Computers, Materials & Continua
DOI:10.32604/cmc.2022.019709
Article

Tech Science Press

Towards Securing Machine Learning Models Against Membership Inference Attacks

Sana Ben Hamida[1,2], Hichem Mrabet[3,4], Sana Belguith[5,*], Adeeb Alhomoud[6] and Abderrazak Jemai[7]

[1]Departement of STIC, Higher Institute of Technological Studies of Gabes, General Directorate of Technological Studies, Rades, 2098, Tunisia
[2]Research Team on Intelligent Machines, National Engineering School of Gabes, Gabes University, Gabes, 6072, Tunisia
[3]SERCOM-Lab., Tunisia Polytechnic School, Carthage University, Tunis, 1054, Tunisia
[4]Department of IT, College of Computing and Informatics, Saudi Electronic University, Medina, 42376, Saudi Arabia
[5]School of Science, Engineering and Environment, University of Salford, Manchester, M5 4WT, UK
[6]Department of Science, College of Science and Theoretical Studies, Saudi Electronic University, Riyadh, 11673, Saudi Arabia
[7]INSAT, SERCOM-Lab., Tunisia Polytechnic School, Carthage University, Tunis, 1080, Tunisia
[*]Corresponding Author: Sana Belguith. Email: S.Belguith@salford.ac.uk
Received: 23 April 2021; Accepted: 20 June 2021

Abstract: From fraud detection to speech recognition, including price prediction, Machine Learning (ML) applications are manifold and can significantly improve different areas. Nevertheless, machine learning models are vulnerable and are exposed to different security and privacy attacks. Hence, these issues should be addressed while using ML models to preserve the security and privacy of the data used. There is a need to secure ML models, especially in the training phase to preserve the privacy of the training datasets and to minimise the information leakage. In this paper, we present an overview of ML threats and vulnerabilities, and we highlight current progress in the research works proposing defence techniques against ML security and privacy attacks. The relevant background for the different attacks occurring in both the training and testing/inferring phases is introduced before presenting a detailed overview of Membership Inference Attacks (MIA) and the related countermeasures. In this paper, we introduce a countermeasure against membership inference attacks (MIA) on Conventional Neural Networks (CNN) based on dropout and L2 regularization. Through experimental analysis, we demonstrate that this defence technique can mitigate the risks of MIA attacks while ensuring an acceptable accuracy of the model. Indeed, using CNN model training on two datasets CIFAR-10 and CIFAR-100, we empirically verify the ability of our defence strategy to decrease the impact of MIA on our model and we compare results of five different classifiers. Moreover, we present a solution to achieve a trade-off between the performance of the model and the mitigation of MIA attack.

Keywords: Machine learning; security and privacy; defence techniques; membership inference attacks; dropout; L2 regularization

1 Introduction

Artificial intelligence and machine learning (ML) make the headlines not only in scientific journals but also in our daily life, an upscale debate on its advances and evolution is highlighted. ML makes it possible, through algorithms, to analyse large amounts of data and provide answers to challenging problems. The importance of ML technology has been recognized by companies across a number of industries that deal with huge volumes of data. ML is used in several domains such as financial services, marketing and sales, government, healthcare, transport, Internet of Things and smart manufacturing [1,2]. Indeed, with the help of machine learning models, companies in the financial sector, for example, can predict changes in the market and even able to prevent any occurrence of financial fraud. ML technology can be also used to analyse the purchase history of customers to generate personalised recommendations for their next purchase. ML is also becoming a trend in healthcare thanks to the evolution of wearable sensors and devices to collect data from patients in real time [1]. It also empowers experts by tools that help providing better diagnostics and treatments proposal.

Despite their wide applications, ML models present various security and privacy issues. Research works have identified different attacks to leak the privacy of the data used in the models, to inject false, or to impact the output of the model. Attacks on ML can be classified according to whether they occur during training or testing/inferring stage [3]. Most known attacks against ML models are poisoning, evasion, impersonate, inversion and inference attacks [4–8]. Poisoning attacks consist on injecting adversarial samples to the training data in order to alter the model prediction. Evasion attacks occur when a conflicting sample is injected in the network to impact the accuracy of the classifier. This injected malicious sample is a carefully disrupted input that looks and feels exactly the same to a human as its unaltered copy. Impersonate attack is a form of fraud in which adversary imitates data samples from victims to pass as trusted person to dupe the model. Inversion attacks try to infer some features about a hidden model input by looking at the model output. Inference attacks target a model to determine whether a data sample was used in the training data set by only looking at the output.

Malicious adversaries increasingly run attacks on ML models to execute automated large-scale inference attacks [5]. An inference attack is an attack based on extracting and discovering patterns by analysing output data in order to illegitimately gain knowledge about the training dataset. It is a type of attack in which user sensitive information is inferred by the data disclosed by the user and used to train the model.

In this paper, we focus on executing membership inference attacks (MIA) and we propose an efficient mitigation technique to reduce the impact of these attacks. For instance, MIA seek to infer whether a data sample was included in the training datasets used to train the model. These attacks can be successful due to the fact that private data are statistically correlated with public data, and ML classifiers can capture such statistical correlations. Knowing that a data sample was used to train a model can lead to a privacy breach. For instance, in medical use cases, inferring that a patient record was used to train a ML model that is designed to predict the existence of a disease and its causes or to propose a suitable medication, can reveal that this patient is suffering from this disease.

The purpose of this research is to study the different vulnerabilities of ML models and to propose techniques to improve the security and privacy of such models especially against membership inference attacks (MIA).

Contributions: In this paper, we design a solution to protect the datasets used to train a machine learning model, against membership inference attacks. The proposed solution aims to train machine learning models while ensuring membership privacy. By using this countermeasure, adversaries should not be able to distinguish between the prediction of the model on its training dataset and other data samples which are not used on the training dataset. Our solution aims to achieve membership privacy while ensuring an acceptable level of accuracy of the model.

Various research works have identified overfitting as the main cause leading to a successful membership inference attack [8,9]. Overfitting occurs when the model is overtrained on the training component of the dataset, such that when the model encounters different data, it gives worse results than expected. Therefore, the proposed solution is developed based on the combination of dropout and regularization techniques to avoid overfitting.

In this paper, we first implement and test Membership Inference Attacks on Conventional Neural Networks (CNN) model. Afterwards, we have tested our proposed defence technique to show its effectiveness in improving the security of the model against MIA attacks. We show that we have enforced the security of ML models by decreasing the model overfitting and we evaluated the effectiveness of using L2 regularization and dropout as defence techniques to mitigate the overfitting of the model which is the main cause of the leakage of information related to the training dataset. We have tested our solution using CNN model trained on two datasets CIFAR-10 and CIFAR-100. Our evaluation showed that our defence technique is able to reduce the privacy leakage and mitigate the impact of the membership inference attacks. However, experimental results showed that the accuracy of the target model has been decreased when the privacy of the data is achieved. Therefore, we have proposed a trade-off between the privacy preservation of the model and its performance.

The paper is structured as follows. In Section 2, we briefly introduce the background of machine learning. In Section 3, we review different attacks on ML, before detailing membership inference attacks in Section 4. Next, we present state-of-the-art defence techniques against different attacks on ML models in Section 5. The experimental setup and results are reported in Section 6. In Section 7, we review related works before concluding in Section 8.

2 Machine Learning Background

ML techniques are usually divided into three classes, characterized by the nature of the data available for analysis: supervised learning, unsupervised learning and reinforcement learning.

2.1 Supervised Learning

This is the most recurrent type, it provides learning algorithms with a training set in the form of (X, Y) with X the predictor variables, and Y the result of the observation. Based on the training set, the algorithm will find a mathematical function that transforms (at best) X into Y.

We can divide supervised learning into two categories:

- Classification: this type of algorithms is used to predict a discrete variable, the output variable is a category, for example gender (male or female). For example, using a dataset of human being photos, each photo is labelled as male or female. At this point the algorithm has to classify the new images into one of these two categories. Some examples of these algorithms are Naive Bayes (NB), Support Vector Machines (SVM) and Logistic Regression (LR).

- Regression: this supervised learning category is used for continuous data. The output variable is a specific value. For instance, it can be used in predicting the price of a house given input criteria such as the area, location, and number of rooms. Examples of regression problems include Linear Regression (LR), Nonlinear Support Vector Machine (SVR) and Bayesian Linear Regression (BLR).

2.2 Unsupervised Learning

In this type, the algorithm takes as input unlabelled data. The algorithm should find possible correlation between given data. In short, there is no complete and clean dataset used as input, unsupervised learning is a self-organized type of learning. This approach is called feature learning. For example, by having the purchasing data of Internet users in an e-commerce site, the clustering algorithm will find the products that sell best together. Unsupervised learning models include K-means, DBSACAN and C-means clustering. There are two types of unsupervised learning, clustering where the purpose is to discover clusters in the data, and association which aims is to identify the rules that will define large groups of data.

