $\varepsilon = 1$, to provide an overview over the training process with and without randomness. For LP-2ST, we only illustrate the training of the second and final stage of the algorithm.

In Figure 3.8, each column demonstrates a different method, and each row indicates one of the privacy budgets. For all three differentially private methods, on the first row, with $\varepsilon = \infty$, the loss of the backdoor samples drops below the clean loss on early training epochs. It is the opposite for all three methods when $\varepsilon = 1$ on the second row. For LP-2ST the backdoor loss does not converge to the clean loss and remains higher. It is consistent with the results of LP-2ST at $\varepsilon = 1$ in Figure 3.4.c. For ALIBI the clean and backdoor losses are changing very closely. It explains the similar values for the ALIBI accuracy and ASR in Figure 3.5. DP-SGD can resist the backdoor samples on early epochs. So one of the suggestions is to stop the training early to avoid backdoors to overfit.

Conclusion (Q2): During training, the model underfits or suppresses the backdoor samples which results in defusing the backdoors' impact on the model. This finding confirms the results of the previous sections.

Chapter 4

Interpretation Attacks on Interpretable Models with Electronic Health Records

4.1 Problem Definition

The emergence of complex deep neural networks made it crucial to employ interpretation methods for gaining insight into the rationale behind model predictions. However, recent studies have revealed attacks on these interpretations, which aim to deceive users and subvert the trustworthiness of the models. It is especially critical in medical systems, where interpretations are essential in explaining outcomes. This chapter presents the first interpretation attack on electronic health records (EHRs). Prior attempts in image interpretation mainly utilized gradient-based methods. In our research we use RETAIN medical interpretable model that uses attentions instead of gradients. We show that our attack can attain success on EHR interpretations using RETAIN. We introduce metrics compatible with EHR data to evaluate the attack's success. Moreover, our findings demonstrate that detection methods that have successfully identified conventional adversarial examples are ineffective against our attack. We then propose a defense method utilizing auto-encoders to de-noise the data and improve the interpretations' robustness. Our results indicate that this de-noising method outperforms the widely used defense method, SmoothGrad, which is based on adding noise to the data.

4.2 Preliminaries and Related Work

4.2.1 Attacks on Images via Model's Gradients

Machine learning algorithms, particularly deep neural networks, are widely used in various real-world tasks. However, their inner workings are often seen as a black box. Thus, interpretation methods are essential for explaining an algorithm's output, allowing users to understand how and why an algorithm arrived at a particular decision. Especially in sensitive applications such as medicine, interpretations improve the system's reliability and enable the discovery of new biomarkers and important features for future decision-making processes. For instance, Quellec et al. [68] use heatmaps to identify local patterns and demonstrate which pixels in retinal fundus photographs are involved in the early signs of retinal disease.

Post-hoc interpretability are a set of interpretation methods that seek to explain the predictions of models without relying on their underlying mechanisms [50]. Gradient-based approaches are commonly used in image classification to extract these explanations [73, 78, 79]. They result in a saliency map that explains the output of the model (usually a convolutional neural network (CNN)) by visualizing the areas of the input image that contribute the most to the network's output. However, saliency maps are less common in Recurrent Neural Networks (RNN) since RNNs are typically used for sequential data such as time-series.

Recent research has shown that these methods are vulnerable to interpretation attacks, where small perturbations deliberately crafted and added to input images to distort the explanations [31]. Several techniques have been proposed to address this issue, including adding randomness to the input called SmoothGrad [80, 96], modification of the model architecture [21], or altering the training process using regularization or integrated gradients [16, 22]. These approaches are highly dependent on the architecture of image models and their gradients.

Although interpretation attacks resemble conventional adversarial attacks, which aim to change the classification of an adversarial example [32], they have received less attention due to the challenge of defending high-dimensional saliency maps and the absence of a ground truth for interpretation. Due to the hardship of the interpretations attack and the methods for their robustness and since they are based on gradient approaches, they are mostly limited to images.

4.2.2 Attack on EHRs via Medical Attention-based Models

Sequential electronic health records (EHR) are crucial data sources in the medical field, containing discrete data of patients' vital values and lab values collected over time and across hospital visits. Due to the importance of these data and their use in many classification based predictive models, recent efforts have been made to enhance the interpretability of models trained on EHR data.

Recent research in the medical field has focused on using the attention mechanism to improve the interpretability and accuracy of predictions made using EHR data [17, 19]. The attention mechanism is an approach used in machine learning models that assigns a weight to each input feature, indicating its relative importance to the model's final decision. They generally use BERT models [49, 69, 74] or multi-layer RNNs [44, 52, 53, 95] as the baseline to obtain the attentions.

Despite the prevalence of interpretation attacks in image classification, to the best of our knowledge, no interpretation attacks have been studied targeting EHR-based models. Conducting interpretation attacks on EHR data presents significant challenges due to the unique characteristics of the data. Firstly, for building interpretable models using EHR data, models are designed to produce predictions and interpretations simultaneously. In contrast, image interpretations are mostly gradient-based and created via post-hoc approaches. Thus, manipulating the EHR interpretations can easily alter the patient phenotype, consequently affecting the predicted class.

Secondly, the structure of EHR data is vastly different from images. As a result, the widely used L_∞ norm based attacks in image domain are less meaningful in the EHR domain since L_∞ does not capture the distance between the sequential data well (e.g., the temporal trends). Also, unlike images, EHR data consist of multiple attributes, such as heart rate or temperature, whose values are sequential and time-dependent. Therefore, moving across time and attributes significantly influences the interpretations. Consequently, the criteria used for assessing the image interpretation's robustness on previous works cannot be directly applied in the EHR domain.

4.3 Proposed Approach

In this section, after decribing the problem setting, we present our approach to the interpretation attack on EHR data and elaborate the rationale behind each objective loss term. We then improve the attack by incorporating dynamic weighing to penalize the attack optimization process and reduce the detectability by modifying the penalty term. We propose new metrics as the current evaluation metrics are unsuitable for EHR data. Finally, we explore methods for defending against the attack and demonstrate that de-noising is more effective than the state-of-the-art method for improving the robustness of interpretations.

4.3.1 Problem Setting

EHR dataset is a set of clinical trajectories for patients where each trajectory is a sequence of hospital or clinic visits, each visit corresponding to a set of attributes / measurements. Given longitudinal EHR data from N patients, denote $X^{(n)}$ as the clinical trajectory of patient n, which is characterized by a sequence of t_n hospital visits. Then $X^{(n)}$ can be formulated as

$$X^{(n)} = [X_1, X_2, ..., X_{t_n}], \tag{4.1}$$

where $X_i \in R^d$ denotes the variables from d vital sign measurements and lab events of the *i-th* visit made by patient n. Each $x_{i,j}$ shows *j-th* attribute in the *i-th* visit. In the following, we omit the superscript (n) to reduce clutter. Given a neural network model $f : R^{(t,d)} \rightarrow R^c$ where c is the number of possible classes, we denote the interpretation that is associated with the parameters of function f as $\Phi_f : R^{(t,d)} \rightarrow R^{(t,d)}$ in which every attribute in a specific visit gets a score that shows its importance on the predicted outcome. Given a test input X, the class and explanations of this input is determined by $c^* = \arg\max_c f(X)$ and $\omega = \Phi_f(X)$, respectively.

