

Masterarbeit am Institut für Informatik der Freien Universität Berlin,
Arbeitsgruppe ID Management

Exploration of checkpoints in the context of membership inference attacks

Nest, Marisa

marisa.f.nest@fu-berlin.de

Matrikelnummer: 5396051

Betreuerin: Franziska Boenisch

1. Gutachter: Prof. Marian Margraf

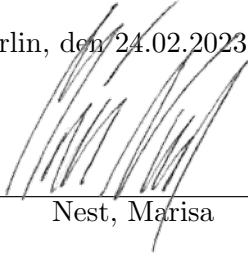
2. Gutachter: Prof. Gerhard Wunder

Berlin, den 24.02.2023

Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 24.02.2023



Nest, Marisa

Abstract

The field of machine learning (ML) has been growing over the last years. An increasing number of systems based on ML models, which are trained on a wide variety of data sets, are publicly accessible. Since more and more models are also based on data that contain private information that also implies that these models and the associated data must be protected in terms of privacy. A first step in protecting a model's privacy is to evaluate its level of protection against attacks. One of such heuristic privacy evaluation methods that has become widespread in recent years are membership inference attacks (MIAs) [2, 15, 18, 19, 21, 23]. In the past, the privacy assessment under MIA did not consider a temporal component of the model, but only considered the final model. This work now aims to test whether the addition of a temporal dimension in the form of so-called checkpoints in the context of MIA can serve to provide a better and more accurate picture of a model's state of privacy. In order to test this, two exploratory experiments are conducted in which multiple time series analysis are performed. In addition, a new MIA using checkpoints is presented. In the end, it can be shown that in certain circumstances, especially when considering incorrectly classified data, checkpoints can help to provide a better evaluation of privacy and that the performance of the newly introduced MIA can compete with the performance of other recent MIA attacks.

Contents

1. Introduction	1
2. Background & related Work	3
2.1. Background	3
2.1.1. Data sets	3
2.1.2. Neural network	4
2.1.3. Membership inference attack	5
2.2. Related Work	6
2.2.1. Membership inference attacks	6
2.2.2. Time series	9
2.2.3. Support vector machines	10
3. Approach	13
3.1. Data sets & target model	13
3.2. Experiment I	14
3.2.1. Methodology	14
3.3. Experiment II	16
3.3.1. Methodology	16
4. Implementation	19
4.1. Software tools & libraries	19
4.2. Data sets & target model	19
4.3. Experiments I	20
4.4. Experiments II	21
5. Results	23
5.1. Data sets & target model	23
5.2. Experiment I	24
5.2.1. CIFAR-10	24
5.2.2. CIFAR-100	28
5.3. Experiment II	31
5.3.1. CIFAR-10	31
5.3.2. CIFAR-100	34
6. Conclusion	37
6.1. Experiment I	37

Contents

6.2. Experiment II	38
7. Discussion and future work	41
A. Appendix	43
A.1. Experiment I	43
A.2. Experiment II	43

1. Introduction

For many years, machine learning (ML) and, in particular, artificial neural networks (ANNs) have been a widely used technology. They are applied in a wide range of areas such as in health care, the legal system, social services, computer vision or language modeling. To utilize these systems effectively, often a lot of data is required. This data might be private and contain sensitive personal information and should therefore be protected in terms of privacy.

In the past, it has been shown that ML models can unintentionally reveal sensitive information about their underlying training data sets. This private data can be exposed, for example, through targeted attacks such as the membership inference attack [19]. Membership inference attacks (MIAs) describe a class of attacks that aim to identify whether or not a particular data point is part of a training data set of a model (also called the target model). Although at first glance this information does not seem to be a serious privacy issue, the following example shows why an ML model should not disclose even such simple information: In a medical study, a distinction is often made between case and control groups, where the first group has a certain health condition that is not present in the second group. Hence, for example, when studying what factors can lead to cancer, the information about whether a person belongs to one group or the other can provide information about that person's health status. The disclosure of such sensitive information is a serious violation of privacy.

In practice, MIAs are used (e.g. by the model owners themselves) to make an empirical statement about whether or not a model poses a risk to the privacy of a training data set. For a realistic assessment, it is therefore important that these attacks are as comprehensive and effective as possible. This work aims to improve the model's privacy risk assessment from the perspective of the model owner in the context of membership inference by adding another layer of information to the evaluation. Currently, only one final version of the target model is used in the context of MIA. In this work, it is now proposed to use not just one version of the target model, but several models, so called checkpoint. Checkpoints represent different temporal states of the model during the training phase. In practice, these checkpoints are often used for different purposes, e.g. to analyze the training progress or to select the best performing model at the end of the training. Since checkpoints are already created during training or can be easily collected and stored (i.e. just by adding storage

1. Introduction

capacity), this is also an easy way to extend the information used by MIA.

In this work, it is hypothesised that the additional information that comes from the multiplicity of checkpoints can be used to better distinguish between member and non-member data points. To test this hypothesis, two experiments are conducted. First, it is investigated whether different behaviour between membership and non-membership data points can be observed over time (i.e. across checkpoints) by performing an exploratory data analysis. In the second experiment, a new membership inference attack based on the checkpoints is conducted to see if the additional data can be used to train an effective adversary.

2. Background & related Work

In the following chapter, the background and the related work of this thesis is introduced.

2.1. Background

2.1.1. Data sets

CIFAR-10 and CIFAR-100 are two data sets which were published by Krizhevsky [9] and are widely used in the field of MIA (c.f. [2, 15, 18, 19, 21]). Both data sets consist of 60,000 images, 50,000 of which are training sample and 10,000 are test sample. The images have a size of 32×32 pixels and are in color (i.e. they have three pixel channels). Each image is labeled with a class to which it belongs. For CIFAR-10 there are a total of 10 classes and for CIFAR-100 there are 100 classes (see table 2.1). The CIFAR-10 data set has a total of 6000 images per class and the CIFAR-100 600.

Data set	Classes
CIFAR-10	airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck
CIFAR-100	apple, aquarium fish, baby, bear, beaver, bed, bee, beetle, bicycle, bottle, bowl, boy, bridge, bus, butterfly, camel, can, castle, caterpillar, cattle, chair, chimpanzee, clock, cloud, cockroach, couch, crab, crocodile, cup, dinosaur, dolphin, elephant, flatfish, forest, fox, girl, hamster, house, kangaroo, keyboard, lamp, lawn mower, leopard, lion, lizard, lobster, man, maple tree, motorcycle, mountain, mouse, mushroom, oak tree, orange, orchid, otter, palm tree, pear, pickup truck, pine tree, plain, plate, poppy, porcupine, possum, rabbit, raccoon, ray, road, rocket, rose, sea, seal, shark, shrew, skunk, skyscraper, snail, snake, spider, squirrel, streetcar, sunflower, sweet pepper, table, tank, telephone, television, tiger, tractor, train, trout, tulip, turtle, wardrobe, whale, willow tree, wolf, woman, worm

Table 2.1.: Classes of CIFAR-10 and CIFAR-100

2. Background & related Work

Since many of the works that deal with membership inference use CIFAR-10 and CIFAR-100 as standard evaluation data sets, both will be used in this work.

Notation A data set is denoted as D and is sampled from a distribution \mathbb{D} : $D \leftarrow \mathbb{D}$. D consists of the data points $X \in \mathbb{R}^{N_D \times N_x}$ and their corresponding one-hot encoded classes $Y \in [0, 1]^{N_D \times N_y}$, where N_D is the number of data points in D , N_x the number of features and N_y the number of possible classes. One-hot encoded means that a label y is described as a vector of N_y features and all features are zero except the one that represents the actual class, also called true class. x_i describes the i -th element of X . y_i is the i -th element of Y and describes the class for x_i . The CIFAR-10 data set is noted as $D_{CIFAR-10}$ and CIFAR-100 as $D_{CIFAR-100}$.

2.1.2. Neural network

Neural networks, also called artificial neural networks (ANNs), are structurally and functionally inspired by the human brain (cf. [7]). ANNs consist of different layers: an input layer, one or more hidden layers, and an output layer. Each layer consists of nodes that connect the layers with each other (cf. figure 2.1). Each connection has learnable parameters. In order to improve the performance of an ANN, these parameters need to be learned by an training algorithm and a data set.

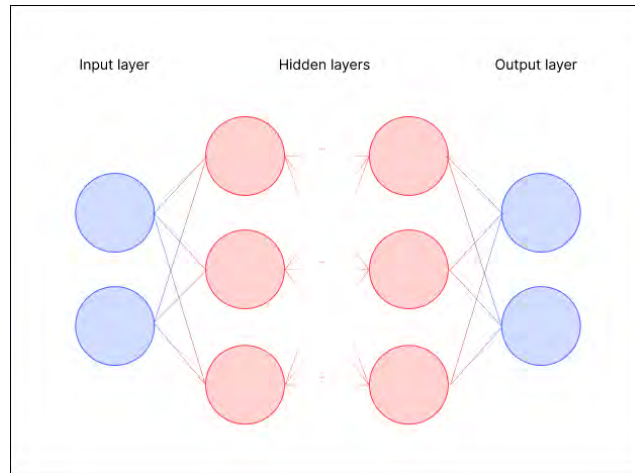


Figure 2.1.: Artificial neural network

Notation A neuronal network can also be described as a parameterized function f_θ where θ are the parameters to be learned. The function f_θ maps an input $x \in X$ to an output $\hat{y} \in [0, 1]^{N_y}$:

$$f_\theta(x) = \hat{y}$$

y and \hat{y} are both one-hot encoded class vectors. In order to get the true class or the predicted class (i.e. the class that is predicted by a network), an argmax function needs to be applied (i.e. $\text{argmax } y$ for the true class and $\text{argmax } \hat{y}$ for the predicted class). Thus a correct classification is noted as $\text{argmax}(\hat{y}) = \text{argmax}(y)$ and an incorrect classification as $\text{argmax}(\hat{y}) \neq \text{argmax}(y)$. The parameters θ of the network f_θ are learned by an training algorithm \mathcal{T} and a training data set D_{train} sampled from \mathbb{D} :

$$f_\theta \leftarrow \mathcal{T}(D_{\text{train}})$$

The network function $f_\theta(x) = \hat{y}$ can also be described as $f_\theta(x) = \sigma(z(x))$. $z(x)$ returns the so called logit outputs of the network where $z : X \rightarrow \mathbb{R}^{N_y}$. $\sigma(z)$ is also referred to as the softmax layer and returns a probability distribution also called confidence values, where $\sigma : X \rightarrow [0, 1]^{N_y}$. The confidence values can also be understood as the probabilities of a sample $x \in X$ belonging to a specific class. This specific class that a confidence value represents can also be called a confidence class. In this work, the function f_θ will be cited as f_{target} .

2.1.3. Membership inference attack

MIA is a privacy attack that attempts to reveal whether or not a particular data point is part of a training data set of a particular model (i.e. the target model). The target model is a neuronal network described as the function f_θ . A membership inference attack can formally be described as a security game. This security game was introduced by Carlini et al. [2] and is defined as follows:

Definition 1 (Membership inference security game). *The game proceeds between a challenger \mathcal{C} and an adversary \mathcal{A} :*

1. *The challenger samples a training data set $D \leftarrow \mathbb{D}$ and trains a model $f_\theta \leftarrow \mathcal{T}(D)$ on the data set D .*
2. *The challenger flips a bit b , and if $b = 0$, samples a fresh challenge point from the distribution $(x, y) \leftarrow \mathbb{D}$ (such that $(x, y) \notin D$). Otherwise, the challenger selects a point from the training set $(x, y) \leftarrow D$.*
3. *The challenger sends (x, y) to the adversary.*
4. *The adversary gets query access to the distribution \mathbb{D} , and to the model f_θ , and outputs a bit $\hat{b} \leftarrow \mathcal{A}^{\mathbb{D}, f_\theta}(x, y)$.*
5. *Output 1 if $\hat{b} = b$, and 0 otherwise*

In this work, $\mathcal{A}^{\mathbb{D}, f}$ is referred to as \mathcal{A}

2.2. Related Work

2.2.1. Membership inference attacks

In the context of ML systems or more precisely of ML classification systems, MIA was first introduced by Shokri et al. [19]. In their work, they show that under certain conditions, it is possible to infer the membership of a data point via a simple black-box access (i.e. where the adversary only has access to the output of the model for a given input) to the target model. The basic assumption Shokri et al. made is that an ML model behaves differently if a particular data point was or was not part of the training set. This different behavior is measured using different information about the target model and the data points. In the case of Shokri et al., like many other publications (e.g. [2, 15, 18, 21]), the confidence values are used. This work will also focus on confidence values when analysing the usability of checkpoints in the context of MIA.