2.3 Reinforcement Learning

In reinforcement learning, the model interacts with a dynamic environment in which it must achieve a certain goal, for example driving a vehicle or facing an adversary in a game. The apprentice program receives feedback in the form of "rewards" and "punishment" while navigating the space of the problem and learning to identify the most effective behaviour in the context. This ML method is used in particular for training the models on which autonomous vehicles are based. These models can be trained in a virtual environment such as a car simulation, in order to teach them to respect the Highway Code.

Tab. 1 presents a synthesis of the differences between ML classes based on various criteria such as definition, type of data, type of problems, examples of algorithms and the target. The research conducted in this paper focuses on supervised learning.

Table 1: Comparison of different classes of Machine Learning

Criteria	Supervised learning	Unsupervised learning	Reinforcement learning
Definition	Learns by using labelled data	Trained using unlabelled data without any control to find correlation between data	Works on interacting with the environment
Type of data	Labelled data	Unlabelled data	No-predefined data
Type of problems	Regression and classification	Association and clustering	Exploitation or exploration
Examples of algorithms	LR, LR, SVM, KNN.	K-means, C-means, DBSCAN.	Q-learning, SARSA
Target	Calculate outcomes	Discover underlying patterns	Learn a series of action

ML follows a cyclic life-cycle process. The life cycle's main aim is to find a solution to the studied problem, it includes seven important steps: data gathering, data preparation, data

wrangling, data analysis, model training, model testing, and model deployment [8]. These life-cycle steps can be grouped in two phases: training and testing/inferring phase.

3 Security and Privacy Attacks on Machine Learning Models

Different classifications of attacks on ML have been introduced in the literature. Based on the technical level, attacks can occur on two different stages: during training or testing/inferring stage [3]. Chen et al. [7] classify attacks according to knowledge restriction. Indeed, adversaries may have different restrictions in terms of the information about a target system, i.e., Black-box and White-Box. In Black-box attack model, the adversary can only send a request to the system and obtains a simple result, he does not know any information about the training set or the model. However, in White-box system everything is known such as weights and data on which this network was trained.

Yeom et al. [9] classify the attacks as being either Causative or Exploratory. Causative attacks affect the training data. However, exploratory attacks strike the model at test time.

Another attacks classification can be done according to the real target of the attacker, which can involve espionage, sabotage and fraud. Attacks on ML cover evasion, poisoning, trojaning, backdooring, reprogramming, and inference attacks [10]. Tab. 2 presents classification of attacks depending on the stage of ML and the goal of the attacker.

Table 2: Categories of attacks on ML models

Stage	Espionage	Sabotage	Fraud
Training	Inference by poisoning	Poisoning Trojaning Backdooring	Poisoning
Testing	Inference attacks	Adversarial reprogramming Evasion (False negative evasion)	Evasion (False positive evasion)

Liu et al. [3] range machine learning security issues according to two criteria depending on whether the attack has been conducted in the training or testing/inferring phases. The authors present a summary of different security threats on ML:

- Poisoning attack: is a type of causative attack aiming to impact the model availability and integrity by injecting malicious data samples to the training data set which distorts the model predictions.
- Evasion attack: in this attack, samples are changed at the inferring phase to evade detection.
- Impersonate attack: this attack consists in imitating data samples from victims. This attack occurs in use case applications involving image and text recognition.
- Inversion attack: this attack aims to gain knowledge about a hidden model input by looking at the model output.

3.1 Poisoning Attack

Poisoning attack is a security threat occurring during the training phase. Papernot et al. [11] define poisoning attacks as an injection of false data in the training dataset by the adversary. In order to do this, the adversary extracts and injects some data to reduce the precision of the

classification. This attack has the potential to totally distort the classification mechanism during its training so that the attacker can in some way define the classification of the system. The magnitude of the classification error depends on the information that the attacker has chosen to poison the training data.

3.2 Evasion Attack

Evasion may be the most frequent attack on machine learning models performed during production. According to Polyakov [5], evasion attack aims at designing an input that appears normal for a person but is wrongly classified by ML models. A common example is to vary some pixels in the image before uploading, where the image recognition system fails to classify the result.

3.3 Impersonate Attack

Polyakov [5] define the impersonate attack as the fact of imitating data samples, particularly in application scenarios of image recognition, malware detection, and intrusion detection. Specifically, the goal of such attack is to obtain specific conflicting samples so that machine learning models outputs a wrong classification of the samples with different labels than those borrowed.

3.4 Inversion Attack

Liu et al. [3] define an inversion attack as an attack aiming at gathering basic information about a target system model. This basic information will be then used in a reverse analysis that target revealing the model data input such as images, medical records, purchase patterns, etc.

3.5 Inference Attack

An inference attack is an attack based on extracting and discovering patterns by analysing output data in order to illegitimately gain knowledge about the training dataset [12]. It is a type of attack in which user sensitive information is inferred by the data disclosed by the user and used to train the model.

4 Membership Inference Attack

Membership Inference Attacks (MIA) are detailed according to definitions introduced different research works presented in the literature [10,11,13–15].

Membership Inference Attacks (MIA) were presented at the first time by Shokri et al. in 2017 [8]. MIA consists of quantifying how much information a machine learning model leaks about its training data, which could contain personal and sensitive information. The proposed mechanism examines the predictions made by machine learning model to determine whether a particular data record was used in its training set [8]. The susceptibility to this form of attack stems from the tendency for models to respond differently to inputs that were part of the training dataset. This behaviour gets worse when models are over-adapted to the training data. An overfitted model learns external noise that is only present in the training dataset. When this occurs, the model makes very good predictions about training data records, while records from outside of the data collection can generate poorer predictions. These predictions from training set and non-training set data records generate two distributions that are learned by the attack model.

The lifecycle of the membership inference attack from training to testing is summarized in the following steps presented on Fig. 1.

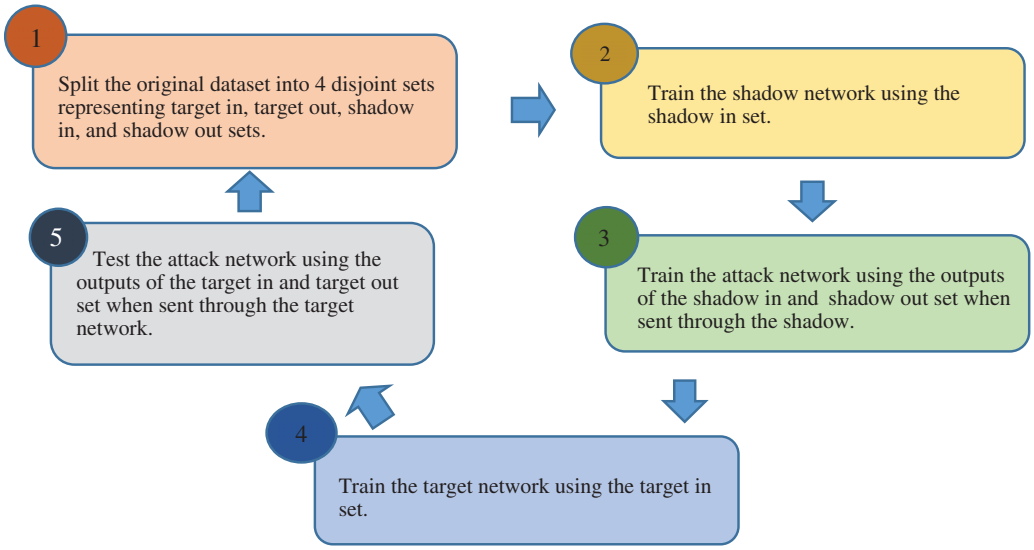


Figure 1: The lifecycle of membership inference attack

The key concept about MIA attack is to use several ML models where each model is used for a prediction class. These attacks that are called *attack models* facilitate inferring membership over the output of the model. In their proposal, Shokri et al. [8] used a black-box model where different shadow models are constructed to imitate the target model behaviour and to enable extracting the required features.

In their paper, Shokri et al. [8] first developed a shadow training technique to create the attack models. Second, the authors construct several "shadow models" that mimic the target model's actions, where the training datasets are known which means that the membership is also known. Afterwards, the attack is trained on the shadow models inputs and outputs. Shokri et al. [8] utilises three different methods to generate training data for the shadow models. These methods are defined as follows:

- Model-based synthesis: this method relies on using black-box to access target model.
- Statistic-based synthesis: the adversary can know some statistical information about the training data used in the target model.
- Noisy real data: the adversary can access some noisy data that are similar to the training data used in the target model.

Shokri et al. [8] presented the issue of deducing correlation between the model output and the training data set as a binary classification. However, Salem et al. [13] relied on three different types of attacks based on the shadow models design and the used training datasets. These attacks are defined as follows:

- The first attack relies on using datasets coming from a similar distribution to the training data used in the target model. This attack also relies on using only one shadow model to reduce the MIA execution cost.
- The second attack relies on using data different from the training data used in the target model. In this attack, the structure of the target model is not known to the attacker. The use of the shadow model facilitates capturing the membership of data samples in the training dataset without imitating the target model.