The chosen model for this study is RETAIN, an EHR attention-based RNN model proposed by [19], with the aim of demonstrating vulnerability as a proof of concept. While there are more advanced attention-based models available, such as [49, 52, 69, 95], we recognize that they were not addressed in this chapter. Therefore, we suggest that these alternative models be explored in future research to potentially enhance the results. RETAIN can give interpretation on both visit (temporal point) and attribute level without requiring to access to some extra meta data. In RETAIN, the impact of each input $x_{i,k}$ on the final classification result is calculated using the two-level attention weights:

$$\omega_{i,k} = \alpha_i W(\beta \odot W_{emb}[:, k]) \ x_{i,k}, \tag{4.2}$$

where α_i is the attention weight on the *i-th* visit, β_i is an attention weight vector for all features x_i, of the *i-th* visit, W is the output weight matrix, W_{emb} is the weight matrix at the embedding layer, and $\odot$ denotes element-wise multiplication. $\omega_{i,k}$ is the corresponding contribution to the input $x_{i,k}$. Therefore, we can obtain the contribution matrix ω using all $\omega_{i,k}$.

4.3.2 Interpretation Attack Formulation

Given a patient record X, the goal is to find a new perturbed record $\widetilde{X}$ that is similar to the original record X both in input space and class predictions but with distorted interpretations. The attack can either be targeted, where we try to make the interpretations of $\widetilde{X}$ closer to a new explanation $\omega^\dagger$, or untargeted, where we attempt to change the interpretations to be far from those of X. Here we aim for a targeted attack and formulate the interpretation adversarial attack as below:

$$\begin{aligned} \min_{\widetilde{X}} \quad & \alpha \cdot \|\Phi_f(\widetilde{X}) - \omega^\dagger\| + \gamma \cdot \|\widetilde{X} - X\|_1 + \\ & \beta \cdot (\max\{Logit(\widetilde{X})_i : i \neq c^*\} - Logit(\widetilde{X})_{c^*})^+ \end{aligned} \tag{4.3}$$

where $(r)^+$ represents $max(r, 0)$, *Logit* is the outcome of the neural network before the Softmax layer and $\widetilde{X}$ is the adversarial example resulting in misleading interpretations. α, β and γ are the coefficients to balance the impact of the loss function terms. We will discuss each term one by one:

1. **Interpretation Loss**: The first term ensures that the interpretations of $\widetilde{X}$ resemble the targeted interpretation $\omega^\dagger$. This attack can be reformulated as an untargeted attack by replacing the current term with $-\|\Phi_f(\widetilde{X}) - \Phi_f(X)\|$. In

the case of the targeted attack, $\omega^{\dagger}$ can come from another set of interpretations with different but still realistic phenotypes, such as the interpretations of a randomly-selected patient, or patients' average interpretations of a different class than the X's class c^*. Since this leads to a more realistic scenario we proceed with targeted attacks.

2. **Perturbation Loss**: To keep the adversarial perturbations small, we regularize the perturbations using L_1 norm rather than renowned L_2-norm or L_∞-norm attacks. L_1 norm for adversarial attacks on EHR data are more meaningful for several reasons: First, EHR data are sparse, where many of the values are either zero or imputed and hence do not carry much information. Second, unlike images, different medical attributes carry different influences and weights on the output. Consequently, L_1 norm is suitable to meet both sparsity and heterogeneity of the EHR data [3, 83].

3. **Classification Loss**: Our interpretation method is non-post-hoc, so the predictions are highly tied to the interpretations. Thus we need a powerful function to keep the class of $\widetilde{X}$ unchanged. We employ this function for that purpose since it can be well optimized for manipulating the class predictions, especially for non-linear objective $f(\widetilde{x}) = c^*$ [13]. We start with this term and in Section 4.3.4 will show that it can be improved so that the output space $Logit(\widetilde{X})$ resembles $Logit(X)$ and hence helps the adversarial example remain undetectable.

4.3.3 Optimization with Dynamic Penalty

Equation 4.2 denotes how the parameters of the model, including weights and attributions, are directly involved in the explanations of the input. We observed that in some cases, the objective to change in interpretations might lead to a different class label. Given that the interpretation attack is conducted using a gradient descent

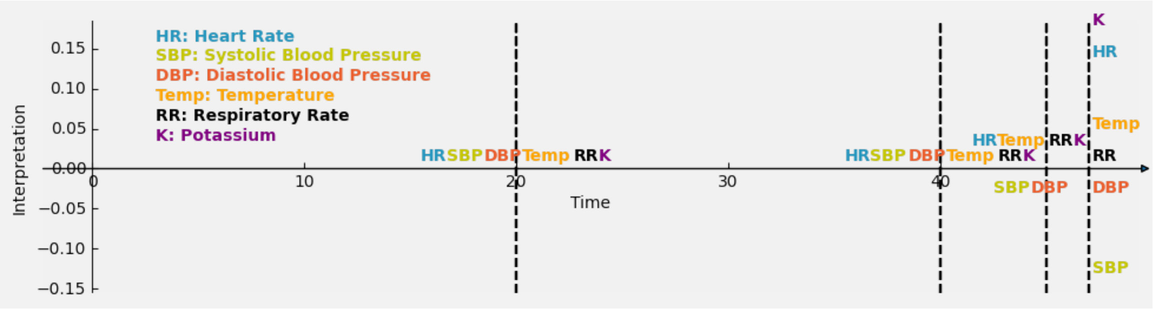


Figure 4.1: **Interpretations of a patient's EHR data for six attributes *(RR, HR, K, SBP, DBP, Temp)* with heart failure at the final time-stamp.** Interpretations of different attributes can be compared with each other in each specific time stamp. Also each attribute separately can be explored for its changes across time. The interpretations for EHR data generally gain more importance as the time of disease onset approaches.

algorithm, we use dynamic penalty for the interpretation and classification loss terms for preventing the prediction change.

Concretely, it involves adjusting the coefficient in Equation 4.3 to prioritize the objective of keeping the prediction label unchanged, i.e., incur a higher penalty whenever encountering a label change in any iteration. We can achieve this by decreasing α and increasing β by a factor (e.g. the factor is set to 2 in our implementation) until the original class label is attained. We can then continue using these coefficients for a few more steps to move away from the class boundaries. If this penalization process continues without successfully restoring the original class, the algorithm is considered to have failed. Algorithm 1 outlines the different components of the attack.