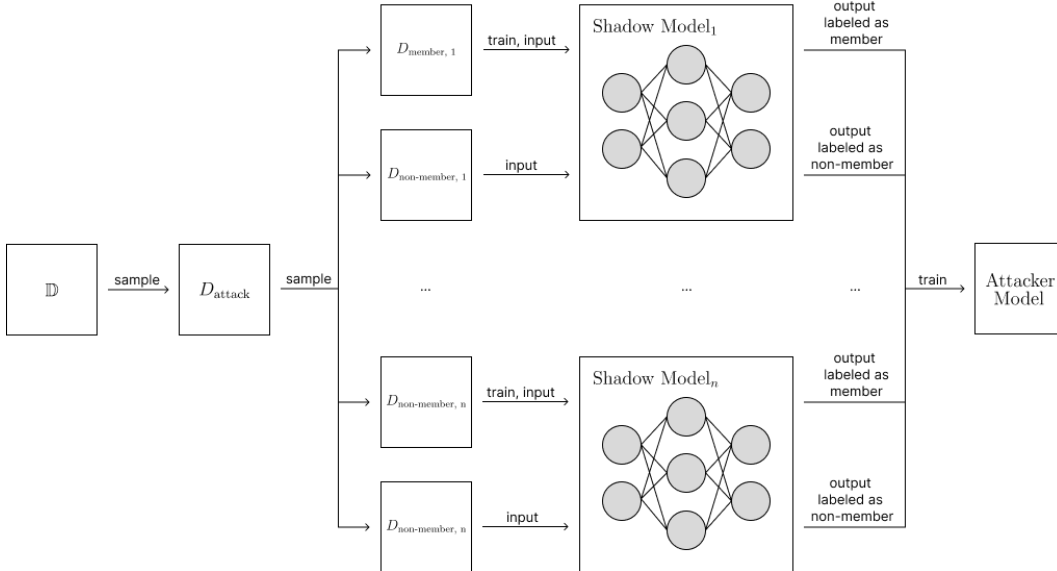


Figure 2.2.: Training of the shadow models and the attacker model

In order to learn how to distinguish between members and non-members, the attack by Shokri et al. needs not only the confidence values, but also the information whether the data point was part of the training data set of the target model or not. Since this information is not available with only a black-box access and, moreover, this is the information that is to be revealed later by the attack, the paper by Shokri et al. introduces so-called shadow models. Shadow models serve collectively as a surrogate model for the target model whose behavior they attempt to imitate. To use shadow models, the attacker \mathcal{A} samples data $D_{\text{attack}} \leftarrow \mathbb{D}$ from the same

data distribution from which the target model’s training data is drawn (i.e. \mathbb{D}). As shown in the visualization 2.2, n shadow models $f_{\text{shadow}, i}(x), i \in \{1, \dots, n\}$ are trained based on this data, with each shadow model receiving a different subset of D_{attack} , i.e. $D_{\text{shadow}, i} \leftarrow D_{\text{attack}}$. A labeled output is then generated for each shadow model and data point $x \leftarrow D_{\text{attack}}$. The output consists of the returned confidence values of the data point x and the shadow model $f_{\text{shadow}, i}$, whereas the label describes whether or not the data point x was used to train the shadow model $f_{\text{shadow}, i}$ (i.e., whether it is a member or non-member). The attacker \mathcal{A} now has access not only to the confidence values of the shadow models, but also to the ground truth about whether or not the data point was part of the training set of a particular shadow model. This information is subsequently used to fit an attacker model $f_{\text{attack}}(x)$ (i.e. a binary ML classifier) to distinguish between member and non-member samples. As described in the figure 2.3, the attacker model can then be used to predict the membership status for an unseen data point $x \leftarrow \mathbb{D}$. This coined methodology by Shokri et al. describes the basic structure of a membership inference attack on which also this work will focus on. Since this work follows an exploratory approach and the experiments are conducted from the perspective of the model owner, the knowledge about the membership status of a data point and the access to any information of the target model are already available. Therefore no shadow models need to be trained in order to obtain these information. Nevertheless, in chapter 7 we will discuss how this approach can be applied to a more realistic attack scenario using shadow models. Furthermore, in this thesis, as the work of Shokri et al., we use an ML system (i.e. a binary classifier) as an attack model.

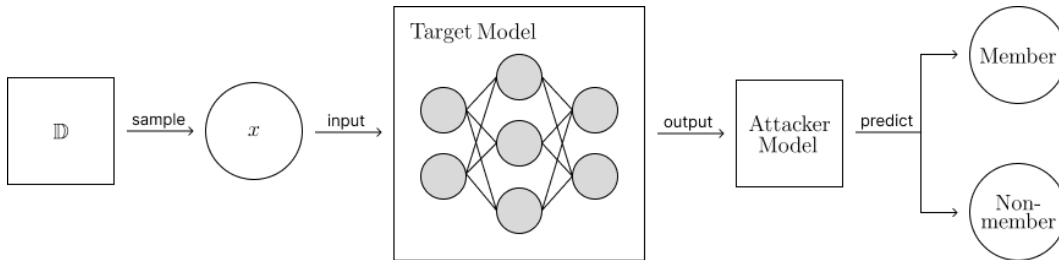


Figure 2.3.: Membership prediction of data point x by the attacker model

One of the most recent and promising publications that followed Shokri et al. is the work by Carlini et al. [2]. They present the likelihood ratio attack (LiRA) as a new membership inference attack. The novelty is that this paper defines the membership inference attack as performing a hypothesis test between the null hypothesis H_0 : a target model f_{target} was trained on a data point (x, y) and the alternative hypothesis H_1 : a target model f_{target} was not trained on a data point (x, y) . Therefore, Carlini et al. specify the two distributions $\mathbb{Q}_{\text{in}}(x, y) = \{f \leftarrow T(D \cup \{(x, y)\}) | D \leftarrow \mathbb{D}\}$, representing f_{target} was trained on (x, y) , and $\mathbb{Q}_{\text{out}}(x, y) = \{f \leftarrow T(D \setminus \{(x, y)\}) | D \leftarrow \mathbb{D}\}$, representing f_{target} was not trained on (x, y) . The attacker then performs a hypothesis test using the Neyman-Pearson lemma that predicts whether it is more likely

2. Background & related Work

that f_{target} comes from the distribution \mathbb{Q}_{in} or \mathbb{Q}_{out} :

$$\Lambda(f; x, y) = \frac{p(f|\mathbb{Q}_{\text{in}}(x, y))}{p(f|\mathbb{Q}_{\text{out}}(x, y))}$$

Since this test is not solvable, the paper define the computable distributions $\tilde{\mathbb{Q}}_{\text{in}}$ and $\tilde{\mathbb{Q}}_{\text{out}}$. $\tilde{\mathbb{Q}}_{\text{in}}$ and $\tilde{\mathbb{Q}}_{\text{out}}$ are described as the distributions of losses on a data point (x, y) for a target model f_{target} either trained, or not trained on the sample (x, y) which results in the final equation:

$$\Lambda(f; x, y) = \frac{p(\ell(f(x), y)|\tilde{\mathbb{Q}}_{\text{in}}(x, y))}{p(\ell(f(x), y)|\tilde{\mathbb{Q}}_{\text{out}}(x, y))}$$

To further reduce the complexity, Carlini et al. assume that $\tilde{\mathbb{Q}}_{\text{in}}$ and $\tilde{\mathbb{Q}}_{\text{out}}$ are Gaussian distributions, thus the attack only needs to estimate the mean and variance of each distribution. To estimate the distributions $\tilde{\mathbb{Q}}_{\text{in}}$ and $\tilde{\mathbb{Q}}_{\text{out}}$, they also employ shadow models. Carlini et al. show that their newly purposed attack outperforms earlier attacks and exemplify this by not only using metrics already known in the context of MIA (e.g. balanced accuracy, area under the curve, etc.) but also their newly propose evaluation metric, namely the receiver operating characteristic (ROC) curve. The ROC curve measures the true-positive rate (TPR) in relation to the false-positive rate (FPR). The TPR describes the number of positive examples (i.e. member data points) that were correctly classified as positive examples and the FPR indicates how many negative examples (i.e. non-member data points) were falsely classified as positive examples. Carlini et al. particularly emphasizes the importance of the false-positive rate and suggest that the true-positive rate should only be considered when the false-positive rate is very low (i.e. when the false-positive rate is between 0.001% and 0.1%). They reason that privacy does not require an average success metric such as balanced accuracy or area under the curve (AUC), as privacy aims to protect specific individuals. This property is taken into account by the ROC curve. Therefore, the proposed metric by Carlini et al. is also applied to evaluate the results of this work. Since LiRA is currently the best performing membership inference attack and is evaluated by the ROC curve, the results of LiRA presented in the paper by Carlini et al. are used for comparison purposes. Since Carlini et al. not only applied the ROC curve metric to their own attack, but also reevaluated attacks from previous publications, these can also be used for comparison with this work.

However, not only Carlini et al. criticizes the use of metrics such as the balanced accuracy. The work of Rezaei and Liu [15] also highlighted the FPR as an crucial metric. This work also reevaluates numerous attacks and finds that many of them are

not effective. However, Rezaei and Liu observe that when the data (i.e. all member and non-member data points) are subdivided into points that were correctly and incorrectly classified by the target model, the performance of attacks for incorrectly classified data points was more promising. In this work, we perform the experiments not only for the total set of member and non-member data points, but also for the subsets consisting of correctly and incorrectly classified points.

Furthermore, Shokri et al., Carlini et al. and other works have calibrated their attacks per class. In the literature this is often referred to as "per class-hardness" and means that, for example, a certain decision threshold is set or even a whole shadow model is trained per class. In this work, this should serve as motivation to also conduct the experiments per class (i.e. to subdivide the underlying data per class).

Another important aspect of membership inference attacks that will be explored in this paper is overfitting. Shokri et al. and others (e.g. [18, 21, 23]) analyzes the effect of overfitting on the effectiveness of membership inference. Overfitting describes when a model performs better on the training data set than on the data it has not seen during training. They observe that overfitting contributes to a model's information leakage, but is not the only reason for it. They explain this relationship by saying that membership inference relies on a model behaving differently on the training data than on the test data, which is exactly what overfitting describes. A metric also used by Shokri et al. to quantify the overfitting of a model is the train-test accuracy gap. The training-test accuracy gap is the difference between the train and test accuracy of the target model. In this work, overfitting will also be measured by the train-test accuracy gap and is related to the results in order to verify whether the statements made by Shokri et al. also apply to this work.

2.2.2. Time series

An important part of the experiments in this work are time series and time series analysis. A time series represents a sequence of observations of a variable at multiple time points ordered by time. Time series often show a high dependency over time in the form of trends or seasonal effects, which means the data points of a time series are not independent and identically distributed (iid). This is problematic since many conventional statistical methods are based on the assumption that the underlying data fulfils the iid property. Methodologies that deal with these specific characteristics of time series are referred to as time series analysis (cf. [13, 20]). With the additional data dimension provided by the checkpoints, the data in this paper can be described as time series. The metrics used for this data are therefore referred to as time series analysis.