- The third attack does not rely on using shadow models. Instead it exploits the target model outcomes when querying it with target data points.

Salem et al. [13] applied statistical methods, such as maximum and entropy, on the target model's outputs to differentiate member and non-member data points.

More recently, Nasr et al. [14] proposed membership inference attacks against white-box ML models. For a data sample, they calculate the corresponding gradients over the white-box target classifier's parameters and use these gradients as the data sample's feature for membership inference. While most of the previous works concentrated on classification models, Hayes et al. [15] studied membership inference against generative models, in particular generative adversarial networks (GANs). They designed attacks for both white and black-box settings. Their results showed that generative models are also vulnerable to membership inference. Tab. 3 details different MIA attack models.

Table 3: Summary of membership inference attacks

Shokri et al. [8]	Salem et al. [13]	Nasr et al. [14]	Hayes et al. [15]
• Uses various shadow models. • Attacks a black box target classifier.	• Uses one shadow model: attack 1 et 2, • No shadow model used in attack 3. • Attacks a black box target classifier	• Uses one shadow model. • Attacks a white box target classifier.	• Uses one shadow model • Designed attacks for both white and black box target models. • Membership inference against GANs.

5 Countermeasures Against Attacks on ML

Although there are a variety of security threats to ML models, one can note a lack of research works that shed light on the issues of security for ML models. Basically, most of the existing robustness indicators are a quantitative evaluation of the ML algorithms' performance rather than an evaluation of the security level. Indeed, security is important in ML systems because they often include confidential information, i.e., the data that will be used and/or the ML model itself. In this section, we discuss research works that focus on ML security and privacy attacks countermeasures.

According to the survey introduced by Xue et al. [16] we can classify the countermeasures into two classes: those who secure the model in the training phase such as principal component analysis PCA-based or Data sanitization and the ones that mitigate the vulnerability of ML models at the testing or the inferring phase. Homomorphic encryption and differential privacy are two effective solutions to upgrade the data security and privacy of the data used in machine learning models.

Different defence techniques can be established against machine learning attacks. Indeed, Liu et al. [3] group defence techniques against security and privacy issues in machine learning into four categories: security assessment mechanisms, countermeasures in the training phase, countermeasures in the testing or inferring phase, and data security and privacy. However, Qiu

et al. [17] identified various adversarial defence methods which can be divided into three main groups: modifying data, modifying models and using auxiliary tools.

Salem et al. [13] propose two defence mechanisms to prevent overfitting which is, according to the authors, the main cause of membership inference attacks. These mechanisms are: dropout [18] and model stacking. An overfitting model is a model that cannot be generalized from the training data to unseen data. This is due to learning the noise instead of the signal, it is considered "overfit" because it fits the training dataset but has poor fit with new datasets. A general defence strategy, approved by Yeom et al. [19], is to prevent overfitting using regularization which is a technique that forces the model to be simple. Lomnitz et al. [20] recommend the use of L1 and L2 regularization for the adversarial regularization. Normalization and Dropout can be used as countermeasure, according to Hayes et al. [15]. Likewise, two sets of defence strategies are proposed by Nasr et al. [14]. The first includes simple mitigation techniques, such as restricting the predictions of the model to top-k classes, therefore reducing the precision of predictions, or regularizing the model (e.g., using L2-norm regularizers).

The Differential Privacy (DP) mechanisms are used for the second major set of protection against ML security and privacy issues. These two approaches deal with protecting machine learning models against black-box membership inference attacks. The authors present their contribution which is the min-max privacy game.

In the next section, we outline techniques that avoid attacks against ML models in the training and in the testing/inferring phase. Defence techniques in the testing/inferring phase mainly focus on the improvement of learning algorithms' robustness. However, those that deal with the attacks of the training phase, are concerned with eliminating the poisoning data.

5.1 Defence Techniques Against Attacks in the Training Phase

Lomnitz et al. [20] found that at training level, maintaining the reliability of training data and improving the robustness of learning algorithms are two key countermeasures towards such adversaries. Huang et al. [21] propose a Principal Component Analysis (PCA) based detection against poisoning attacks to improve the robustness of learning algorithms. This defence technique is called Antidote, it is based on statistics to minimise the impact of outliers and can illuminate poisoned data. However, Yeom et al. [9] use bagging classifiers to minimize the impact of the added outliers with the poisoning attack which is an ensemble method. Ensemble method is a paradigm of machine learning in which we train and combine several models in order to produce better results to solve the same problem. The key hypothesis is that we can obtain more accurate and/or robust models when weak models are correctly combined.

Chen et al. [4] present Kuafudet technique to secure malware detection systems against poisoning attacks. This security technique incorporates a system for self-adaptive learning and uses a detector for suspicious false negatives. Another defence technique is purifying the data. Nelson et al. [22] and Laishram et al. [23] use data sanitization to ensure that training data is filtered by extracting the inserted data, by the poisoning attack, from the original ones and then deleting these malicious samples.

All the defence techniques described above are against poisoning attacks. However, there are other attacks in the training phase such as evasion attack. Ambra et al. [24] propose a secure SVM called Sec-SVM to provide an efficient protection against evasion attacks with feature manipulation by enhancing the linear classifiers' protection relying on learning uniformly distributed feature weights.

5.2 Defence Techniques Against Attacks in the Testing/Inferring Phase

Xue et al. [16] suggest invariant SVM algorithms that uses the min-max approach to deal with the testing phase with the feature manipulation operations (i.e., addition, deletion and modification). To make the learning algorithms more robust, Brückner et al. [25] uses Stackelberg Games for adversarial prediction problems and a Nash SVM algorithm based on the Nash equilibrium. Rota Bulò et al. [26] propose a randomised prediction game base on probability distribution specified over the respective strategy set by considering randomized strategy selections. Besides, Zhao et al. [27] propose to incorporate full label adversarial samples into training data in order to provide more robust model training.

Cryptographic techniques can also be used to secure ML models [28,29]. Chen et al. [30] assess the effectiveness of using Differential Privacy as a genomic data protection mechanism to minimize the danger of membership inference attacks.

Table 4: Defence techniques against attacks in the training and testing/inferring phase

Phase of the attack	Attack	Defence technique	Paper
Training	Poisoning attacks	**PCA-based** detection	Benjamin et al. [31]
		Use of **bagging classifiers**	Biggio et al. [32]
		Kuafudet technique	Chen et al. [4]
		Data sanitization	Nelson et al. [22] and Laishram et al. [23]
	Evasion attacks	**Secure SVM** called Sec-SVM	Ambra et al. [24]
Testing/inferring	Adversarial attacks	**Invariant SVM algorithms** using the min-max method	Globerson et al. [33] and Teo et al. [34]
		Stackelberg Games for adversarial prediction problems	Brückner et al. [25]
		A **Nash SVM algorithm** based on the Nash equilibrium	Facchinei et al. [35]
		Randomized prediction game	Rota Bulò et al. [26]
		Introduce adversarial samples with full labels into training data to train a more robust model.	Zhao et al. [27]
	Attacks against data privacy and security	**Min-max privacy** game	Nasr et al. [14]
		Differential privacy (DP)	Chen et al. [30]
		Homomorphic encryption is a technique to provide data privacy via data encryption	Biggio et al. [32]

We presented the main existing countermeasures against machine learning attacks, as shown in Tab. 4. Defence techniques can be summarized as follows: in the training phase, the

countermeasures are working against poisoning attacks that aims to purify the data, it is often called data sanitization during which the anomalous poisoned data is filtered out first before feeding into the training phase. Within the test phase, the defence techniques against sensitive information leakage consist of the adversarial training and ensemble method. To avoid data security and privacy issues, differential privacy and homomorphic encryption are two cryptographic techniques used to address data security and privacy issues.

6 Contribution and Experimental Evaluation

In this section, we detail our implementation of MIA then we propose our defence technique against this attack before evaluating our results. Our proposed solution focuses on securing CNN models against MIA attacks. The purpose of this research is to empirically show the robustness of our privacy-preserving model against MIA attacks.

6.1 Description

Our defence technique is based on the fact that MIA attack exploits the data leakage of the ML model due to overfitting [8]. To this end, we present our solution to mitigate overfitting that consists on using a combination of two techniques which are: dropout and L2 regularization.

Dropout is an efficient method to decrease overfitting based on empirical evidences [18]. The key idea is to randomly drop units from the neural network during training. This prevents units from co-adapting too much. In fact, it is executed by randomly deleting in each training iteration a fixed proportion (dropout ratio) of edges in a fully connected neural network model. We can apply dropout for both the input layer and the hidden layer of the target model. Dropout is specific to neural network.