4.3.4 Minimizing Detectability

To carry out a stealthy attack, two aspects must be considered. The first is to keep the perturbations in the input space minimum, while the second is to maintain the integrity of the output space which includes the final class predictions and their associated logits. The reason is that many state-of-the-art defense methods for adversarial examples check changes both in the input feature space and the output logits space

Algorithm 1: Interpretation Attack on EHR

```
Function :  MINIMIZE-ATTACK-LOSS(.) : returns X and the
            corresponding Y by minimizing Eq. 4.3
Input :  initial clean sample ($X_{clean}, Y_{clean}$), initial coefficients ($\alpha_{init}, \beta_{init}$) in
         Eq. 4.3, number of iterations T, the maximum possible $\beta$ value
         $\beta_{treshold}$ and the number of extra steps for penalizing $steps_{extra}$
Initialize : $\alpha,\ \beta = \alpha_{init},\ \beta_{init}$
             $X_0,\ Y_0 = X_{clean},\ Y_{clean}$
for $t \in \{1, ..., T\}$ do
    $X_t,\ Y_t$ = MINIMIZE-ATTACK-LOSS($X_{t-1}$,$\alpha$,$\beta$)
    if $Y_t \neq Y_{clean}$ then        // Dynamically penalize the optimization
        while $Y_t \neq Y_{clean}$ do
            $\alpha,\ \beta = \alpha/2,\ \beta \times 2$
            $X_t,\ Y_t$ = MINIMIZE-ATTACK-LOSS($X_t$,$\alpha$,$\beta$)
            if $\beta > \beta_{threshold}$ then return Attack-failure;
        end
        for $s_e \in \{1, ..., steps_{extra}\}$ do
            $X_t,\ Y_t$ = MINIMIZE-ATTACK-LOSS($X_t$,$\alpha$,$\beta$)
        end
        $\alpha,\ \beta = \alpha_{init},\ \beta_{init}$
    end
end
Return $X_i$ from $\{X_1, ..., X_T\}$ with $Y_i = Y_{clean}$ and its interpretations have the
  least distance to the target interpretations (i.e. min $\|\Phi_f(X_i) - \omega^{\dagger}\|$)
```

[58, 87]. So in order to minimize the detectability, it is necessary to ensure that the logits do not change drastically during the attack. We observed that as we repeatedly apply and remove the penalty according to algorithm 1, it causes the output space of the adversarial example to oscillate near the classfication boundaries. Consequently, while the final label is the same as the original class, the logits do not resemble the original logits, nor does the confidence level of the adversarial prediction. This difference in logits, which we will refer to as output space, can be used to detect the attack.

To address this issue, we propose enhancing (4.3) by replacing the classification loss with two different alternatives; first we use the Kullback-Leibler divergence to directly compare the distribution of the original sample and adversarial example logits and by doing so keep them similar. We denote this divergence by $KL(Logit(X)||Logit(\widetilde{X}))$ (*KL* **attack**). Second, similar to the idea of C&W conventional adversarial attacks [13], we use $\max(\max\{Logit(\widetilde{X})_i : i \neq c^*\} - Logit(\widetilde{X})_{c^*}, -\kappa)$ where κ is a positive adjustable value and maintains a margin between the predicted logit and the second largest logit value to ensure high confidence in the predicted class (**Confident attack** parameterized by κ). Since the classifier is trained based on the clean examples' manifold, it can classify them with high confidence. So by ensuring high confident predictions for the adversarial examples, we can keep their logits similar to their original counterparts.

4.3.5 Metrics for Evaluation

For conventional adversarial examples, attack success rate (ASR) is measured as percentage of examples with flipped class labels. However, interpretation attacks aim to alter the multi-dimensional interpretation vector, making it difficult to establish a clear binary metric for measuring the success of the attack. In the subsequent discussion, we will outline two particular aspects of EHR data that must be taken into account when defining evaluation metrics.

In many application of EHR data, the interpretations may carry either positive or negative connotations, each with its unique significance. For example, when predicting the likelihood of a specific disease, the use of a particular medication may negatively affect the prognosis and decrease the chance of disease onset. For a clinician, the classifier's explanation of such a drug is no less important than the factors that indicate positive interpretations towards the prediction.

Another characteristic of EHR data is the heterogeneity and time sensitivity. Unlike pixels in images, the diverse attributes in EHR data hold distinct meanings, and clinician's interpretation may differ for each attribute. Additionally, the value of interpretations for clinicians is affected by the timing of attribute collection. Clinicians attach more significance to the data points that are closer to the disease onset. Figure 4.1 displays interpretations of some attributes calculated by RETAIN for predicting heart failure in a patient. Given these factors, we propose three metrics to evaluate the sucess of the interpretation attack, which consider the connotations of the interpretations, the attribute-level heterogeneity, and the visit-level time awareness.

Signed top-K intersection: According to Ghorbani et al. [31], in many cases, when interpreting a model, the explanations of the most important features are often of interest. In a gradient-based saliency map, the top-K features are determined by their magnitudes. Here we involve the connotation of the interpretations and assess the success of the attack by comparing the proportion of top-K features with consistent signs before and after the attack. So if $A = \{a_1, ..., a_k\}$ and $B = \{b_1, ..., b_k\}$ are the sets of the K largest absolute-value dimensions of $\Phi(\widetilde{X})$ and $\Phi(X)$ respectively, and $C = A \cap B$, then we have

$$topK = |\ \{c_i \in C : \Phi(\widetilde{X})_{c_i} * \Phi(X)_{c_i} > 0\}\ |. \tag{4.4}$$

Asymmetrical signed top-K intersection: Since the EHR is sequential and time-sensitive, the importance of different attributes are comparable separately in each timestamp that they are collected. To reflect that, we suggest a new metric that measures the top-K salient features in corresponding multivariate time series at each time point and then aggregate them.