2. Background & related Work

One statistical value to be analyzed in this work is the correlation between time series by the Pearson correlation coefficient (cf. Pearson [12]). In order to test time series for correlation, they must be stationary. A time series is stationary if there is no time dependency (i.e. they do not exhibit trends or seasonal patterns). To determine whether a time series is stationary or not, the Dickey-Fuller test (DF_{test}) or augmented Dickey-Fuller test (ADF_{test}) can be applied. The DF_{test} was developed in 1979 by Dickey and Fuller [4]. The ADF_{test} is a further development of this test and was developed by Said and Dickey [17] and can be used for a broader range of models (i.e. autoregressive models). One statistical value the ADF_{test} calculates is the p-value. To address the p-value of the test $ADF_{\text{test,p-value}}$ is written. For a given time series t it can then be assumed that t is stationary if $ADF_{\text{test,p-value}}(t) < 0.05$ holds true. Since in this thesis it is important to operate on time series that are stationary and it is not clear which order the autoregressive models of the time series of this work follow, the ADF_{test} is used in this thesis (cf. Greene [5]).

If the ADF test determines that the time series must be assumed to be non-stationary, stationarity can be created by the calculation of differences (i.e. the differences between successive observations). The results of the first process of differencing are often called first differences and is usually sufficient to make the time series stationary (cf. Hyndman and Athanasopoulos [6]). The first differences, denoted as $\text{Diff}_{(1)}$, for a given time series t are calculated by $x'_i = x_{i+1} - x_i$ where x describes all time points of t .

If it is certain that time series are stationary, two time series can be compared using the Pearson correlation coefficient. The Pearson correlation coefficient was defined by Pearson [12] in 1895. The Pearson Correlation Coefficient measures the linear correlation of two variables. The resulting ratio lies between -1 and 1. 1 means a strong positive correlation and -1 a strong negative correlation, 0 stands for no correlation (cf. Kirch [8]).

In this work the Pearson correlation coefficient is used in order to analyse and compare time series. If a time series is tested as non-stationary, differences will be calculated.

2.2.3. Support vector machines

Support vector machines (SVMs) are a set of supervised learning methods that are used for different purposes in machine learning, such as binary classification. SVMs were developed in 1963 by Vladimir N. Vapnik and Alexey Ya. Chervonenkis and then improved by Boser, Guyon, and Vapnik [1] in 1992 and Cortes and Vapnik [3] in 1995. Since SVMs are simple but well performing classification systems, this method will be applied in this work in order to train an attacker model (i.e. a binary

classifier).

3. Approach

The goal of this work is to test whether checkpoints contain valuable information that can help better distinguish membership data points from non-membership data points. The associated hypothesis H0 and H1 are denoted as:

Hypothesis 0 (H0). *Checkpoints do not provide useful information to make a more accurate conclusion about whether or not a data point was part of the training data set of the target model.*

Hypothesis 1 (H1). *Checkpoints provide useful information to make a more accurate conclusion about whether or not a data point was part of the training data set of the target model.*

To test these hypotheses, two exploratory experiments are conducted. Both experiments are carried out from the perspective of the model owner (i.e. full access is granted to the target model including the checkpoints and the used data sets). The reason for this is that the methods explored in the experiments are intended to contribute to a better evaluation of the privacy of a target model from the perspective of the model owner, rather than to present a realistic attack scenario. The experiments are described in the subsequent sections.

3.1. Data sets & target model

Both experiments use the same data sets and target models. As data sets $D_{\text{CIFAR-10}}$ and $D_{\text{CIFAR-100}}$ are used. The training data sets or, in this context, also called member data sets, are denoted as $D_{\text{CIFAR-10,member}}$ and $D_{\text{CIFAR-100,member}}$. The test data sets or also called non-member data sets, are described as $D_{\text{CIFAR-10,non-member}}$ and $D_{\text{CIFAR-100,non-member}}$. For each data set, one target model is trained by a training algorithm:

$$\begin{aligned} f_{\text{target,CIFAR-10}} &\leftarrow \mathcal{T}(D_{\text{CIFAR-10,member}}), \\ f_{\text{target,CIFAR-100}} &\leftarrow \mathcal{T}(D_{\text{CIFAR-100,member}}) \end{aligned}$$

Since checkpoints are now introduced as a further dimension, the target model at checkpoint $c_j \in C$, where C are all checkpoints and c_j the j^{th} checkpoint, is noted

3. Approach

as $f_{\text{target},j}$. When f_{target} is noted without the checkpoint index j , the target model including all checkpoints is meant. N_c describes the total number of checkpoints, which means that $j \in \{1, \dots, N_c\}$ and thus c_{N_c} notes the last checkpoint. A checkpoint is generated after n_b batches during each epoch and at the end of each epoch. Batches are the smallest data unit during the training of a model. A batch contains a small part of the size, called batch size, of the whole training data set. If all batches are run through, it is called an epoch. For each checkpoint $c_j \in C$ and data point $x_i \in D_{\text{CIFAR-10}}$ or data point $x_i \in D_{\text{CIFAR-100}}$ there is an output vector of confidence values $\hat{y}_{i,j}$, where i marks the output for the i^{th} data point and j the output for the j^{th} checkpoint. This can also be described as $f_{\text{target},j}(x_i) = \hat{y}_{i,j}$. In the following, the output vector previously described as $\hat{y}_{i,j}$ is referred to as $\text{conf}_{i,j}$. If $\text{Conf}_{\text{member}}$ or $\text{Conf}_{\text{non-member}}$ is noted, the confidence values of all member or non-member data points are meant respectively.

3.2. Experiment I

In order to prove H0 or H1, in experiment I it is explored whether the confidence values of the member and non-member data points show differences over all checkpoints. To test this, the time series of member and non-member data points resulting from the confidence values of all checkpoints are compared using time series analysis. If differing characteristics become evident, the hypothesis H0 can be disproved and H1 can be accepted.

3.2.1. Methodology

For the first experiment, schematized in figure 3.1, the confidence values of member $\text{Conf}_{\text{member}}$ and non-member $\text{Conf}_{\text{non-member}}$ data points of all checkpoints and for each data set $D_{\text{CIFAR-10}}$ and $D_{\text{CIFAR-100}}$ separately are used to form the time series T_{member} and $T_{\text{non-member}}$. These time series T_{member} and $T_{\text{non-member}}$ are further aggregated by different averaging methods. For experiment I, there are three different aggregations for T_{member} and $T_{\text{non-member}}$:

1. Aggregation per confidence class
2. Aggregation per confidence class and true class
3. Aggregation per confidence class, true class and predicted class

Each aggregation analyzes a certain level of detail. The first part provides a coarser

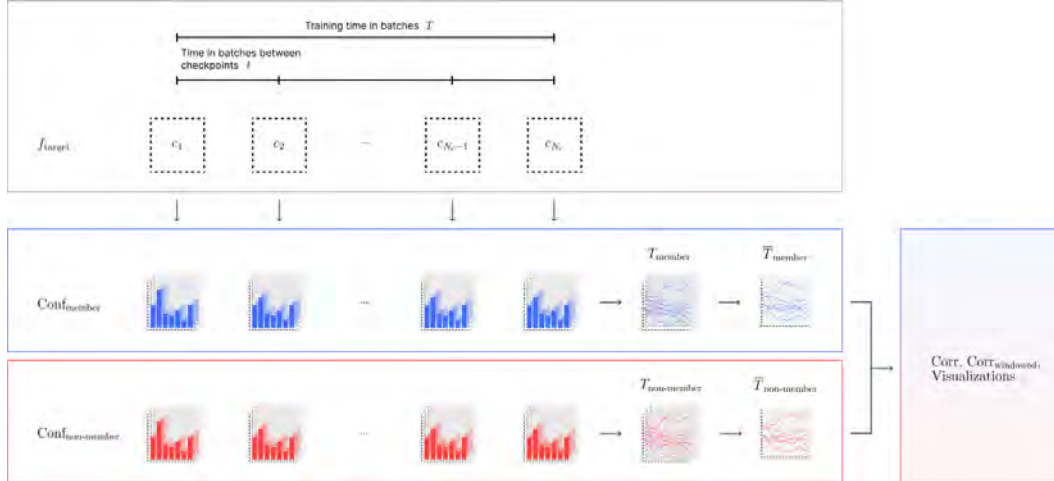


Figure 3.1.: Schema of experiment I

overview, whereas the second part examine in more detail where dissimilarities that may have already been identified in part one come from or whether new anomalies can be found. The third part provides the most in-depth view and builds on findings of the previous aggregations. The resulting time series of these aggregations are denoted as \bar{T}_{member} and $\bar{T}_{\text{non-member}}$. For the first and second type of aggregation the time series \bar{T}_{member} and $\bar{T}_{\text{non-member}}$ are further subdivided into an unfiltered part, into a part filtered by confidence values only leading to correct classifications and a part filtered by confidence values only leading to incorrect classifications. For the resulting time series \bar{T}_{member} and $\bar{T}_{\text{non-member}}$, ADF_{test} is used to check whether they are stationary or not. If they are not stationary, $\text{Diff}_{(1)}$ is applied to each time series and then ADF_{test} is used again to check if the time series are now stationary. This process is repeated until all time series are stationary. The resulting time series \bar{T}_{member} are then compared with the time series $\bar{T}_{\text{non-member}}$ via the Pearson correlation Corr and the windowed Pearson correlation $\text{Corr}_{\text{windowed}}$. The windowed Pearson correlation $\text{Corr}_{\text{windowed}}$ is defined as the Pearson correlation Corr between two time series where only a window of size x_w of both time series are considered (i.e. only x_w time points are considered). This window slides over the two time series with a step size of x_s . For each point of the window, the Pearson correlation Corr is again calculated. The window slides from the beginning of the time series to the end. Using this technique, it can be analysed if the correlation is more distinct during a specific time of the overall times series. The resulting correlation coefficients are subsequently called $\text{Corr}_{\text{member, non-member}}$ and $\text{Corr}_{\text{windowed, member, non-member}}$ respectively. These aggregated time series \bar{T}_{member} and $\bar{T}_{\text{non-member}}$, the correlation coefficients $\text{Corr}_{\text{member, non-member}}$ and windowed correlation coefficients $\text{Corr}_{\text{windowed, member, non-member}}$ are then visualized.

3. Approach

3.3. Experiment II

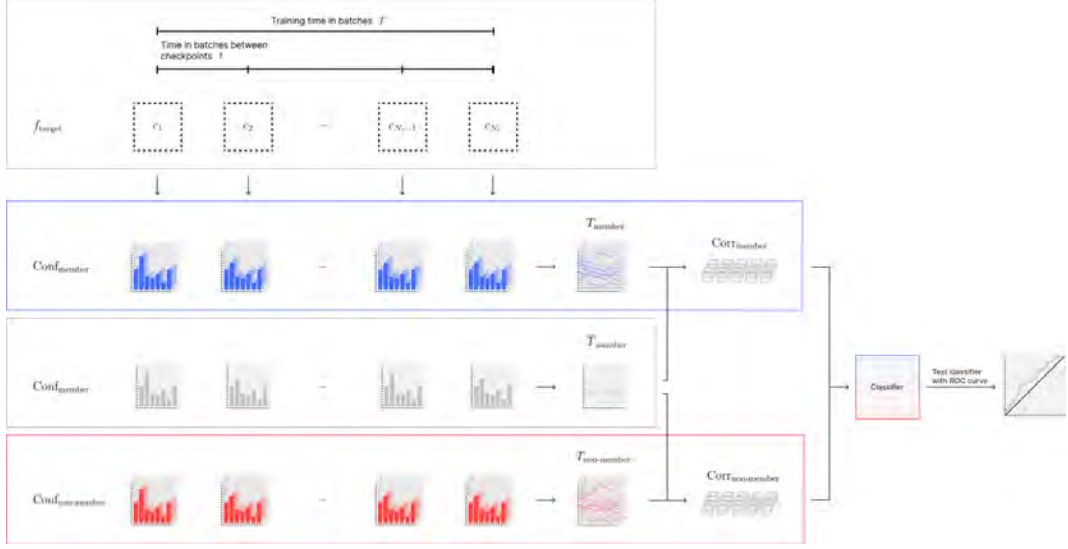


Figure 3.2.: Schema of experiment II

The second experiment aims at exploring new effective MI attacks using checkpoints as additional information. In order to prove H0 or H1, the performance of the resulting attacks will be tested with the ROC curve metric. If the resulting attacks will perform better than comparable MI attacks according to the ROC metric, the hypothesis H0 can be disproved and H1 can be accepted.