L2 regularization penalizes the loss function to discourage the complexity of the target model. λ is the penalty term or regularization parameter which determines how much to penalizes the weights. L2 regularization forces the weights to be small but does not make them zero and does non sparse solution. In L2 regularization, regularization term is the sum of square of all feature weights (θ_i^2) as shown in the equation below:

$$\mathrm{L(x, y)} = \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^{n} \theta_i^2$$

To find the best dropout ratio, we measure the impact of varying the dropout ratio of our defence. We test different dropout ratios for both input and fully connected layers when tracking the results of the performance of the MIA attack and the accuracy of the target model. We note that raising the dropout ratio leads to, the lower attack performance. On the other hand, we have obtained very low accuracy of the target model. This means that the accuracy of the target model is stronger when the dropout ratio is mediated. We decide then to use 0.5 and 0.4 as dropout ratios to our defence strategy to maximize our target model accuracy.

As we decide to use regularization technique to overcome overfitting of our model. We test L2 regularization with various values for the regularization factor λ to discourage the complexity of the target model. It achieves this by penalizing the loss function. Furthermore, λ is the penalty term or regularization parameter which determines how much to penalize the weights. L2 regularization forces the weights to be small but does not make them zero and does non sparse solution.

In L2 regularization, regularization term is the sum of square of all feature weights as shown in the equation below:

$$L(x, y) = \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^{n} \theta_i^2$$

where $\sum_{i=1}^{n} (y_i - h_\theta(x_i))^2$ is the loss function

and $\lambda \sum_{i=1}^{n} \theta_i^2$ is the regularization term

We need to find an optimal value of λ leading to a smaller generalization error. To find the optimal value of λ we test our training model with different values (0.05, 0.02, and 0.01). We obtained best result with $\lambda = 0.01$. That's why in all the next experimentation, we have kept the penalty term fixed at 0.01.

In this experimentation, first we investigate the vulnerability of our model against MIA on two trained models on CIFAR-10 and CIFAR-100 datasets, and evaluate the effectiveness of combining Dropout and L2-regularization as a new defence mechanism.

We train a simple image classification model on the CIFAR-10 and CIFAR-100 datasets [36], and then we use the "membership inference attack" against these models to assess if the attacker is able to "guess" whether a particular sample belongs to the training set. Next, we train our model using dropout and L2 regularization to mitigate the leakage of sensitive data of the model. Then, we retest the MIA attack against the model to verify if the attack was mitigated.

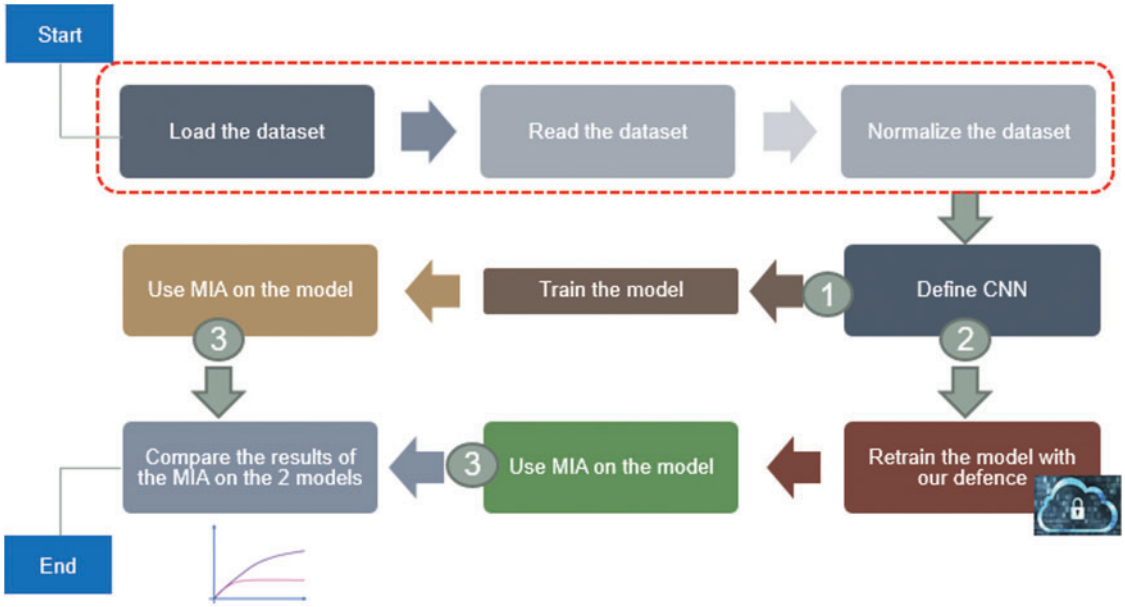


Figure 2: Steps of our experimentation

Fig. 2 presents the steps of our experimentation, first we start with loading the datasets (CIFAR-10 and CIFAR-100), then, once we have read and normalized the data, we define our

model. We use a Convolution Neural Network (CNN) with 3 convolution layers, and we use the Rectified Linear Unit (ReLU) [35] as an activation function because it is the most widely used activation function in neural networks and presented the advantage that it does not activate all neurons at the same time. It only activates a node if the input is above a certain level, while the input is below zero, the output is zero, but when the input rises above a certain threshold, it has a linear relationship with the dependent variable. Next, we train the models and calculate its accuracy using the two datasets (CIFAR-10 and CIFAR-100) to evaluate the performance. Once we trained the model, we tested the MIA attack before defining the defence strategy to verify the privacy vulnerabilities of our model. After defining our strategy of defence based on dropout and regularization, we re-test the MIA attack to show if the defence strategy has mitigated the attack.

6.2 Datasets

We began our experimentation by training our network to classify images from CIFAR-10 and CIFAR-100 datasets [37] using CNN built in TensorFlow environment [38]. Tensorflow is a framework developed by Google, it is an open source library used to facilitate the process of acquiring data, training models, serving predictions, and refining future results.

CIFAR-10 is a standard machine learning dataset consists of 60000 32×32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

CIFAR-100 dataset is like CIFAR-10. However, it has 100 classes containing 600 images each, the global number of data is then 60000. There are 50000 training images and 10000 testing images. i.e., 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. Each image is marked with two labels: first label indicates the class to which it belongs and the second specifies the superclass to which it belongs.

6.3 The CNN Model

We use a CNN with three convolution layers followed by two densely connected layers and an output layer dense layer of size respectively 10 and 100 for CIFAR-10 and CIFAR-100 datasets. Then, we use ReLU as the activation function for hidden layers and sigmoid for the output layer. As, we use the standard categorical cross-entropy loss. Fig. 3 shows the CNN architecture for CIFAR-10.

6.4 Model Training

First, we define the CNN model for the CIFAR-10 and CIFAR-100 datasets, then we train it. To evaluate our model, we used accuracy metrics. Figs. 4 and 5 show respectively the accuracy curve of the model with different value of epochs (38 and 100) for CIFAR-10 dataset and (50 and 100) for CIFAR-100, respectively. We notice that when we increase the number of epochs during the training of the model, it overfits.

6.5 Membership Attack Testing

We use an open-source library of MIA to conduct MIA attack on our trained models [38]. We build one shadow model on the shadow dataset to imitate the target model, and we generate the

base to train the attack model. The attack dataset is constructed by concatenating the probability vector output from the shadow model and true labels. If a sample is used to train the shadow model, the corresponding concatenated input for the attack dataset is labelled 'in', and 'out' otherwise.

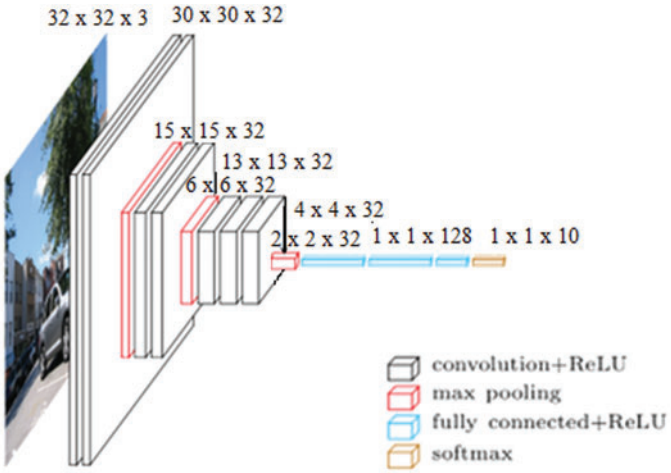
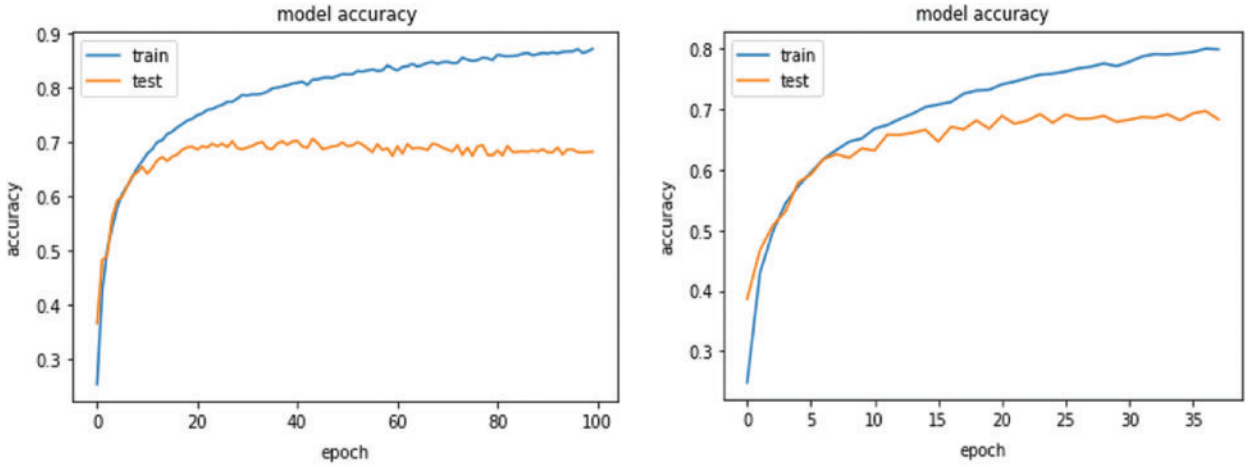