Also, we assign weight ϕ_i to each time to better attain the perspectives of clinicians who may place greater emphasis on certain times. These weights can be achieved by background knowledge (e.g., higher weight on certain time points before the disease onset) or approximated by how the interpretable model weight different times, e.g., by taking 100 random samples from the clean data and summing up their interpretation values of all attributes at any given time. The resulting values are averaged over all samples to derive the weight that should be assigned to that specific time. For time $t_i \in \{1, \ldots, t\}$, we denote $A_{t_i} = \{a_j^{t_i}\}_{j=1}^k$ and $B_{t_i} = \{b_j^{t_i}\}_{j=1}^k$ the sets of the K largest absolute-value dimensions of $\Phi(\widetilde{X}_{t_i})$ and $\Phi(X_{t_i})$, respectively, and their intersectino as $C_{t_i} = A_{t_i} \cap B_{t_i}$.

$$topK\_asym = \sum_{i=1}^{t} \phi_i * topK(C_i). \tag{4.5}$$

Wasserstein distance: The Wasserstein distance measures the cost of moving a variable mass and is well-suited for comparing changes in time series. Its ability to capture perturbations has made it increasingly popular in the context of adversarial examples. We use the Wasserstein distance to measure the changes of contribution by each attribute as time series - since the modality of data is different across different attributes as discussed before. The resulting distances are then summed to obtain the final Wasserstein distance. Given attribute index $d_j \in [d]$, we denote $X_{[t]}^{d_j}$ as the sequential values of a specific attribute, and $Wass$ as the Wasserstein distance. Then, we calculate the final distance as:

$$Wass\_dist = \sum_{j=1}^{d} W_1(\Phi(\widetilde{X}^j_{[t]}), \Phi(X^j_{[t]})). \tag{4.6}$$

where W_1 denotes 1-Wasserstein distance for one dimensional data.

To make Equations 4.4, 4.5 and 4.6 consistent to our targeted attack, we calculate these relative metrics:

$$topK^{targeted} = topK(\Phi(\widetilde{X}_i), \omega^\dagger_i)/topK(\Phi(\widetilde{X}_i), \Phi(X_i)); \tag{4.7}$$

$$topK\_asym^{targeted} = topK\_asym(\Phi(\widetilde{X}_i), \omega^\dagger_i)/topK\_asym(\Phi(\widetilde{X}_i), \Phi(X_i)); \tag{4.8}$$

$$Wass\_dist^{targeted} = Wass\_dist(\Phi(\widetilde{X}_i), \omega^\dagger_i)/Wass\_dist(\Phi(\widetilde{X}_i), \Phi(X_i)). \tag{4.9}$$

These three new metrics not only measure how the adversarial interpretations are distant from the original ones, but also reflect how they resemble the target interpretations $\omega^\dagger$. The attacks with larger $topK^{targeted}$ and $topK\_asym^{targeted}$, and with smaller $Wass\_dist^{targeted}$ are more powerful. From now on, when we mention these metrics, we are specifically referring to their targeted version.

4.3.6 Robustness

To provide robustness, we propose using an RNN-based auto-encoder to de-noise the data and recover the original information. A typical auto-encoder comprises an encoder that compresses the data (X) into a smaller intermediate representation and a decoder that attempts to reconstruct the data (X') from those embeddings. Since temporal EHR data constitutes multivariate time series data, it is essential to construct a recurrent auto-encoder framework capable of capturing temporal and feature correlations. These auto-encoders utilize RNNs, such as stacked LSTM networks, as both their encoder and decoder (refer to Figure 4.2).

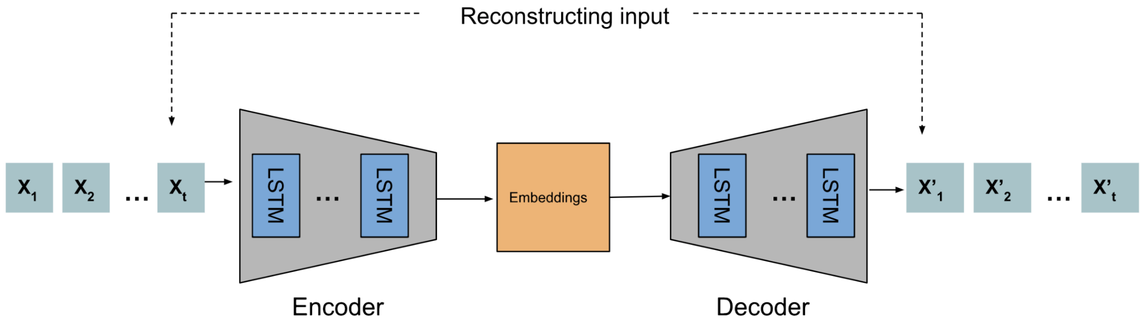


Figure 4.2: Structure of RNN auto-encoders.

As the encoder and decoder process the data, the output becomes de-noised. We train the auto-encoder on clean data so it learns the normal manifold. Hence, at inference time, it can remove the noise that caused the input data to become far from this manifold. As a result, the reconstruction error for rebuilding X' from X is high when X is an adversarial example compared to a clean normal sample. This aspect of auto-encoders previously was used to detect outliers. However, here we use the de-noised output of auto-encoders to defend against interpretation attacks. To do so, we utilize the interpretations of the decoder's output ($\Phi_f(X')$) instead of those of the input ($\Phi_f(X)$). Our results show that this approach leads to robust interpretations.

There are two reasons for this. Firstly, the EHR attack perturbations are sparse and have a greater magnitude wherever the features have notable interpretations. Therefore, the de-noiser can restore the original interpretations by reducing the large sparse perturbations on the salient features. Secondly, interpretation attacks differ from traditional adversarial examples in that they aim to modify smoothly distributed, high-dimensional interpretations, especially in EHR data. Once the de-noiser eliminates sudden, sparse perturbations, the interpretations can be regained by relying on the information present in the surrounding neighborhood.

We compare our method with SmoothGrad, a known and strong defense against interpretation attacks [80]. Although the attack in our case is gradient-free, the idea of SmoothGrad is still applicable. It involves adding noise to the data multiple times (usually 10 to 50) and averaging their contributions. However, this method is neither computationally efficient nor effectively provides robustness against EHR attacks.

4.4 Experiments

In this section, we will address these questions: 1) What is the effectiveness of the attack in altering the interpretations while maintaining the classification outcomes 2) Can existing defense methods against adversarial examples detect the interpretation attack? 3) How does the proposed de-noiser approach help with the robustness?

4.4.1 Experimental Setup

Dataset

The MIMIC-III dataset is a collection of electronic health records from thousands of patients in intensive care units. We use a dataset that was processed by [83] for the binary task of mortality prediction, resulting in 3177 positive samples and 30344 negative samples, each comprising 19 attributes across 48 timestamps. These features include vital signs and lab events such as heart rate, temperature, Creatinine, and Glucose, among others. Missing features were filled using the average value across all timestamps, and outliers were removed and imputed according to interquartile range criteria. Finally, each sequence was truncated or padded to 48 hours, and each feature was normalized using min-max normalization. The classification task is the binary task of predicting heart failure. We use 80% of the data for training and the rest for testing.

Model Architecture and Parameters

Adversarial examples were generated using RETAIN [19], which includes an embedding layer of size 128 and two GRU layers with 128 hidden units. Table 4.1 shows

Table 4.1: RETAIN's performance on the clean test set.