3.3.1. Methodology

In the second experiment, visualized in figure 3.2, a new attack on membership inference will be performed using checkpoints in different settings. Since in experiment I it was found that the most promising results come from confidence values that lead to incorrect classifications, this experiment will also focus on this subset. As for experiment I, the attack is applied to $D_{\text{CIFAR-10}}$ and $D_{\text{CIFAR-100}}$ separately and the confidence values $\text{Conf}_{\text{member}}$ and $\text{Conf}_{\text{non-member}}$ of a specific number of checkpoints are again used to form the time series T_{member} and $T_{\text{non-member}}$. For experiment II, there are two parts, each with a different number of checkpoints used for $\text{Conf}_{\text{member}}$ and $\text{Conf}_{\text{non-member}}$:

1. All N_c checkpoints are used
2. The first n_c checkpoints are used, where $n_c < N_c$

n_c is chosen such that the train-test accuracy gap of checkpoint c_{n_c} is significantly lower than that of checkpoint c_{N_c} . The idea behind this approach is that research has shown that attacks on models that are subject to strong overfitting (i.e. also measured by the train-test accuracy gap) can sometimes yield better attack results than those that are not so strongly overfitted. Therefore, it is tested whether a more overfitted target model (i.e. a target model f_{target,N_c}) provides a better attack surface for the performed attacks than a less overfitted target model (i.e. the target model f_{target,n_c}).

The resulting time series T_{member} is used two times: once without further aggregation and once aggregated per confidence class. The latter results in the time series \bar{T}_{member} and is representative for the target model. For the resulting time series \bar{T}_{member} , T_{member} and $T_{\text{non-member}}$, again ADF_{test} is used to check whether they are stationary or not. If they are not stationary, $\text{Diff}_{(1)}$ is applied to each time series and ADF_{test} is used again to check if the time series are now stationary. This process is repeated until all time series are stationary. The resulting time series T_{member} and $T_{\text{non-member}}$ are compared with the time series \bar{T}_{member} using the Pearson correlation Corr . This is done for all N_{member} and $N_{\text{non-member}}$ data points in T_{member} and $T_{\text{non-member}}$, respectively, and all N_{cl} confidence classes. Subsequently, the resulting correlation coefficients are called $\text{Corr}_{\text{member}}$ and $\text{Corr}_{\text{non-member}}$. $\text{Corr}_{\text{member}}$ and $\text{Corr}_{\text{non-member}}$ are then used as the data set $D_{\mathcal{A}}$ to train the adversary \mathcal{A} (i.e. its attacker model $f_{\mathcal{A}}$ which is a binary classifier) and test it afterwards. The attacker model is trained in two ways: over all true classes and per true class (i.e. to build an attack that learns per class-hardness scores). The resulting trained adversaries \mathcal{A} are tested using a ROC curve.

4. Implementation

4.1. Software tools & libraries

For the implementation purposes, Python is used as the programming language. Python has many scientific libraries that are also used in this work (e.g. NumPy, pandas, SciPy, statsmodels, scikit-learn Pytorch, matplotlib and seaborn). As the development environment, JupyterLab is employed as it provides a suitable web-based interactive interface for machine learning and for performing data analysis.

4.2. Data sets & target model

For the training of the target models $f_{\text{target,CIFAR-10}}$ and $f_{\text{target,CIFAR-100}}$ 50% of the training data of the respective data set (i.e. $D_{\text{CIFAR-10}}$ and $D_{\text{CIFAR-100}}$) is used. $D_{\text{CIFAR-10, member}}$, $D_{\text{CIFAR-10, non-member}}$, $D_{\text{CIFAR-100, member}}$ and $D_{\text{CIFAR-100, non-member}}$ thus consist of 25,000 data points each. As network architecture for the target models a wide ResNet [24] with a depth of 28 and a width of 2 is employed. All other network parameters can be found in the table 4.1. For the implementation, we utilize the PyTorch implementation [14] of the original paper code [11] of the wide ResNet. Additionally, to train the target models, again the Python library PyTorch is used.

During training, the data sets $D_{\text{CIFAR-10, member}}$ and $D_{\text{CIFAR-100, member}}$ are divided into batches with a batch size of 128. For a data set size of 25,000, this results in 195 batches (i.e. $25,000/128 = 195$), where the last batch has a size of only 40 data points (i.e. $25,000 \bmod 128 = 40$). Furthermore, as by Carlini et al., 200 epochs are defined as the total training length. The learning rate is implemented as in the original paper on Wide ResNet by Zagoruyko and Komodakis. The learning rate is adjusted after a certain number of epochs. The exact learning rates per epoch can be taken from the table 4.2. Further training parameters can be found in table and 4.3. The checkpoints are stored after a batch interval of 100 (i.e. $n_b = 100$). Since also at the end of each epoch an checkpoint is saved, in total, 400 checkpoints are stored (i.e. 200×2). For each checkpoint a test- and train-accuracy is calculated during the training. After training, all data points, separated into member and non-

4. Implementation

member data sets, are fed through the checkpoints C of the trained target models $f_{\text{target, CIFAR-10}}$ and $f_{\text{target, CIFAR-100}}$. The output values (i.e. the confidence values) $\text{Conf}_{\text{member}}$ and $\text{Conf}_{\text{non-member}}$ are then stored. The target models $f_{\text{target, CIFAR-10}}$ and $f_{\text{target, CIFAR-100}}$ and the generated data $\text{Conf}_{\text{member}}$ and $\text{Conf}_{\text{non-member}}$ (i.e. for CIFAR-10 and CIFAR-100, respectively) serve as the starting point for experiments I and II.

Parameter	Value	Epoch	Learning Rate
Network architecture	wide ResNet	0 - 60	0.1
Depth	28	61 - 120	0.02
Widen factor	2	121 - 160	0.004
Drop out rate	0.3	161 - 200	0.0008

Table 4.1.: Network hyper parameters

Table 4.2.: Learning rate over time

Parameter	Value
Batch size	128
Epochs	200
Optimizer	Stochastic Gradient Descent
Momentum	0.9
Weight decay	0.0005
Initial learning rate	0.1
Learning rate decay ratio	0.2

Table 4.3.: Training hyper parameters

4.3. Experiments I

For the experiment I, the resulting data $\text{Conf}_{\text{member}}$ and $\text{Conf}_{\text{non-member}}$ are further processed: the time series T_{member} and $T_{\text{non-member}}$ are build, which are then aggregated and on which stationarity is tested and enforced. For these steps, NumPy and Pandas are used for basic data processing and handling, whereas statsmodels is employed for the augmented Dicky-Fuller test. The resulting time series \bar{T}_{member} and $\bar{T}_{\text{non-member}}$ are stored and used for the execution of the time series analysis. SciPy is used for the Pearson correlation and the windowed Pearson correlation. For the windowed Pearson correlation a window size of 100 and a step size of 10 is selected. To visualize the results we employ seaborn and matplotlib.

4.4. Experiments II

For the implementation of experiment II, at first the resulting data $\text{Conf}_{\text{member}}$ and $\text{Conf}_{\text{non-member}}$ is again further processed. The time series T_{member} and $T_{\text{non-member}}$ are build, but this time only for T_{member} a further aggregation takes place. Then, on all resulting time series \bar{T}_{member} , T_{member} $T_{\text{non-member}}$ stationarity is tested and enforced. Furthermore, for \bar{T}_{member} , T_{member} $T_{\text{non-member}}$ the Pearson correlation is calculated so that it results in the the data set $D_{\mathcal{A}}$. Again, NumPy, Pandas and statsmodels are used for these purposes. In the next step the attacker model $f_{\mathcal{A}}$ is trained. Therefore the data set $D_{\mathcal{A}}$ is split into 75% training data $D_{\mathcal{A}-\text{train}}$ and 25% test data $D_{\mathcal{A}-\text{test}}$. For n_C 120 is chosen (i.e. for the training of some attacker models, only the first 120 checkpoints are considered). As attacker model architecture a SVM from the scikit-learn library is used as a binary classifier. In the last step the trained attacker model is tested via ROC. The ROC functionality is again provided by the scikit-learn library. For visualisation purposes, matplotlib was used again.

5. Results

In the following section, the results of the experiment I and II are presented.

5.1. Data sets & target model

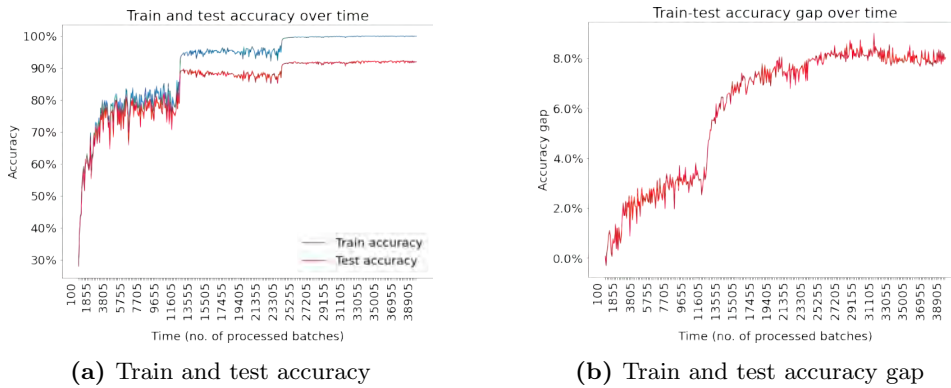


Figure 5.1.: CIFAR-10 target model

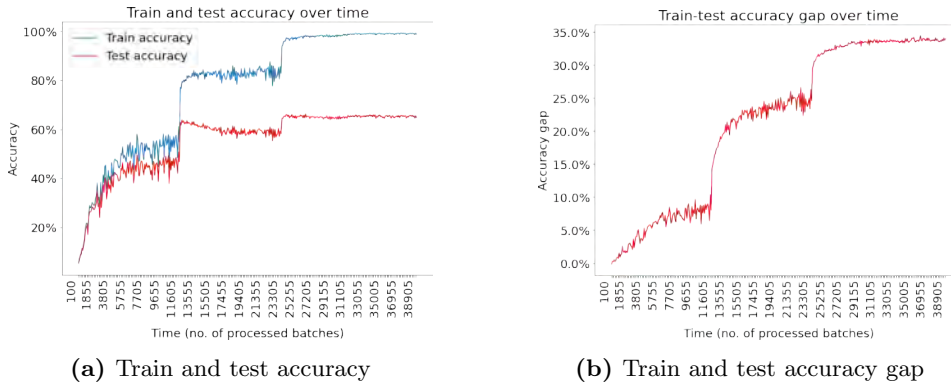


Figure 5.2.: CIFAR-100 target model

For the CIFAR-10 data set, the training of the target model f_{Target} , CIFAR-10 resulted in a final training accuracy of 99.92% and a test accuracy of 91.91% (cf. 5.1a). The difference between training and test accuracy was thus 8.02% (cf. 5.1b). This is a

5. Results

better result in terms of test accuracy and train-test gap compared to the work of Carlini et al. For CIFAR-100, the target model $f_{\text{target, CIFAR-100}}$ achieved a training accuracy of 98.88% and a test accuracy of 64.91%, giving an absolute train-test accuracy gap of 33.97% (cf. ??). Here, too, the target model used is better than the one used by Carlini et al. in terms of the train-test accuracy and gap.

Furthermore, the figures ?? and ?? show that the accuracies jump after 60 and 120 epochs. This is due to the fact that the learning rate is adjusted at these points in time (cf. table 4.2). The learning rate is also adjusted after 160 epochs, but no significant change in accuracies can be seen. The leaps are stronger for the CIFAR 100 model than for the CIFAR 10 model.

5.2. Experiment I

5.2.1. CIFAR-10

Aggregation per confidence class

To get a first overview, the aggregated time series based on all confidence values and the confidence values that led to correct or incorrect classifications were plotted separately. These plots show that at first glance the time series of members and non-members based on incorrect classifications have the strongest dissimilarities. Examples can be found in the figures A.1 in the appendix.