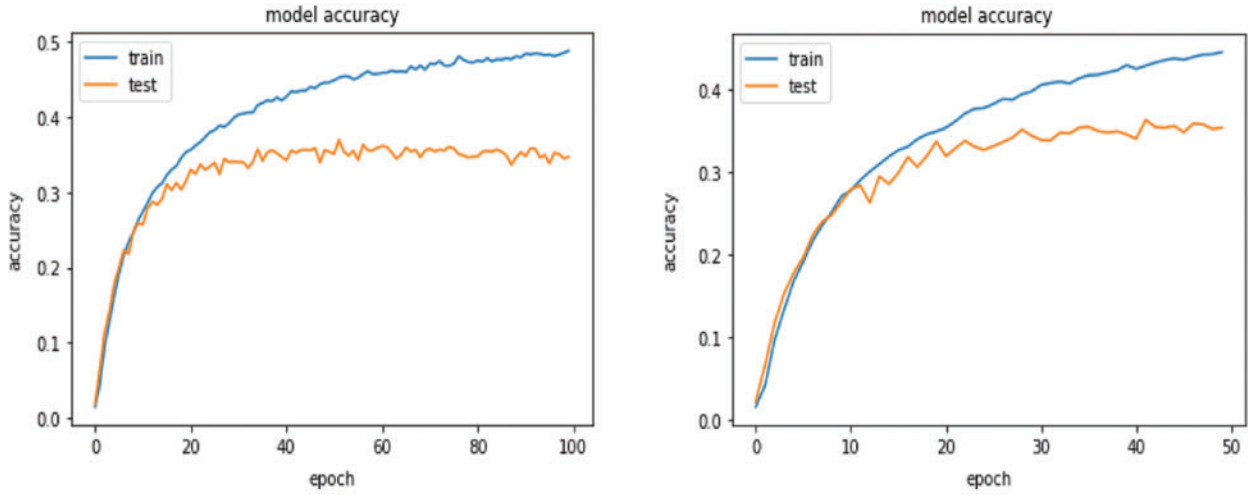
Figure 3: CNN architecture for CIFAR-10 dataset



CIFAR-10, epoch = 100 CIFAR-10, epoch = 38

Figure 4: Accuracy curve of the model

Afterwards, we execute a MIA against the previously models trained on the two chosen training datasets CIFAR-10 and CIFAR-100. As it was defined in Section 4, MIA consists of quantifying how much information a machine learning model leaks about its training data, which could be personal and sensitive. The main idea of MIA consists on the examination of the predictions made the model to guess if a particular data sample was used in the training dataset or not.



CIFAR-100, epoch = 100 CIFAR-100, epoch = 50

Figure 5: Accuracy curve of the model

Shokri et al. [8] first developed the idea of using shadow models, where multiple shadow models with varying in/out splits were used to train a single attack model. We only used a single shadow model in the same way as used the paper presented by Salem et al. [13].

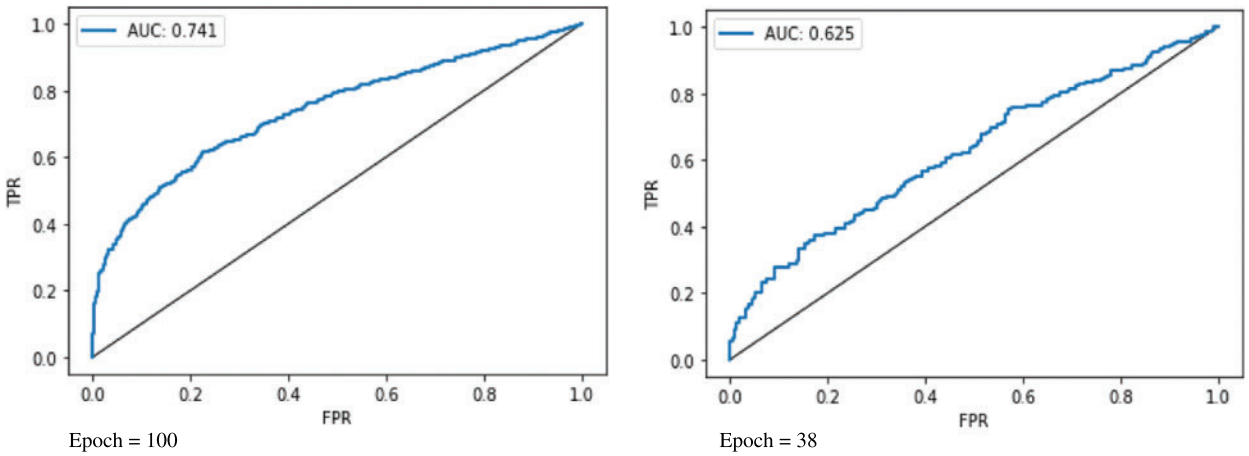
To evaluate our MIA attack, we choose to use Area Under the Curve (AUC) metric. This is one of the popular metrics which measures the ability of a classifier to differentiate between classes. It is used as a summary of the ROC (Receiver operating characteristic) curve.

The ROC curve is the plot between sensitivity and (1-specificity). Sensitivity is also known as True Positive rate and (1-specificity) is also known as false positive rate. The biggest advantage of using ROC curve is that it is independent of the change in proportion of responders.

An AUC of close to 0.5 means that the attack wasn't able to identify training samples, which means that the model doesn't have privacy issues according to this test. However, higher values indicate potential privacy vulnerabilities.

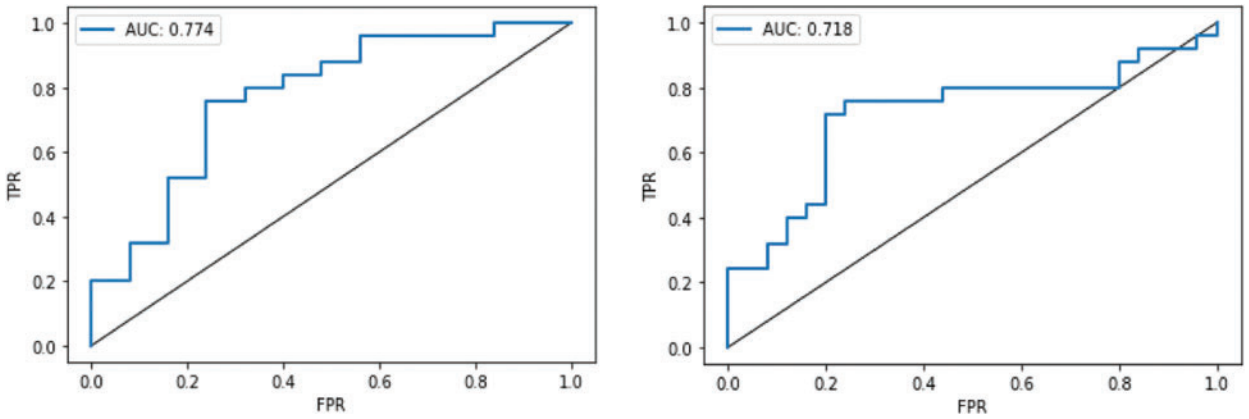
Fig. 6 exposes the AUC of the execution of MIA on our model trained on CIFAR-10. We noticed that the first curve with epoch = 100 presents higher values (i.e., AUC = 0.741) which indicates much privacy vulnerabilities than the second one for epoch = 38 leading to an AUC equal to 0.625. Indeed, the closer this curve is to the upper left corner, the more efficiently the classifier behaves. This can be explained by the early stopped of the model trained (i.e., epoch = 38 *vs.* epoch = 100).

Fig. 7 exposes the evaluation of the MIA attack on the model trained on CIFAR-100. The two curves present a very near values (0.774 and 0.718), this can be explained by the fact that at the training phase of the model there was no degradation of the results when using the test data (the curve was almost constant).