AUROC	AUPR	F1 Score	Accuracy
0.92	0.73	0.57	0.86

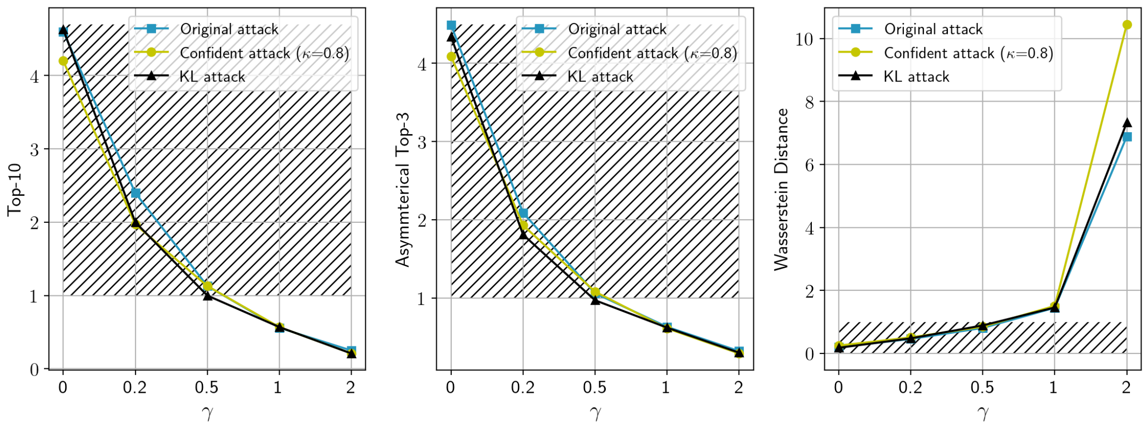
the model performance on clean unperturbed test data. We evaluate the detectability of the interpretation attack using RADAR [87]. It is a robust detection method, specifically developed for traditional EHR adversarial examples where the objective is to change the class. This detector identifies adversarial examples through both changes in input space and also output space relative to the normal manifold, making it well-suited for our purposes. Finally to enhance the data robustness, we de-noised data by the same auto-encoder architecture suggested in [87].

4.4.2 Attack Performance

Comparison of Attacks

We evaluate the attack performance based on three different metrics introduced in Section 4.3.5. We compare the original attack 4.3 with two alternatives, the KL attack and the Confident attack, proposed in Section 4.3.4. In our experiments with the Confident attack, we set $\kappa = 0.8$, as it provides a high level of undetectability. Our comparison is based on different values of the coefficient γ in Equation 4.3, which constrains the perturbation size. The higher the value of γ, the more restricted the attack is in terms of its distance from the original sample. Since the parameters α and β are dynamically ajusted by algorithm 1, we simply select their initial values as 1.

Figure 4.3 illustrates the results based on the three metrics (4.7, 4.8, 4.9) from left to right, respectively. The hatched area in each figure demonstrates the most desirable results. For Figures 4.3.a and b, a ratio of over 1 implies that the interpretations are more similar to the targeted interpretations than the original ones, and the larger the



(a) Top-K metric **(b)** Asymmterical targeted top-K metric **(c)** Wasserstein distance metric

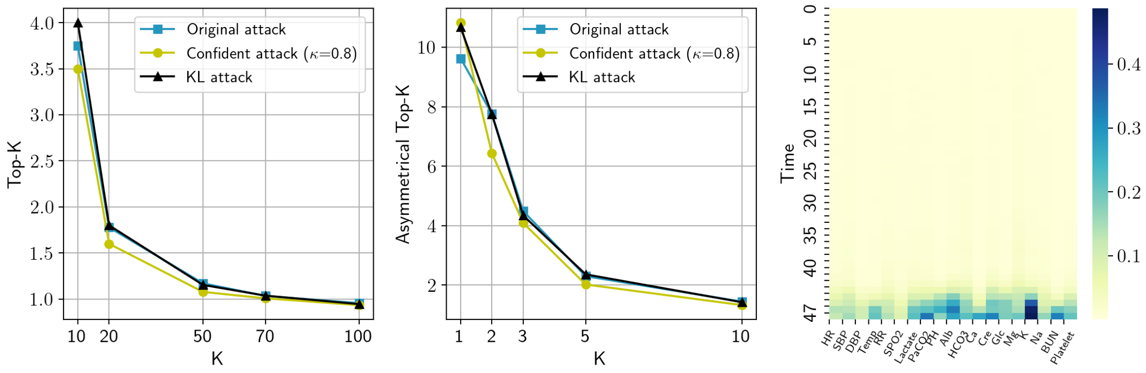
Figure 4.3: **The comparison of three interpretation attacks, which differ in their penalty term, shown using three metrics.** The desirable results are located in the hatched area. A lower perturbation achieved by a smaller γ leads to better attack success, but may also result in a higher detection rate.

ratio, the better. Conversely, in Figure 4.3.c, the opposite is true, as this measurement employs a distance metric rather than the intersection of salient features. Although the attacks are very similar, in the next section, we will show the main difference lies in the stealthiness of each of these attacks.

Selection of K

Figure 4.4 demonstrates that how the selection of K in top-K metrics (4.7 and 4.8) impacts our evaluation of the attack's success when $\gamma = 0$. In metric 4.4 since K is calculated in each time and over a lower dimension than the entire EHR data, we set the value of K to a lower number than in metric 4.8. As expected, the value of K affects the degree of overlap between interpretations before and after attack.

Figure 4.4.c shows the average perturbations of all the adversarial examples when γ is zero and there is no constraint on the input space. The perturbations are concentrated on the latest time-stamps which hold the most significant interpretations in the model and clinical environments. Therefore, selecting a large K does not yield significant interpretations, particularly since many interpretations that are distant



(a) Comparison in top-K

(b) Comparison in asymmetrical top-K

(c) Perturbations

Figure 4.4: **The comparison of different values of K in two metrics, top-K (a) and asymmetrical top-K (b).** The concentration of perturbations on the latest time-stamps (c) confirms that small values of K are sufficient for evaluation.

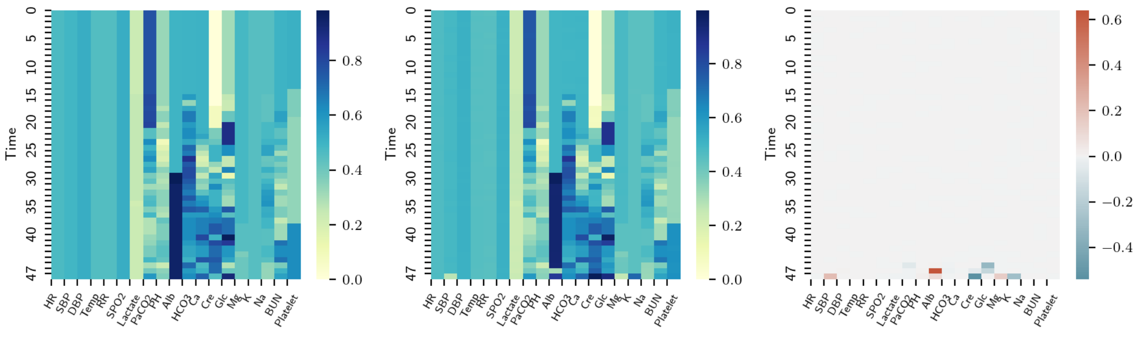
from these timestamps have close to zero. Consequently, considering large K results in overlapping interpretations that do not offer meaningful insights into the attack's success.