The Pearson correlation coefficients of these time series are visualised in the figures 5.3. The shown heat maps confirm the previous statement: the greatest differences between member and non-member time series were archived based on incorrectly classified data. It could also be observed that for incorrectly classified data some confidence classes showed larger (e.g. "bird") or smaller (e.g. "horse") differences.

The windowed Pearson correlation coefficients for the incorrectly classified data in figure 5.4 show that these dissimilarities are more significant towards the end of the model's training. In the same figure, the train-test accuracy gap is plotted above the windowed Pearson correlation coefficients. It can be seen that as the dissimilarity increases, there also seems to be a greater discrepancy between the train and test accuracy.

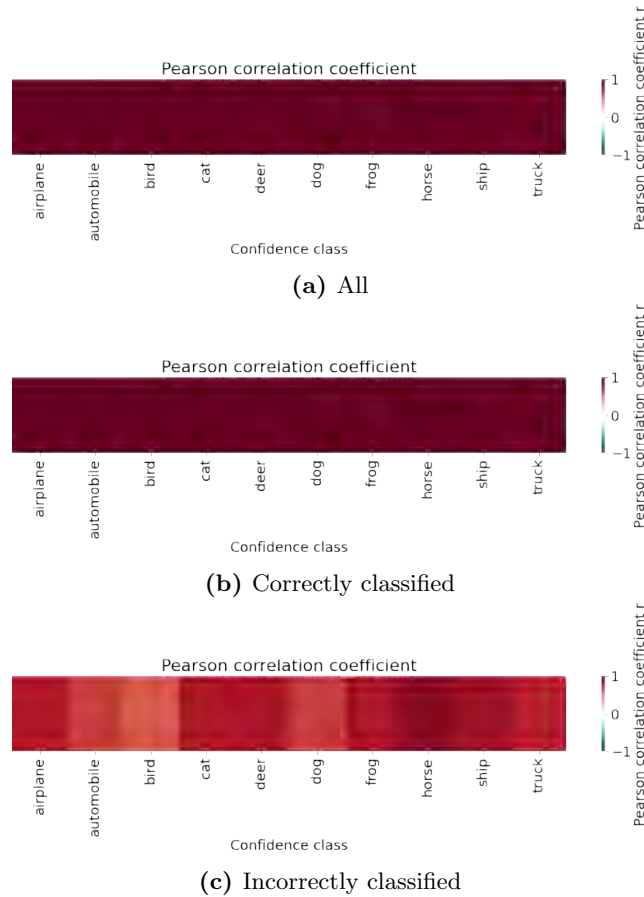


Figure 5.3.: CIFAR-10: Pearson correlation coefficients per confidence class of member vs. non-member time series of aggregated confidence values

Aggregation per confidence class & true class

For part II, again, the aggregated time series were visualized. Even with this more detailed data analysis (i.e. subdivided by true class), only the time series based on incorrect classifications showed clear differences. Examples can be found in the figures A.2 in the appendix.

The same picture emerged for the correlation coefficients. We found that there were differences between the time series of members and non-members only for incorrectly classified data points. The complete heat maps can be found in the figures A.3 in the appendix. For incorrectly classified data, again, some true classes in combination with a confidence class showed stronger differences (e.g. true class = "truck", confidence class = "frog").

5. Results

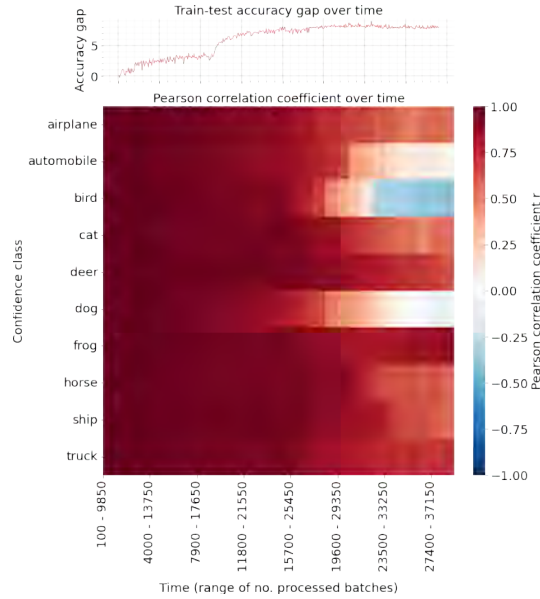


Figure 5.4.: CIFAR-10: Windowed Pearson correlation coefficients per confidence class of member vs. non-member time series of aggregated, incorrectly classified confidence values with train-test accuracy gap over time

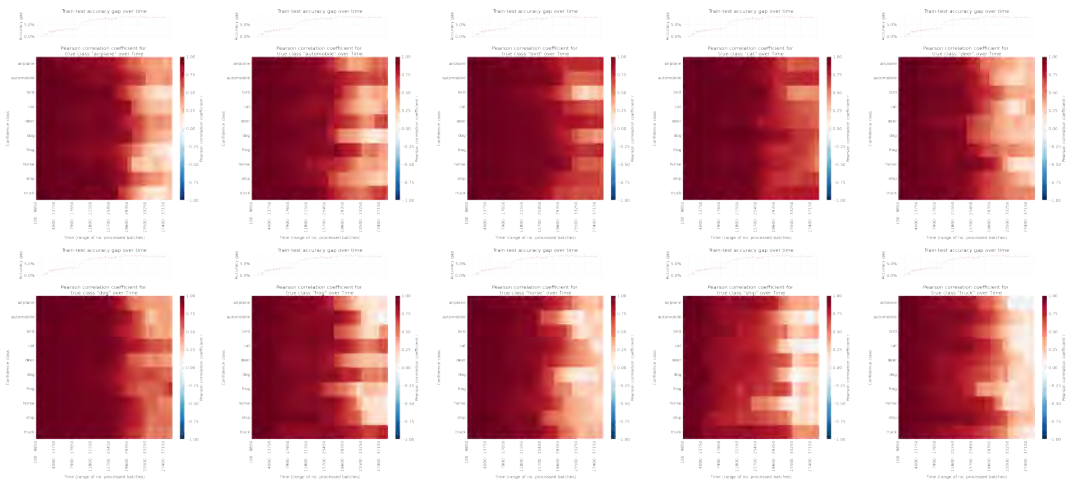


Figure 5.5.: CIFAR-10: Windowed Pearson correlation coefficients per true and confidence class of member vs. non-member time series of aggregated, incorrectly classified confidence values with train-test accuracy gap over time

Looking at the windowed correlation coefficients of the incorrectly classified data in the figure 5.5, it becomes apparent that the later checkpoints again lead to greater dissimilarities.

Aggregation per confidence class, true class & predicted class

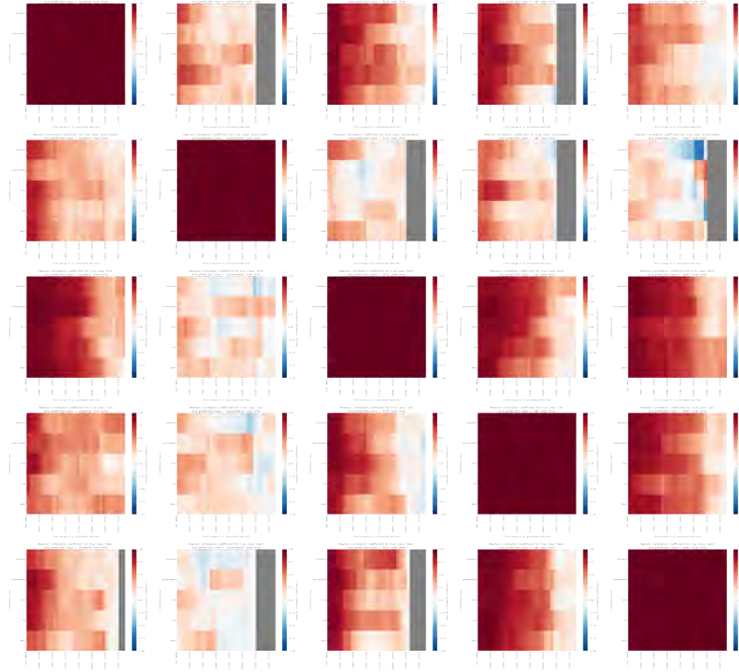


Figure 5.6.: CIFAR-10: Windowed Pearson correlation coefficients per true, predicted and confidence class of member vs. non-member time series of aggregated confidence values for confidence, true and predicted classes "airplane", "automobile", "bird", "cat", "deer"

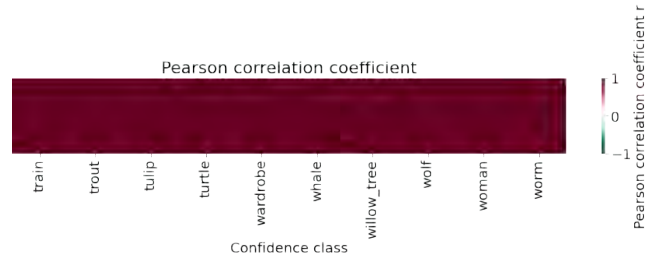
In the last part, we could see that if the predicted class corresponds to the true class (i.e. a correct classification), there are hardly any differences. On the other hand, if the predicted class does not correspond to the true class (i.e. a false classification), stronger dissimilarities can be seen. The corresponding time series plots and correlation coefficient heat maps can be found in the appendix figures A.4 and A.5

Here too, the windowed correlation coefficients depicted in figure 5.6 show a tendency towards greater deviations between the data points of members and non-members towards the end of the training. The grey fields indicate that there was not enough data available to provide meaningful values. It is noticeable that for some true classes in combination with a predicted class and confidence class, the differences were much stronger than in figure 5.5.

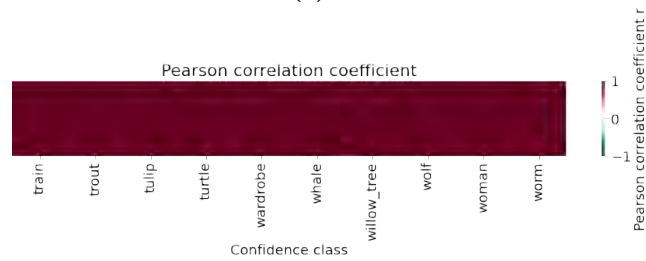
5. Results

5.2.2. CIFAR-100

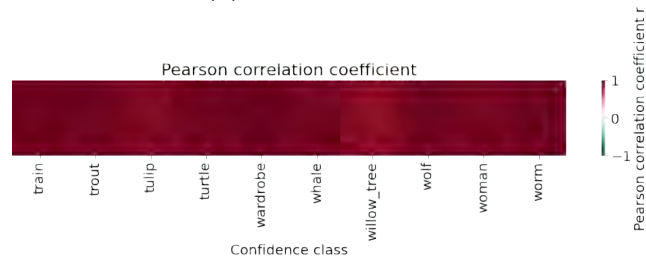
Aggregation per confidence class



(a) All



(b) Correctly classified



(c) Incorrectly classified

Figure 5.7.: CIFAR-100: Pearson correlation coefficients per confidence class of member vs. non-member time series of aggregated, correctly classified confidence values for confidence classes "train", "trout", "tulip", "turtle", "wardrobe", "whale", "willow_tree", "wolf", "woman", "worm"

While similar overall results could be observed for CIFAR-100 as for CIFAR-10, it is noticeable that for part I, the differences between the aggregated time series of incorrectly classified member and non-member data points are not as strong as for CIFAR-10. This can be viewed in the figures 5.7. Further plots can be found in the appendix figures A.6. Their correlation coefficients over time depicted in plot 5.8 suggest that the deviations become stronger towards the end of the model's training. This is equivalent to the result of CIFAR-10.