Epoch = 100 Epoch = 38

Figure 6: AUC of MIA attack on model trained on CIFAR-10



100 epochs 50 epochs

Figure 7: AUC of MIA attack on model training on CIFAR-100

6.6 Evaluating the Solution Performance Against Membership Inference Attacks

The purpose of this section is to empirically show the robustness of our privacy-preserving model against Membership Inference Attacks. As we mentioned in previous sections, the main cause of the success of MIA attack is overfitting, therefore we propose to use techniques that mitigate it. Overfitting occurs when the model is overtrained on the training component of the dataset, such that when the model encounters different data, it gives worse results than expected.

There are various ways to prevent overfitting. We are focused on two techniques: dropout and L2-regularization. In addition, we have discussed as shown in Fig. 8, the early stopping which is a technique consisting in the interruption of the training when the performance on the validation set starts dropping.

On the other hand, regularization is a technique intended to discourage the complexity of a model by penalizing the loss function. It assumes that simpler models are better for generalization,

and thus better on unseen test data. L2 regularization is known as Least Square Error. It minimizes the square of the sum of the difference between target values and estimates values.

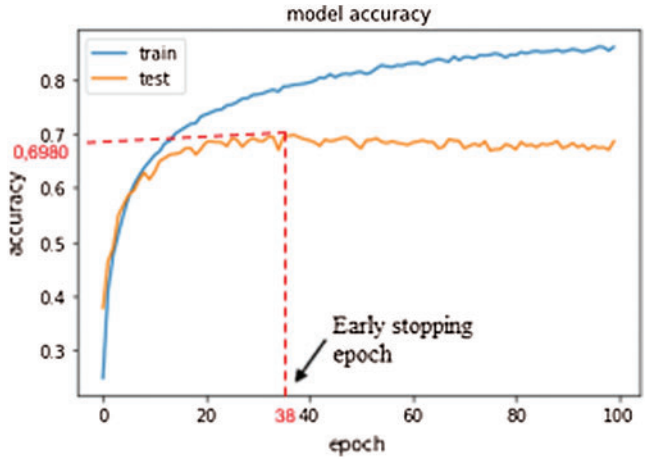


Figure 8: The early stopping

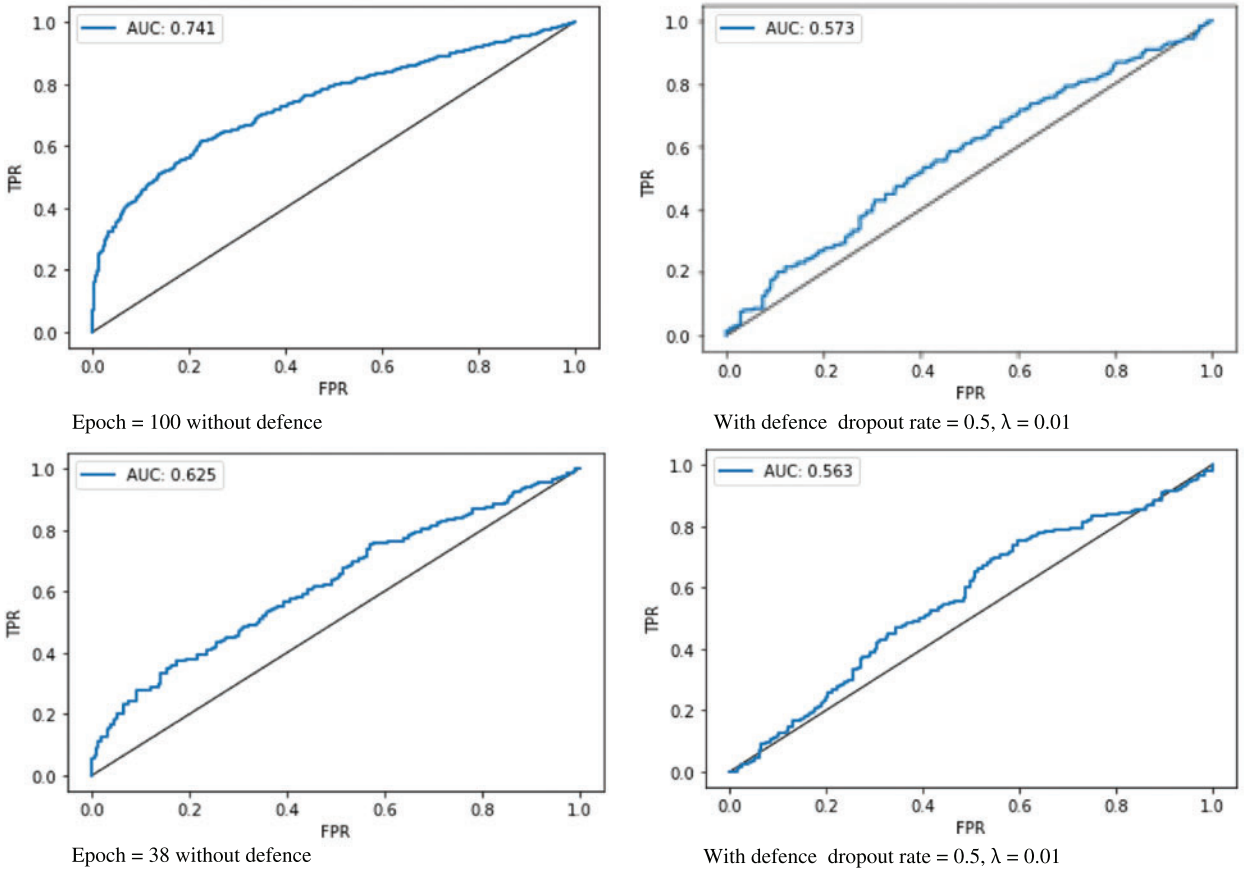
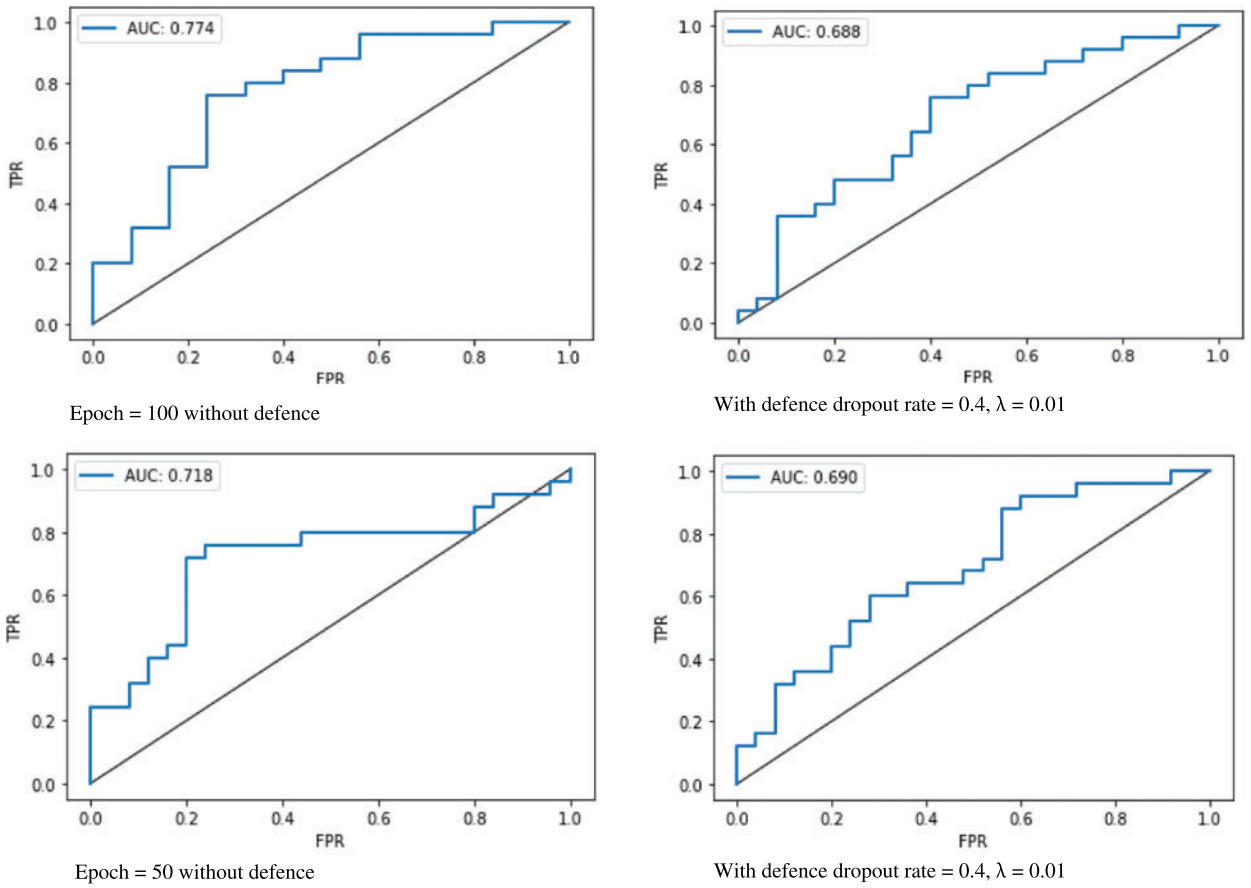


Figure 9: Degradation of MIA on CIFAR-10

The main idea of dropout is to randomly drop units from the neural network during training, which prevents units from co-adapting too much. Dropout is introduced by Hinton et al. [39] to prevent co-adaption among the training data. In our experimentation we use respectively 0.5 and 0.4 as dropout rate for the CIFAR-10 and CIFAR-100 datasets.