4.4.3 Attack Detectability

Figure 4.5 illustrates an example before and after the attack and their difference for the confident attack with $\gamma = 0.5$. The attack causes sparse but strong perturbations, which lead the interpretations to shift from the original to the target interpretations. As previously discussed, the low number of perturbations and their sparsity make them undetectable in EHR data. By decreasing γ, the magnitude and density of the perturbations become more flexible.

Figure 4.6 illustrates the interpretations of the original sample and its adversarial counterpart from Figure 4.5 as well as the target interpretations across the latest timestamps for six attributes. It reveals that the sparse perturbation attack caused the adversarial interpretations to deviate from their original values and align more closely with the target interpretations.

We evaluated RADAR to demonstrate whether our proposed interpretation at-



(a) Original sample **(b)** Adversarial sample **(c)** Perturbations

Figure 4.5: An example before (a), and after attack with γ=5 (b), and its additive perturbation (c).

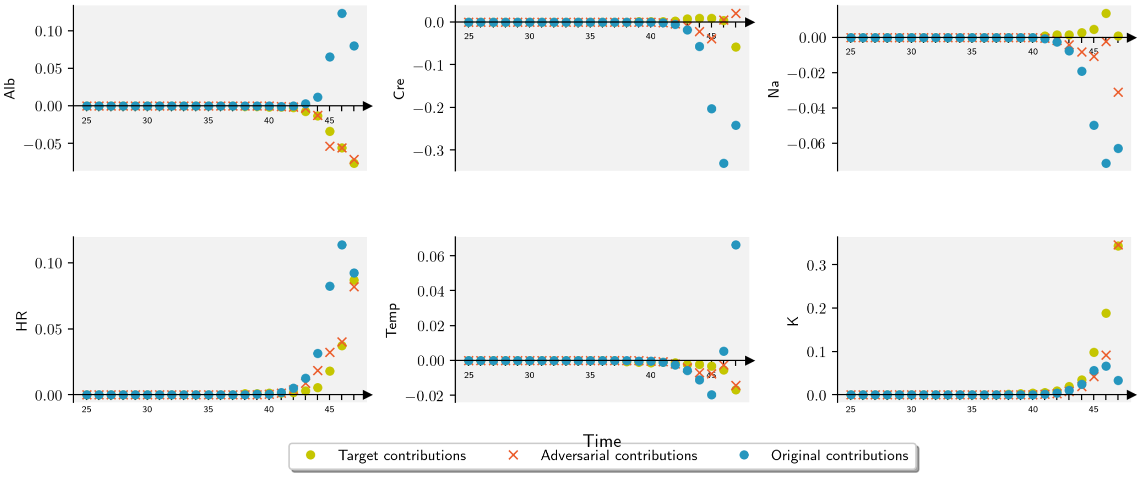


Figure 4.6: Comparison of Adversarial, original and target contributions (interpretations) of six attributes over time.

tacks can be detected by existing defense methods against conventional adversarial examples. RADAR exhibits a 100% detection rate for conventional adversarial examples on RETAIN. Our results shows the detection rate on our interpretation attack is in general low, compared to the high detection rate for adversarial examples by RADAR.

Figure 4.7 presents the detection percentage of different attacks by RADAR. As γ increases, the perturbations become smaller, resulting in a decrease in the detection rate in input space. Additionally, considering the detection in output space,, when γ is small, the adversarial example has more flexibility during optimization, allowing

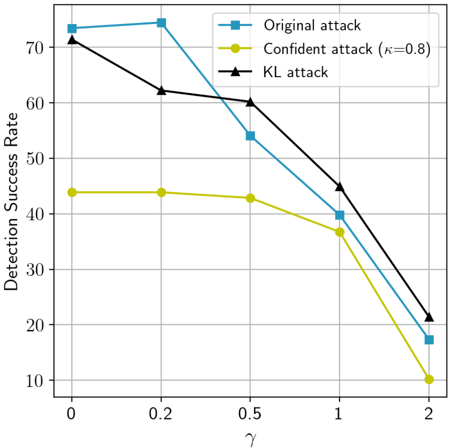


Figure 4.7: **The detection results of RADAR on the input and output spaces of various attacks.** Although the objective of interpretation attacks is more complex than conventional prediction attacks, detecting such attacks on EHR interpretations is more difficult.

it to approach the classification decision boundary more closely and activate the penalty process in algorithm 1 more frequently. In Section 4.3.4, we discussed how KL and confident attacks better maintain similarity between the original and output space in such cases. However, for larger values of γ, the original attack is less likely to trigger the penalty process and remains more stealthy than the KL attack. Generally, the confident attack keeps the output space less detectable and maintains a greater distance from the class boundary. Note that RADAR exhibits a 100% detection rate for conventional adversarial examples on RETAIN.

4.4.4 Robustness

In this section, we employ a robustness method by passing both the original and adversarial examples through the auto-encoder and comparing the interpretations of the corresponding outputs. We repeat this comparison for both the results of the attack itself and for the SmoothGrad method. We select the confident attack with $\kappa = 0.8$ and $\gamma = 0.5$ as the representative of successful attacks with reasonably high success rate and low detection rate. For SmoothGrad, the best results are reported based on selecting a noise level of 0.1 and calculating the average over 50 samples,

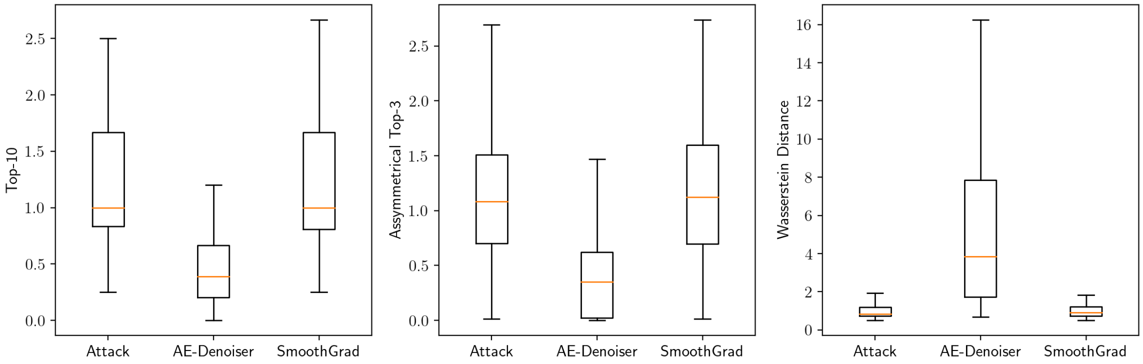


Figure 4.8: **Robustness of de-noising method vs. SmoothGrad based on three metrics.** All figures show the de-noising method outperforms SmoothGrad.

which is consistent with the result of the main paper [80].