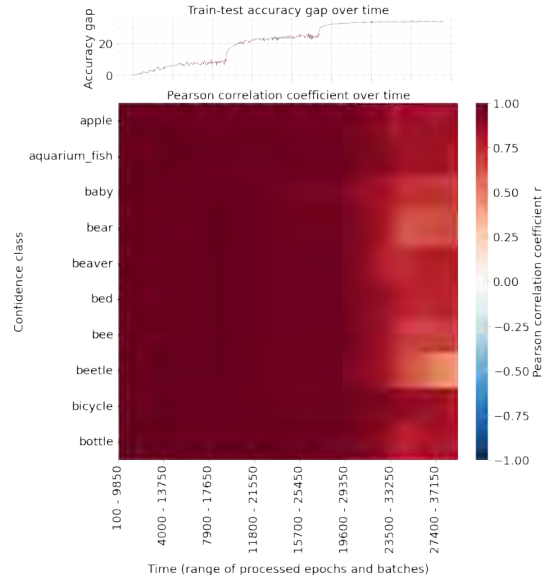


Figure 5.8.: CIFAR-100: Windowed Pearson correlation coefficients per confidence class of member vs. non-member time series of aggregated, incorrectly classified confidence values with train-test accuracy gap over time for confidence classes "apple", "aquarium_fish", "baby", "bear", "beaver", "bed", "bee", "beetle", "bicycle", "bottle"

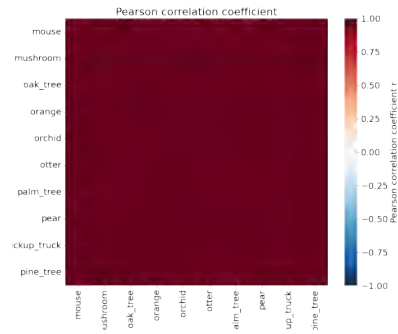
Aggregation per confidence class & true class

In part II, it was found that differences between members and non-members could not only be identified based on incorrectly classified data but also for time series based on correctly classified data (cf. 5.9). But, here too, it could be observed that the strongest dissimilarities are present among the incorrectly classified data. Additional plots can be found in the appendix figures A.7. Again, a clear temporal trend can be found: the correlation of member and non-member time series decreases towards the end. Example figures can be found in the appendix A.8.

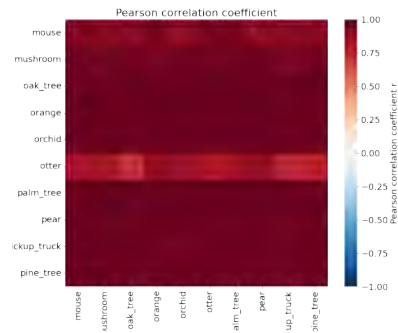
Aggregation per confidence class, true class & predicted class

The previously found results were also confirmed in part III. However, it is interesting to note, that some true classes combined with a predicted class and confidence class have substantially larger dissimilarities than others (cf. 5.10). Furthermore, there are significantly more grey areas for CIFAR-100 than for CIFAR-10. This is because there are also significantly more classes and therefore less data available per true class and predicted class. The temporal trend of the correlation coefficients of this analysis corresponds to previous findings. A visualisation illustrating this can be

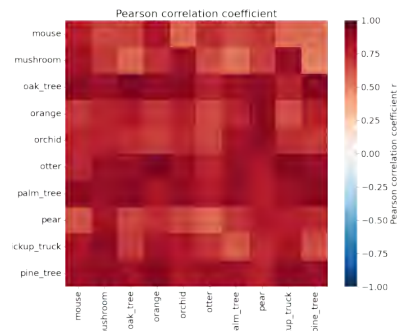
5. Results



(a) All



(b) Correctly classified



(c) Incorrectly classified

Figure 5.9.: CIFAR-100: Pearson correlation coefficients per true and confidence class of member vs. non-member time series of aggregated confidence values for confidence classes "apple", "aquarium_fish", "baby", "bear", "beaver", "bed", "bee", "beetle", "bicycle", "bottle"

found in the appendix A.9 and A.10.

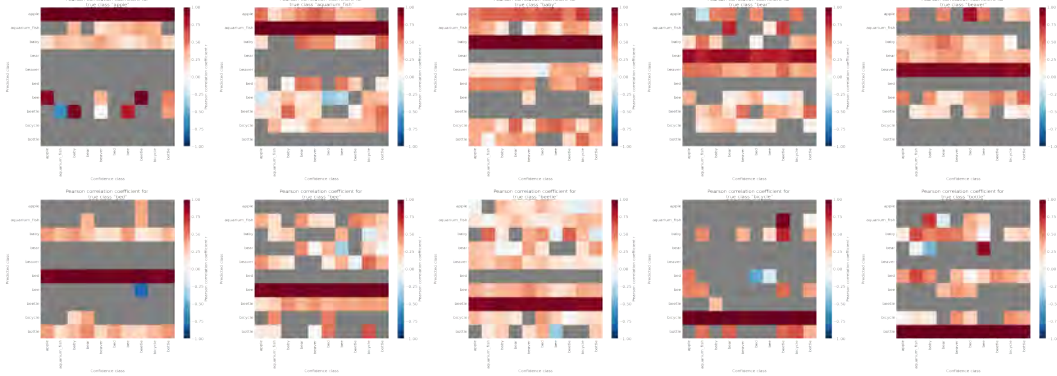


Figure 5.10.: CIFAR-100: Pearson correlation coefficients per true class, predicted class and confidence class of member vs. non-member time series of aggregated confidence values for confidence, true and predicted classes "apple", "aquarium_fish", "baby", "bear", "beaver", "bed", "bee", "beetle", "bicycle", "bottle"

5.3. Experiment II

5.3.1. CIFAR-10

All 200 checkpoints are used

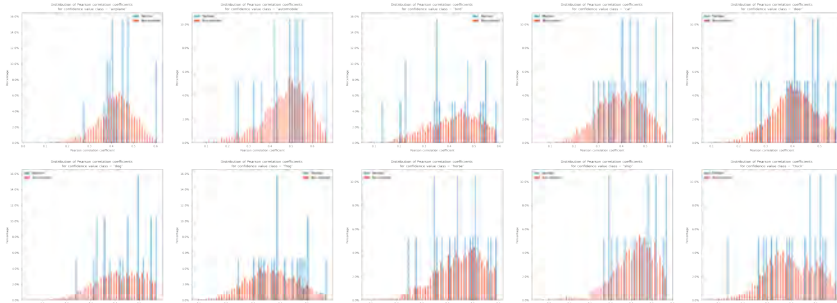


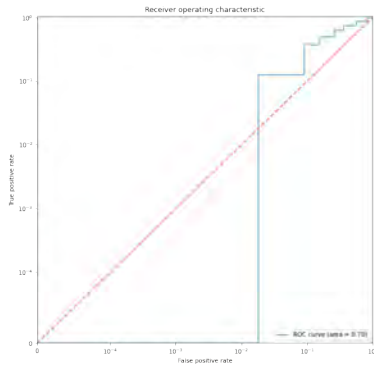
Figure 5.11.: CIFAR-10: Histogram of Pearson correlation coefficients per confidence class of member and non-member time series of aggregated, incorrectly classified confidence values

For experiment II, part I, the plotted histograms in 5.11 show an overview of the distributions of the Pearson correlation coefficients the attack was based on. At first sight, no clear separation of the two distributions is visible and the correlation coefficients of the member data points seem to be unevenly distributed. The latter

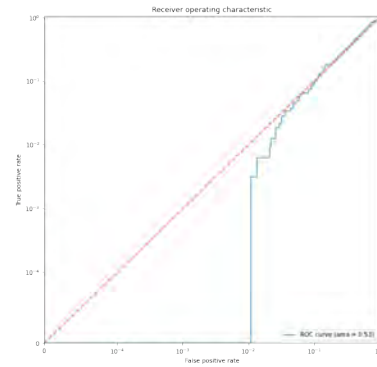
5. Results

could be due to the fact that only few data was available: The histograms of the correlation coefficients of members consist of only 19 data points. The results of the histograms indicate that it does not seem to be easy to distinguish between data points from members and non-members and that the results of the attack may not be interpreted as meaningful since too little data is available. As expected, the results of the attack based on the available data suggest that no effective adversary could be trained (cf. 5.12a).

For part I, there was not enough data available to carry out the attack per true class.



(a) max. epoch 200



(b) max. epoch 120

Figure 5.12.: CIFAR-10: ROC curves of membership inference attack based on Pearson correlation coefficients of member and non-member time series of aggregated, incorrectly classified confidence values

The first 120 checkpoints are used

For part II, since in epoch 120 an adjustment of the learning rate took place and a jump in the difference between test and training accuracy was observed, this time point was chosen as the maximum epoch. Although, the jump in the test-train accuracy gap was significantly higher for CIFAR-100 than for CIFAR-10, for both data sets the same maximum epoch was chosen in order to be able to compare the results.

Unlike in part I, the histograms seen in 5.13 show that without subdividing the data per true class, the correlation coefficients of members look more evenly distributed. This might be due to the fact that more data is available (i.e. 1292 data points). Now, the distributions of the correlation coefficients of members and non-members

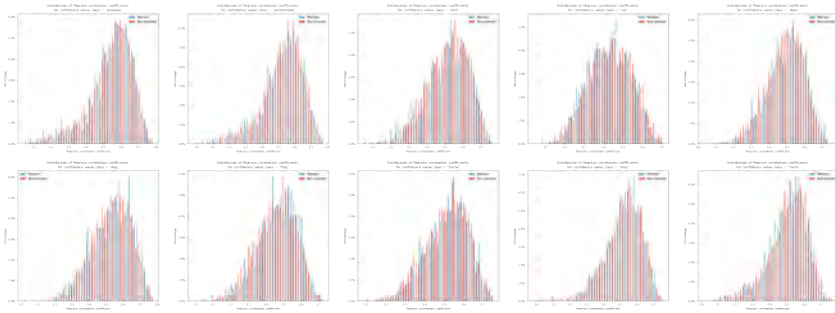


Figure 5.13.: CIFAR-10: Histogram of Pearson correlation coefficients per confidence class of member and non-member time series of aggregated, incorrectly classified confidence values with max. epoch 120

look very similar. The results for the corresponding attack suggested that no effective attack can be carried out despite the larger amount of data available (cf. 5.12b).

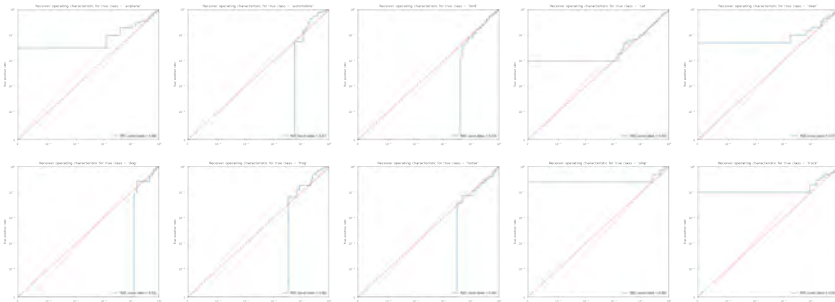


Figure 5.14.: CIFAR-10: ROC curve of membership inference attack based on Pearson correlation coefficients per true class of member and non-member time series of aggregated, incorrectly classified confidence values with max. epoch 120

For the histograms of the correlation coefficients subdivided per true class, we found that for some true classes there were not enough data points to present a meaningful distribution and that a clear separation between members and non-members was not given at first glance. The full histograms can be found in the appendix figure A.11. Nevertheless, the ROC curves for the attacks that were performed per true class, show that for certain true classes (e.g. for the true class "airplane") good performing attacks can be trained (cf. 5.14). The TPR and FPR of the best performing true class compared to the results of other methods can be found in table 5.1.