Experimental results show that the attack performance using L2-regularization and dropout, in the training phase, is lower than the same attack without introducing neither dropout nor L2-regulrization. Figs. 9 and 10 show the degradation of the performance of MIA after applying L2 regularization and dropout (from AUC = 0.741 to AUC = 0.573 with CIFAR-10 also an improvement of 22.6% for epoch = 100% and 9.92% for epoch = 38) but the accuracy of the target model decreased (i.e., the model accuracy of test dataset is decreased of 22.16% from 0.6643 to 0.4427 for CIFA10 with epoch = 100 as shown in Fig. 10.



Epoch = 100 without defence

With defence dropout rate = 0.4, λ = 0.01

Epoch = 50 without defence

With defence dropout rate = 0.4, λ = 0.01

Figure 10: Degradation of MIA on CIFAR-100

6.7 Discussion

After loading datasets on which we trained the target model, we defined the CNN model that we will try to optimize its accuracy. We measure the CNN model vulnerability against MIA. We tested our attack on two datasets CIFAR-10 and CIFAR-100, and we compared the results of our attack on the two trained models, we notice that the attack is more efficient on the second dataset (with 10 times more classes), that matches the results announced by Shokri et al. [8]. This shows that models with more efficiency classes should be able to remember more about their training datasets, and therefore they can leak more data about them. As we found that by reducing the number of epochs when training the models on the same dataset, the performance of the attack was reduced. This is can be explained by the fact that when we early stopped the training of the model, we reduce its overfitting.

We investigate the dropout and L2-regularization to mitigate overfitting of the target model in order to avoid the privacy leakage. We verify that the modified model is more resistant to the MIA attack. Indeed, our results show an improvement in preserving membership privacy of 22.67% for the CIFAR-10 with epoch = 100% and 9.92% for the same training dataset with epoch = 38. However, there is a degradation of the accuracy of the target model (from 68.92% to 39.67%, i.e., a degradation of 29.25% with epoch = 38 after adding dropout and L2 regularization as a defence technique.

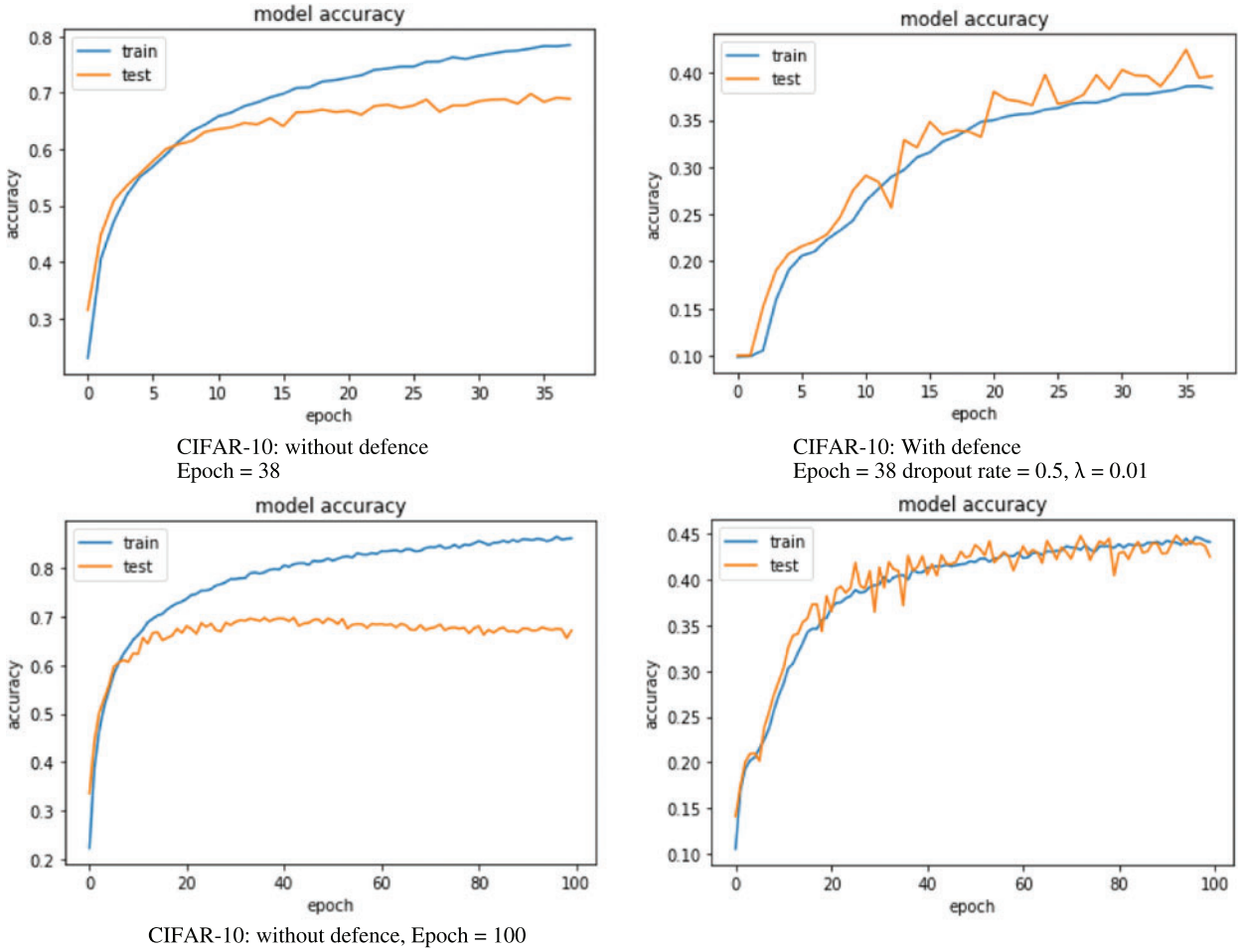
Tab. 5 shows that bigger accuracy gaps between the training and testing datasets are associated with higher precision of membership inference.

Table 5: The accuracy of the target model and the performance of the attack

Training dataset		Training accuracy	Testing accuracy	Attack AUC
CIFAR-10 epoch = 38	Without defence	0.7896	0.6892	0.625
	With defence	0.3800	0.3967	0.563
Degradation of MIA				9.92%
CIFAR-10 Epoch = 100	Without defence	0.8713	0.6643	0.741
	With defence	0.4261	0.4427	0.573
Degradation of MIA				22.6%

After applying our defence strategy, we notice that the performance of the attack was mitigated for the two models trained on CIFAR-10 and CIFAR-100. We achieve a degradation of 22.6% and 11.56% of the attack on respectively CIFAR-10 and CIFAR-100 with epoch = 100. However, we observe a degradation of the accuracy of the trained models which fell by 22.16% and 16.49% for the two models trained on respectively CIFAR-10 and CIFAR-100 with epoch equal to 38 and 50.

Experimental results show that the accuracy of the target model is decreased when we try to preserve privacy of the data. That is why we have to find a trade-off between preserving privacy of the model and its performance (as it is shown in Fig. 11).



CIFAR-10: without defence
Epoch = 38

CIFAR-10: With defence
Epoch = 38 dropout rate = 0.5, λ = 0.01

CIFAR-10: without defence, Epoch = 100

Figure 11: Degradation of the accuracy of the target model

7 Related Works

Membership inference attacks seek to infer whether a particular data record was used in the model training dataset or not. An adversary can have black-box access to a machine learning-as-a-service API [13]. Various countermeasures are defined in different research works to mitigate the leakage of information and enforce the privacy of the target model.

Differential Privacy (DP) is a privacy preserving technique that can be implemented in training algorithms in a multitude of fields. It was developed in data processing with relation to privacy concerns. DP is often obtained by applying a procedure that introduces randomness into the data. DP has been the most widely used method, according to Chen et al. [30], to assess privacy exposure relating to persons. In addition, Chen et al. [30] has evaluated DP uses and its efficiency as solution to MIA in genomic data. The authors presented a trade-off between securing the model against MIA and the accuracy the of target model using various settings of DP. Moreover, DP was applied by Dwork [29], to explain group statistics while preserving participants records within the training datasets. DP enables achieving a similar outcome of two different datasets processing where only one record is different between the two datasets.