Figure 4.8 displays a comparison of the median and quartile charts of the attack versus the robustness achieved through the de-noising method and SmoothGrad for 100 samples. Smaller values for top-K and asymmetric top-K indicate better robustness, whereas higher values for Wasserstein distance indicate better robustness. As depicted in the figure, the de-noising method outperforms SmoothGrad in all metrics.

Chapter 5

Conclusion

This dissertation addresses the critical concern of data security in machine learning. The research proposes approaches to enhance the robustness of machine learning models against malicious attacks that manipulate training data and defending against poisoning and backdoor attacks. Additionally, the dissertation suggests a defense strategy aginst interpretation attacks for EHR. Overall, the research contributes to the development of more reliable and trustworthy machine learning systems, with implications for a range of applications. The results of the research demonstrate promising outcomes in providing data security.

5.1 Summary

Chapter 2 utilized auto-encoders to defend against various types of poisoning attacks for the first time. We proposed CAE, a two-component auto-encoder that enjoys an auxiliary classification layer 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.

Chapter 3 posed important questions regarding the ability of DP to provide robustness against backdoor attacks in practice. In addition to DP-SGD, we explored the other commonly used DP algorithm (PATE) and two Label-DP algorithms (LP-2ST and ALIBI) for the first time for this purpose. We have several main findings. First, the noise and randomness added to the private models can indeed decrease the attack success rate of the backdoors, but at the cost of utility drop for clean input. In a nutshell, a model trained with privacy guarantee have inherent benefit in robustness against backdoor attacks. This statement holds for all four methods mentioned above. A somewhat unexpected outcome is that PATE delivers the best results, even without the use of noise (without DP guarantee).

Second, contrary to the claims of some previous studies, DP-SGD provides good resistance against backdoors while keeping the accuracy relatively high. We also observed the same phenomenon for Label-DP algorithms. The accuracy-ASR trade-off is diverse among the DP and Label-DP methods we analyzed. One model may outperform the others depending on the privacy budget, algorithm parameters, and attack specifications. Therefore it is possible to use DP models as defense strategies. A proper selection of the above mentioned factors can adequately balance the accuracy and ASR. This work was an empirical study on two benchmark datasets, MNIST and CIFAR-10. It offered new empirical understandings of the connection between DP and backdoor attacks in relation with existing theoretical understandings.

Chapter 4 was the first study to develop and adapt interpretation attacks for EHR data. We investigated various aspects of EHR data, including their heterogeneity and sparsity, as well as attribution-based models that are designed specifically for EHR data. Using this knowledge, we expanded the interpretation attack on EHR data by testing various loss functions and evaluating the attack using customized metrics that address EHR specifications. Our results show that with a good choice of loss function, 2/3 of the data can evade the detector RADAR, which is capable of detecting 100% of conventional adversarial examples. To counteract the attack, we suggested using a de-noiser and demonstrated that it successfully made the attacked interpretation closer to its clean counterpart, with a 0.4 improvement in the top-K measurement compared to SmoothGrad.

5.2 Future Work

While these chapters propose effective defense methods against specific types of attacks, future research could focus on developing more general defense strategies that are effective against multiple types of attacks (e.g. both training-time attacks and test-time attacks). This could involve investigating the underlying causes of vulnerability to different types of attacks and developing methods that address these underlying vulnerabilities.

Another area of future work could be to investigate the interaction between different types of attacks and defenses. For example, it would be interesting to explore whether using a combination of different [proactive and reactive] defense methods can provide better overall protection against attacks than using a single defense method.

Also, we can further generalize the proposed defense strategies by exploring their effectiveness on other models and datasets. For example, in the case of EHR interpretation attacks, it is beneficial to investigate the effectiveness of the proposed defense methods on a wider range of interpretable models. Similarly, in the case of poisoning attacks, investigating the effectiveness of the proposed defense methods on non-convex models could provide valuable insights.

By addressing these future research directions, we can continue to improve the robustness and security of machine learning models in the face of evolving attack methods.

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1), 2015.

[3] Sungtae An, Cao Xiao, Walter F Stewart, and Jimeng Sun. Longitudinal adversarial attack on electronic health records data. In *The world wide web conference*, pages 2558–2564, 2019.

[4] Galen Andrew, Steve Chein, and Nicolas Papernot. Tensorflow privacy library. https://github.com/tensorflow/privacy, 2020.

[5] Caglar Aytekin, Xingyang Ni, Francesco Cricri, and Emre Aksu. Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2018.

[6] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49, 2012.

[7] Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, and Jaehoon Amir Safavi. Mitigating poisoning attacks on machine learning models: A data provenance based approach. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 103–110, 2017.

[8] Battista Biggio, Igino Corona, Giorgio Fumera, Giorgio Giacinto, and Fabio Roli. Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In *International workshop on multiple classifier systems*, pages 350–359. Springer, 2011.

[9] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[10] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. 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)*, pages 3855–3859. IEEE, 2021.

[11] Eitan Borgnia, Jonas Geiping, Valeriia Cherepanova, Liam Fowl, Arjun Gupta, Amin Ghiasi, Furong Huang, Micah Goldblum, and Tom Goldstein. Dp-instahide: Provably defusing poisoning and backdoor attacks with differentially private data augmentations. *arXiv preprint arXiv:2103.02079*, 2021.

[12] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. In *ICML TPDP workshop*, 2022.

[13] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[14] Javier Carnerero-Cano, Luis Muñoz-González, Phillippa Spencer, and Emil C Lupu. Regularisation can mitigate poisoning attacks: A novel analysis based on multiobjective bilevel optimisation. *arXiv preprint arXiv:2003.00040*, 2020.

[15] Jian Chen, Xuxin Zhang, Rui Zhang, Chen Wang, and Ling Liu. De-pois: An attack-agnostic defense against data poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 16:3412–3425, 2021.

[16] Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. Robust attribution regularization. *Advances in Neural Information Processing Systems*, 32, 2019.

[17] Peipei Chen, Wei Dong, Jinliang Wang, Xudong Lu, Uzay Kaymak, and Zhengxing Huang. Interpretable clinical prediction via attention-based neural network. *BMC Medical Informatics and Decision Making*, 20(3):1–9, 2020.