5. Results

Method	max. epoch	TPR @ 0.001% FPR	TPR @ 0.1% FPR
Carlini et al. [2]		2.2%	8.4%
Sablayrolles et al. [16]		0.1%	1.7%
Long et al. [10]		0.0%	2.2%
Watson et al. [22]		0.1%	1.3%
This work	120	6.7%	6.7%

Table 5.1.: CIFAR-10: Comparison between other methods and the work’s method for true class ”frog”

5.3.2. CIFAR-100

All 200 checkpoints are used

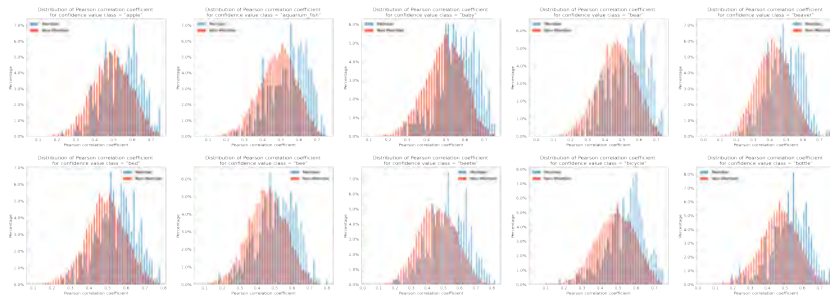


Figure 5.15.: CIFAR-100: Histogram of Pearson correlation coefficients per confidence class of member and non-member time series of aggregated, incorrectly classified confidence values for 10 sample confidence classes

For CIFAR-100, it can already be seen in the distributions of the correlation coefficients of members and non-members that even though they look very similar, there is a slight offset of the distributions (cf. 5.15). Furthermore, the ROC curve in figure 5.16a shows that an attack could be carried out that reaches a good performance (i.e. a high TPR at a low FPR). A comparison of this attack to other methods can be found in table 5.2. Again, there was not enough data points to perform the attacks per true class.

The first 120 checkpoints are used

In part II, only the data points up to the 120 epoch were used. The distributions of the correlation coefficients of members and non-members not separated by the true class show that they are now more similar than in part I. The appendix figure A.13

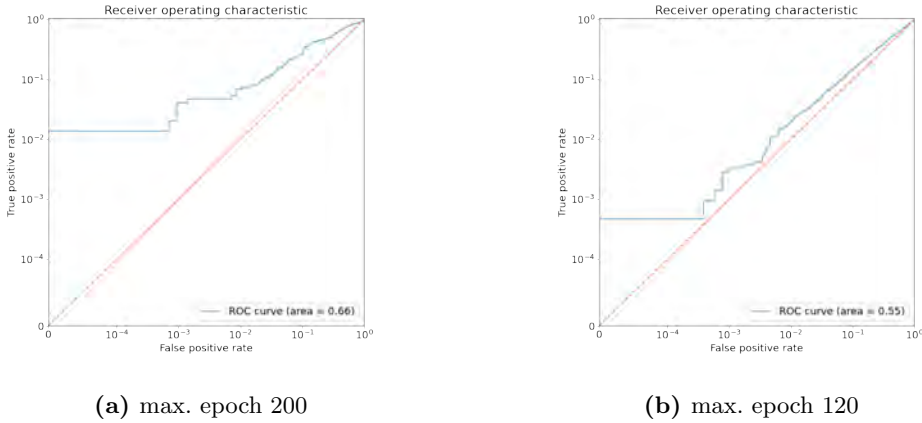


Figure 5.16.: CIFAR-100: ROC curves of membership inference attack based on Pearson correlation coefficients of member and non-member time series of aggregated, incorrectly classified confidence values

shows all distributions. The corresponding ROC curve also shows that the attack performs worse than in part I (cf. 5.16b).

Method	max. epoch	TPR @ 0.001% FPR	TPR @ 0.1% FPR
Carlini et al. [2]		11.2%	27.6%
Sablayrolles et al. [16]		0.8%	7.4%
Long et al. [10]		0.0%	4.7%
Watson et al. [22]		0.9%	5.4%
This work	200	1.4%	4.1%
	120	0.0%	0.0%

Table 5.2.: CIFAR-100: Comparison between other methods and the work’s method

The ROC curves for the attacks that were carried out per true class show that some attacks achieve better results (e.g. the true class "apple") than the previous attack. The ROC curves for this attack can be found in the appendix figure A.14. However, since there is little data available for each true class, these results might not be assumed to be sufficiently significant.

6. Conclusion

6.1. Experiment I

For experiment I, it was observed that the correlation between member and non-member data points in terms of all their confidence values and all their confidence values leading to correct classifications over time was very high. Hence, in this employed experimental setup, it can be assumed that little to no information can be gained for a better distinction between member and non-member data points. However, when only incorrectly classified points were evaluated, the correlation decreased and it can thus be assumed that the use of checkpoints in this setting could help to better identify member data points. This is similar to what Rezaei and Liu [15] found in their work. Since privacy aims to protect each individual, this incorrectly classified data is also relevant when it comes to privacy. Thus, even if a target model poses a privacy risk only for incorrectly classified data points, it still poses a privacy risk for each of these samples.

Furthermore, when analysing the correlation coefficients of the incorrectly classified points per true and predicted class, it was found that some combinations were more dissimilar than others and thus it can be concluded that some classes are more vulnerable to the attack than others. This could be due to a more diverse data set for certain classes (e.g. the data set has more outliers), which could also lead to a larger difference between member and non-member data points.

We could also see that the analysed time series of incorrectly classified confidence values became increasingly dissimilar towards the end. This is consistent with the increase in the train-test accuracy gap (i.e. higher overfitting). This supports the results of previous research (e.g. [18, 19, 21, 23]) where a higher overfitting was associated with a higher difference between the outputs of member and non-member data points. Nevertheless, a high train-test accuracy gap is not sufficient to explain what causes dissimilarity between member and non-member scores. As it has been found, there are also clear dissimilarities in CIFAR-10, although the train-test accuracy gap is significantly smaller than in CIFAR-100.

Finally, for experiment I, it can be concluded that the hypothesis H0 made in this thesis could be partially falsified and the hypothesis H1 could be confirmed since

6. Conclusion

for incorrectly classified confidence values a different behavior between membership and non-membership data points could be observed over time. This indicates that checkpoints could be used as additional information to better evaluate the privacy risk of an attack.

6.2. Experiment II

For experiment II, it was found that in the context of the CIFAR-10 data set, an effective attack could only be trained for single true classes using the first 120 epochs. Whereas for CIFAR-100, it was shown that with a max. epoch of 200 and without subdividing per true class, the best results were archived. Using all 200 training epochs for both data sets and using 120 epochs for the CIFAR-100 data set, there was not enough data to produce meaningful results when separating the attack by true class. This would likely have increased the performance of the attack. Comparing the attack results for max. 200 and max. 100 without subdividing per true class, it can be concluded that the CIFAR-100 data set was more vulnerable to the attack than the CIFAR-10 data set. This supports results from previous research (e.g. [2, 15, 18, 19]) where it was found that CIFAR-100 is generally easier to attack than CIFAR-10. Furthermore, for both data sets, we found, that some classes were more vulnerable to the attack than others.

Even though LiRA performs better than our attack, the presented results can compete with other previous MI attacks. Moreover, it is noticeable that LiRA was performed with a more poorly trained model (i.e. in terms of the train-test accuracy gap). Therefore, our attack could perform even better with the same model used by Carlini et al. for LiRA.

In general, it has been shown that at a higher train-test accuracy gap (i.e. at max. epoch 200), fewer erroneous examples were produced and thus fewer data points were available to train an attack. In the context of MIA, a higher train-test accuracy gap is often problematic, as it is an indicator of a higher dissimilarity between member and non-member data points, and thus a more vulnerable target model. Nevertheless, in this work, we have shown that a higher train-test accuracy gap (i.e. coupled with a higher train accuracy) also leads to a smaller attack surface. For the CIFAR-10 data set, this even led to an attack that could not be effectively trained or results that could not be considered meaningful. In contrast, for a smaller train-test accuracy gap (i.e. at max. epoch 120), the target model is often less vulnerable to attack because the data points from members and non-members are more similar. This was also shown in Experiment I: later checkpoints produced more similar results than earlier checkpoints. However, this work has also demonstrated that a smaller gap between training and test accuracy can lead to a larger attack surface (i.e. more incorrectly

classified samples), allowing for an effective attack, as seen in CIFAR-10. This can be described as a trade-off between training a target model with a higher train-test accuracy gap (i.e. coupled with a higher train accuracy), which may result in higher dissimilarity between members and non-members but a smaller attack surface, or opting for more general training, which may result in lower dissimilarity between members and non-members but a larger attack surface.

Finally, we can conclude for experiment II that the hypothesis H0 stated in this thesis could again be partially disproved and the hypothesis H1 could be confirmed. It was found that adding checkpoints to an membership inference attack can lead to effective results in certain scenarios and can thus be used to better assess the privacy risk of a target model.

7. Discussion and future work

Experiment I demonstrated that the analysis performed can be used to effectively assess the similarity of member and non-member data points over time. In practice, this could be used to better assess the risk of an MI attack on the privacy of the target model, not only in general but also with respect to a specific training time point (i.e. assessing which checkpoints of the model are more prone to vulnerabilities). Whether higher dissimilarity between member and non-member data points always implies higher privacy risk and vice versa remains an open question for future work.

The proposed attack in Experiment II is simple but still achieves good results compared to other attacks (i.e. except LiRA). However, this only refers to incorrectly classified points. It remains open whether the points not taken into account can also be made vulnerable using checkpoints using other methods. Furthermore, it remains to be critically considered how meaningful the results are when the attacks are based on a very limited number of data.

Moreover, in this work we have chosen the perspective of the model owner to conduct the analysis. The potential attacks carried out thus had full access to the target model's information. To be able to argue from the perspective of a realistic attacker, shadow models could be trained in future work. It must be taken into account that in most real scenarios there is no access to the checkpoints of the target model. The attack presented in this work would therefore not be applicable. It remains open, if checkpoints of the shadow models could then still be used to gain better result. Nonetheless, this work is primarily to be understood as a contribution to a better assessment of the data security of a target model from the model owner's perspective.

Furthermore, in future work, more data sets could be evaluated using the methodology presented in this thesis. In addition, only one metric (i.e. Pearson correlation coefficient) was applied to analyse the checkpoints. In subsequent work, other metrics could be employed. It could also be investigated whether the use of a limited number of checkpoints could lead to better results. For example, only the later checkpoints could be used, as this work has shown that the confidence values of members and non-members of later checkpoints show the greatest differences. It is also conceivable that the use of checkpoints presented here could be used in combination with other MI attacks to improve their performance. For example, in the context of LiRA, the

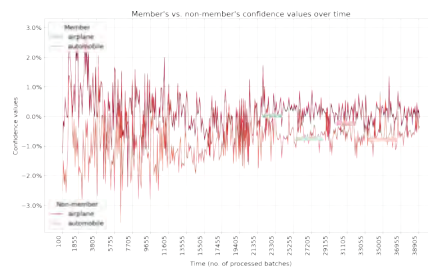
7. Discussion and future work

checkpoints could be used as additional shadow models.

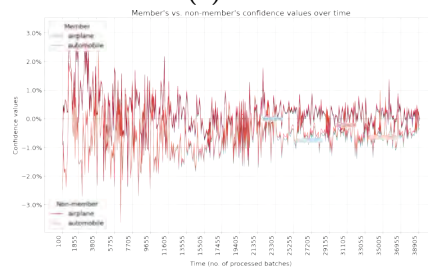
Finally, this work does not allow a general conclusion on whether using checkpoints in the context of MIA gives better results than not using them. While the attack presented in this work is better than some other attacks, a direct comparison is missing. LiRA could be used for this purpose. In this way, one could compare whether LiRA is improved by the use of checkpoints or not.