The min-max privacy game proposed by Nasr et al. [14] introduces a specific setting in which the adversary wants to achieve the maximum inference advantage and the defender has to find the classification model that not only minimises his loss, but also minimises the maximum gain of the adversary. This is a Stackelberg min-max game [40]. To mitigate the information leakage of machine learning, Nasr et al. [14] have offered a new privacy mechanism against membership inference in training datasets, through their predictions. The authors have proposed a trade-off to both increase privacy and accuracy. The solution consists in a model where its predictions on its training data cannot be distinguished from its predictions on any other data sample from the same distribution.

Salem et al. [13] present another defence technique, namely model stacking, which works independently of the used ML classifier. This solution consists of training the model using different subsets of data which makes the model less prone to overfitting.

We can classify the existing defence techniques into two major groups. The first group consists on including simple mitigation techniques, consisting of reducing the accuracy of predictions, or regularizing the model (e.g., using L2 regularization). These techniques may incur a negligible utility loss to the model. The second group is composed by differential privacy techniques.

8 Conclusion

In this paper, we have presented our implementation of MIA on CNN model, then we have introduced our defence technique to evaluate its effectiveness in increasing the security of the model against these attacks. We evaluated the effectiveness of using L2 regularization and dropout as a defence technique to mitigate the overfitting of the model which is considered as the main cause of the information leakage related to the training dataset [8]. Indeed, we reached a decrease of the AUC of the attack from 0.625 to 0.563 (with epoch = 38 for CIFAR-10) and from 0.741 to 0.573 (with epoch = 100 for CIFAR-10). Our evaluation showed that our defence technique is able to reduce the privacy leakage and mitigate the impact of the membership inference attacks. However, experimental results showed that the accuracy of the target model was decreased when we tried to preserve privacy of the data. That is why we have presented a trade-off between preserving privacy of the model and its performance. The problem will then be transformed to find the optimal solution to maintain the performance of the target model while raising his membership privacy. As future work, we aim to enhance the proposed solution to achieve better accuracy of the model while preserving membership privacy.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Mrabet, S. Belguith, A. Alhomoud and A. Jemai, "A survey of IoT security based on a layered architecture of sensing and data analysis," *Sensors*, vol. 20, no. 13, pp. 3625, 2020.

[2] S. B. ElMamy, H. Mrabet, H. Gharbi, A. Jemai and D. Trentesaux, "A survey on the usage of blockchain technology for cyber-threats in the context of industry 4.0," *Sustainability*, vol. 12, no. 21, pp. 9179, 2020.

[3] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu *et al.*, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE Access*, vol. 6, pp. 12103–12117, 2018.

[4] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu *et al.*, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach," *Computers and Security*, vol. 73, pp. 326–344, 2018.

[5] A. Polyakov, "How to attack machine learning (evasion, poisoning, inference, trojans, backdoors)," Aug. 06, 2019. [Online]. Available: https://towardsdatascience.com/how-to-attack-machine-learning-evasion-poisoning-inference-trojans-backdoors-a7cb5832595c. Accessed: April 22, 2021.

[6] L. Seo Jin, P. D. Yoo, A. T. Asyhari, Y. Jhi, L. Chermak, C. Y. Yeun *et al.*, "IMPACT: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction," *IEEE Access*, vol. 8, pp. 65520–65529, 2020.

[7] S. Chen, R. Jia and G.-J. Qi, "Improved techniques for model inversion attacks," arXiv preprint, arXiv: 2010.04092 [cs], Oct. 2020, Accessed: Mar. 25, 2021.

[8] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symp. on Security and Privacy*, San Jose, California, USA, pp. 3–18, 2017.

[9] S. Yeom, I. Giacomelli, A. Menaged, M. Fredrikson and S. Jhab, "Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine Learning," *Journal of Computer Security*, vol. 28, no. 1, pp. 35–70, 2020.

[10] B. Biggio, G. Fumera and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 4, pp. 984–996, 2013.

[11] N. Papernot, P. Mcdaniel, A. Sinha and M. Wallman, "Sok: Towards the science of security and privacy in machine learning," in *Proc. of 3rd IEEE European Symp. on Security and Privacy*, London, UK, pp. 1–19, 2018.

[12] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis and G. Loukas, "A taxonomy and survey of attacks against machine learning," *Computer Science Review*, vol. 34, pp. 100199, 2019.

[13] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz *et al.*, "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Network and Distributed Systems Security Symp.*, San Diego, California, USA, 2019.

[14] M. Nasr, R. Shokri and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security*, Toronto, Canada pp. 634–646, 2018.

[15] J. Hayes, L. Melis, G. Danezis and E. De Cristofaro, "LOGAN: Membership inference attacks against generative models," in *Proc. on Privacy Enhancing Technologies*, Bacelona, Spain, pp. 133–152, 2018.

[16] M. Xue, C. Yuan, H. Wu, Y. Zhang and W. Liu, "Machine learning security: Threats, countermeasures, and evaluations," *IEEE Access*, vol. 8, pp. 74720–74742, 2020.

[17] S. Qiu, Q. Liu, S. Zhou and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," *Applied Sciences*, vol. 9, no. 5, 2019.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[19] S. Yeom, I. Giacomelli, M. Fredrikson and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting,"in *IEEE 31st Computer Security Foundations Symp.*, Oxford, UK, pp. 268–282, 2018.

[20] M. Lomnitz, N. Lopatina, P. Gamble, Z. Hampel-Arias, L. Tindall *et al.*, "Reducing audio membership inference attack accuracy to chance: 4 defenses', arXiv preprint, arXiv: 1911.01888 [cs, Eess], Oct. 2019.

[21] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein and J. D. Tygar, "Adversarial machine learning," in *Proc. of the 4th ACM Workshop on Security and Artificial Intelligence*, New York, NY, USA, pp. 43–58, 2011.

[22] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein *et al.*, "Misleading learners: Co-opting your spam filter," in *Machine Learning in Cyber Trust: Security, Privacy, and Reliability*, Boston, MA: Springer US, pp. 17–51, 2009.

[23] R. Laishram and V. V. Phoha, "Curie: A method for protecting SVM classifier from poisoning attack," arXiv preprint, arXiv arXiv: 1606.01584 [cs], Jun. 2016, Accessed: April, 2021.

[24] D. Ambra, M. Melis, B. Biggio, D. Maiorca, D. Arp *et al.*, "Yes, machine learning can be more secure! a case study on android malware detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 711–724, 2017.

[25] M. Brückner and T. Scheffer, "Nash equilibria of static prediction games," *Advances in Neural Information Processing Systems*, vol. 22, pp. 171–179, 2009.

[26] S. Rota Bulò, B. Biggio, I. Pillai, M. Pelillo and F. Roli, "Randomized prediction games for adversarial machine Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 1–13, 2016.

[27] Z. Zhao, D. Dua and S. Singh, "Generating natural adversarial examples," in *Int. Conf. on Learning Representations*, Vancouver, Canada, 2018.

[28] G. Alagic, G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper *et al.*, "Status report on the first round of the NIST post-quantum cryptography standardization process," *National Institute of Standards and Technology*, Washington, DC: US Department of Commerce, 2019.

[29] C. Dwork, "Differential privacy," in *33rd Int. Colloquium on Automata, Languages and Programming, Part II*, Venice, Italy, vol. 4052, pp. 1–12, 2006.

[30] J. Chen, W. H. Wang and X. Shi, "Differential privacy protection against membership inference attack on machine learning for genomic data," bioRxiv, World Scientific, 2020.

[31] R. Benjamin IP, B. Nelson, L. Huang, A. D. Joseph, S. Lau *et al.*, "ANTIDOTE: Understanding and defending against poisoning of anomaly detectors," in *Proc. of the 9th ACM SIGCOMM Conf. on Internet Measurement*, New York, NY, USA, pp. 1–14, 2009.

[32] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić *et al.*, "Evasion attacks against machine learning at test time," in *Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, Springer, Berlin, Heidelberg, pp. 387–402, 2013.

[33] A. Globerson and S. Roweis, "Nightmare at test time: Robust learning by feature deletion," in *Proc. of the 23rd Int. Conf. on Machine Learning*, Pittsburgh, Pennsylvania, USA, pp. 353–360, 2006.

[34] C. H. Teo, Q. Le, A. J. Smola and S. V. N. Vishwanathan, "A scalable modular convex solver for regularized risk minimization," in *Proc. of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Jose, California, USA, pp. 727–736, 2007.

[35] F. Facchinei and C. Kanzow, "Generalized nash equilibrium problems," *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, 2010.

[36] "CIFAR-10 and CIFAR-100 datasets," [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html, Accessed: Dec. 30, 2020.

[37] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," in *Int. Joint Conf. on Neural Networks*, Anchorage, Alaska, USA, IEEE, pp. 2684–2691, 2017.

[38] "Introducing TensorFlow Privacy: Learning with Differential Privacy for Training Data," [Online]. Available: https://github.com/tensorflow/privacy, Accessed: Dec. 30, 2020.

[39] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv: 1207.0580 [cs], Jul. 2012.

[40] H. von Stackelberg, *Market Structure and Equilibrium*. Berlin, Heidelberg: Springer Science & Business Media, 2010.