[18] Ruoxin Chen, Zenan Li, Jie Li, Junchi Yan, and Chentao Wu. On collective robustness of bagging against data poisoning. In *International Conference on Machine Learning*, pages 3299–3319. PMLR, 2022.

[19] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems*, 29, 2016.

[20] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[21] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be ma-

nipulated and geometry is to blame. *Advances in neural information processing systems*, 32, 2019.

[22] Ann-Kathrin Dombrowski, Christopher J Anders, Klaus-Robert Müller, and Pan Kessel. Towards robust explanations for deep neural networks. *Pattern Recognition*, 121:108194, 2022.

[23] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *ICLR 2020*, 2020.

[24] C. Dwork. Differential privacy. volume 2006, pages 1–12. ICALP, 2006. URL https://link.springer.com/chapter/10.1007/11787006_1.

[25] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 486–503. Springer, 2006.

[26] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[27] Enrique Estellés-Arolas and Fernando González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2): 189–200, 2012.

[28] Minghong Fang, Minghao Sun, Qi Li, Neil Zhenqiang Gong, Jin Tian, and Jia Liu. Data poisoning attacks and defenses to crowdsourcing systems. In *Proceedings of the Web Conference 2021*, pages 969–980, 2021.

[29] Jie Geng, Jianchao Fan, Hongyu Wang, Xiaorui Ma, Baoming Li, and Fuliang

Chen. High-resolution sar image classification via deep convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters*, 12(11):2351–2355, 2015.

[30] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. *Advances in Neural Information Processing Systems*, 34, 2021.

[31] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.

[32] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[33] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7: 47230–47244, 2019.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[35] Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *arXiv preprint arXiv:2002.11497*, 2020.

[36] Matthew Jagielski and Alina Oprea. Does differential privacy defeat data poisoning. In *DPML Workshop*, 2021.

[37] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and

countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

[38] Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7961–7969, 2021.

[39] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

[40] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

[41] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *Machine Learning*, pages 1–47, 2021.

[42] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[43] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[44] Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo. Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. *IEEE transactions on visualization and computer graphics*, 25 (1):299–309, 2018.

[45] Ricky Laishram and Vir Virander Phoha. Curie: A method for protecting svm classifier from poisoning attack. *arXiv preprint arXiv:1606.01584*, 2016.

[46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.

[47] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[48] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[49] Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):1–12, 2020.

[50] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3): 31–57, 2018.

[51] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[52] Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 647–656, 2020.

[53] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirec-

tional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1903–1911, 2017.

[54] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 4732–4738. AAAI Press, 2019. ISBN 9780999241141.

[55] Pooria Madani and Natalija Vlajic. 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*, pages 1–8, 2018.

[56] Mani Malek Esmaeili, Ilya Mironov, Karthik Prasad, Igor Shilov, and Florian Tramer. Antipodes of label differential privacy: Pate and alibi. *Advances in Neural Information Processing Systems*, 34, 2021.

[57] Marco Melis, Ambra Demontis, Maura Pintor, Angelo Sotgiu, and Battista Biggio. secml: A python library for secure and explainable machine learning. *arXiv preprint arXiv:1912.10013*, 2019.

[58] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147, 2017.

[59] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[60] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pages 2574–2582, 2016.

[61] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38, 2017.

[62] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D Joseph, Benjamin IP Rubinstein, Udam Saini, Charles A Sutton, J Doug Tygar, and Kai Xia. Exploiting machine learning to subvert your spam filter. *LEET*, 8:1–9, 2008.

[63] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[64] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[65] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

[66] Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C Lupu. Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv preprint arXiv:1802.03041*, 2018.

[67] Andrea Paudice, Luis Muñoz-González, and Emil C Lupu. Label sanitization against label flipping poisoning attacks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 5–15. Springer, 2018.

[68] Gwenolé Quellec, Katia Charriere, Yassine Boudi, Béatrice Cochener, and Mathieu Lamard. Deep image mining for diabetic retinopathy screening. *Medical image analysis*, 39:178–193, 2017.

[69] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):1–13, 2021.

[70] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965, 2020.

[71] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

[72] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, pages 9389–9398. PMLR, 2021.

[73] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[74] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. In Sarit Kraus, editor, *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*, IJCAI International Joint Conference on Artificial Intelligence, pages 5953–5959. International Joint Conferences on Artificial Intelligence, 2019. doi: 10.24963/ijcai.2019/825.

[75] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021.

[76] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *IEEE Symposium on Security and Privacy*, 2022.

[77] Shiqi Shen, Shruti Tople, and Prateek Saxena. Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519, 2016.

[78] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[79] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[80] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[81] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pages 3517–3529, 2017.

[82] Jingwei Sun, Ang Li, Louis DiValentin, Amin Hassanzadeh, Yiran Chen, and Hai Li. Fl-wbc: Enhancing robustness against model poisoning attacks in federated learning from a client perspective. *Advances in Neural Information Processing Systems*, 34, 2021.

[83] Mengying Sun, Fengyi Tang, Jinfeng Yi, Fei Wang, and Jiayu Zhou. Identify susceptible locations in medical records via adversarial attacks on deep predictive models. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 793–801, 2018.

[84] Farnaz Tahmasebian, Li Xiong, Mani Sotoodeh, and Vaidy Sunderam. Crowdsourcing under data poisoning attacks: A comparative study. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 310–332. Springer, 2020.

[85] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[86] Lulu Wang, Junxiang Zheng, Yongzhi Cao, and Hanpin Wang. Enhance pate on complex tasks with knowledge transferred from non-private data. *IEEE Access*, 7:50081–50094, 2019.

[87] Wenjie Wang, Pengfei Tang, Li Xiong, and Xiaoqian Jiang. Radar: Recurrent autoencoder based detector for adversarial examples on temporal ehr. In

Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 105–121. Springer, 2020.

[88] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309): 63–69, 1965.

[89] Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In *ECAI*, pages 870–875, 2012.

[90] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[91] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning*, pages 1689–1698, 2015.

[92] Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.

[93] Chen Xing, Li Ma, and Xiaoquan Yang. Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. *Journal of Sensors*, 2016, 2016.

[94] Chang Xu, Jun Wang, Francisco Guzmán, Benjamin Rubinstein, and Trevor Cohn. Mitigating data poisoning in text classification with differential privacy. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4348–4356, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.369. URL https://aclanthology.org/2021.findings-emnlp.369.

[95] Yanbo Xu, Siddharth Biswal, Shriprasad R Deshpande, Kevin O Maher, and Jimeng Sun. Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In *Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*, pages 2565–2573, 2018.

[96] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

[97] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. Interpretable deep learning under fire. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.

[98] Mengchen Zhao, Bo An, Wei Gao, and Teng Zhang. Efficient label contamination attacks against black-box learning models. In *IJCAI*, pages 3945–3951, 2017.

[99] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452, 2020.

[100] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017.