A. Appendix

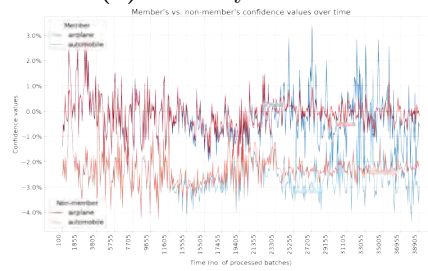
A.1. Experiment I



(a) All



(b) Correctly classified

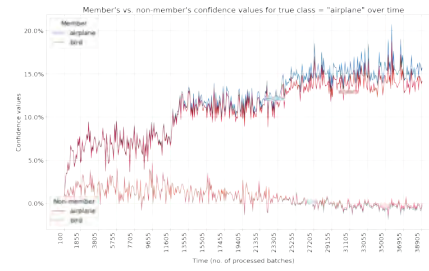


(c) Incorrectly classified

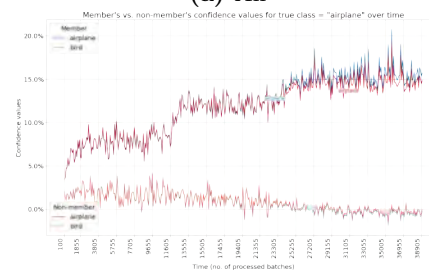
Figure A.1.: CIFAR-10: Member and non-member time series of aggregated confidence values for confidence classes "airplane" and "automobile"

A.2. Experiment II

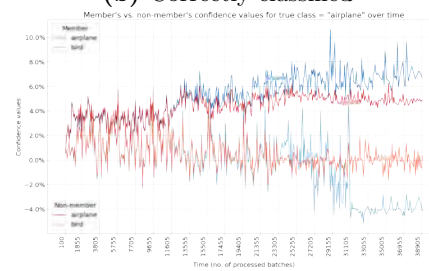
A. Appendix



(a) All

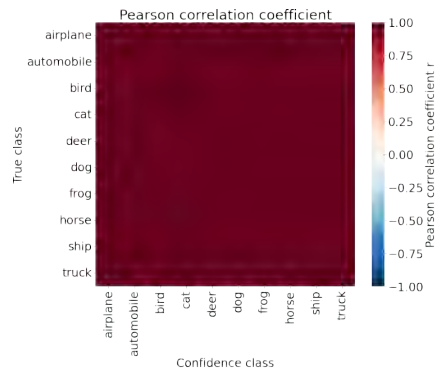


(b) Correctly classified

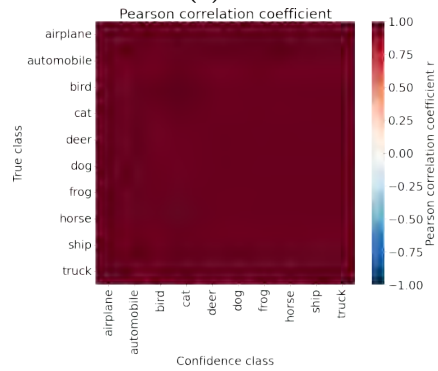


(c) Incorrectly classified

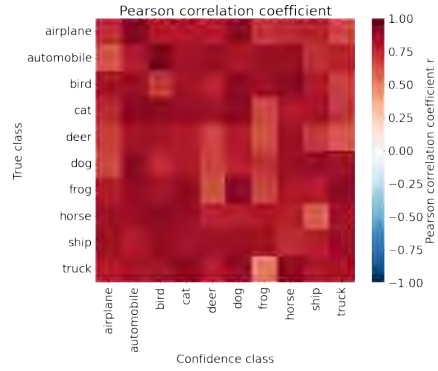
Figure A.2.: CIFAR-10: Member and non-member time series of aggregated classified confidence values for true class "airplane" and confidence classes "airplane" and "automobile"



(a) All



(b) Correctly classified



(c) Incorrectly classified

Figure A.3.: CIFAR-10: Pearson correlation coefficients per true and confidence class of member vs. non-member time series of aggregated confidence values

A. Appendix

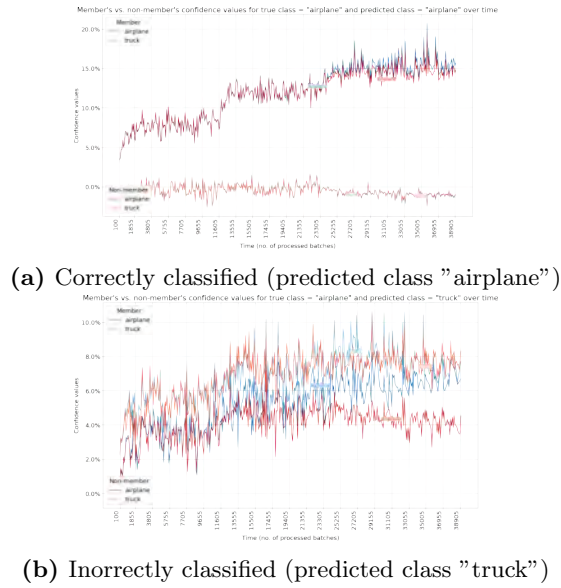


Figure A.4.: CIFAR-10: Member and non-member time series of aggregated confidence values for true class "airplane" and confidence classes "airplane" and "truck"

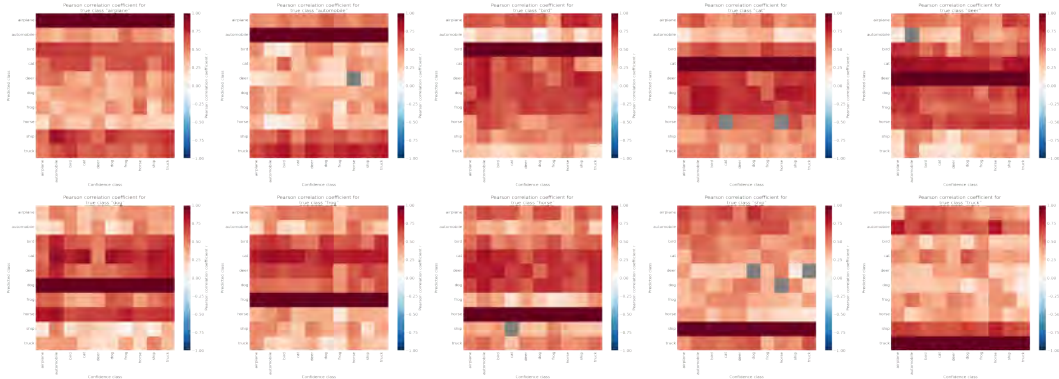
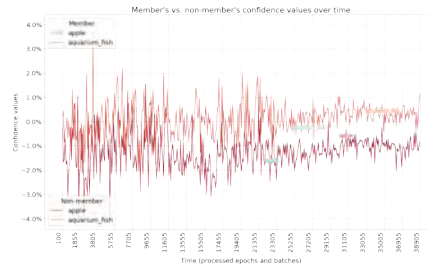
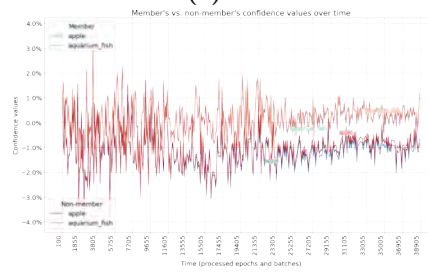


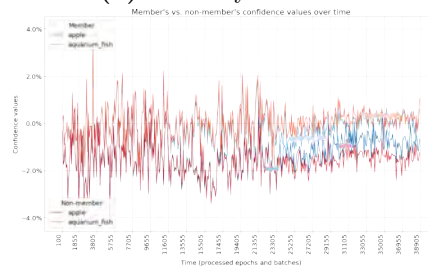
Figure A.5.: CIFAR-10: Pearson correlation coefficients per true, predicted and confidence class of member vs. non-member time series of aggregated confidence values



(a) All



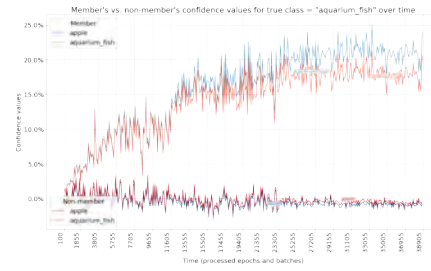
(b) Correctly classified



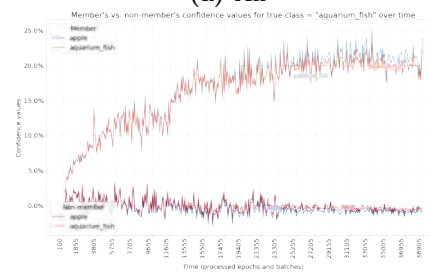
(c) Incorrectly classified

Figure A.6.: CIFAR-100: Member and non-member time series of aggregated confidence values for confidence classes "apple" and "aquarium fish"

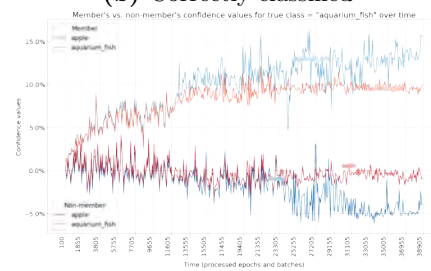
A. Appendix



(a) All



(b) Correctly classified



(c) Incorrectly classified

Figure A.7.: CIFAR-100: Member and non-member time series of aggregated classified confidence values for true class "aquarium_fish" and confidence classes "apple" and "aquarium_fish"

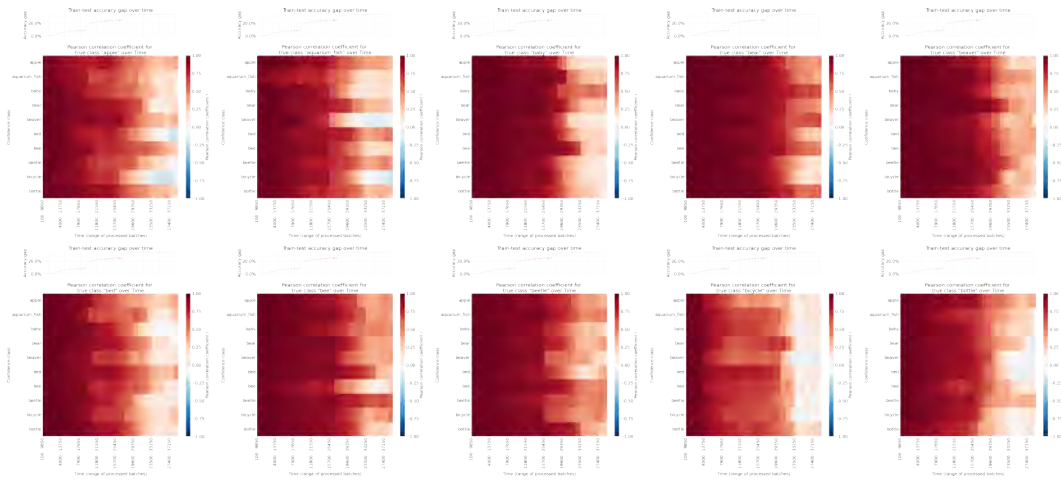
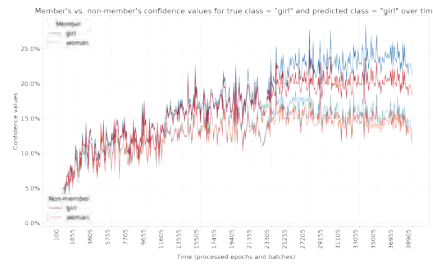
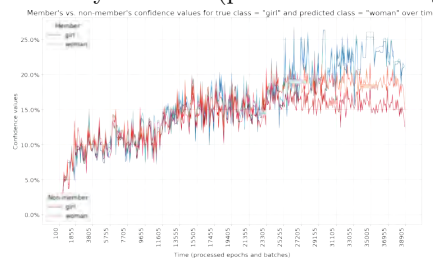


Figure A.8: CIFAR-100: Windowed Pearson correlation coefficients per true and confidence class of member vs. non-member time series of aggregated, incorrectly classified confidence values with train-test accuracy gap over time for confidence and true classes "apple", "aquarium_fish", "baby", "bear", "beaver", "bed", "bee", "beetle", "bicycle", "bottle"



(a) Correctly classified (predicted class = "girl")



(b) Incorrectly classified (predicted class = "woman")

Figure A.9: CIFAR-100: Member and non-member time series of aggregated confidence values for true class "girl" and confidence classes "girl" and "woman"

A. Appendix

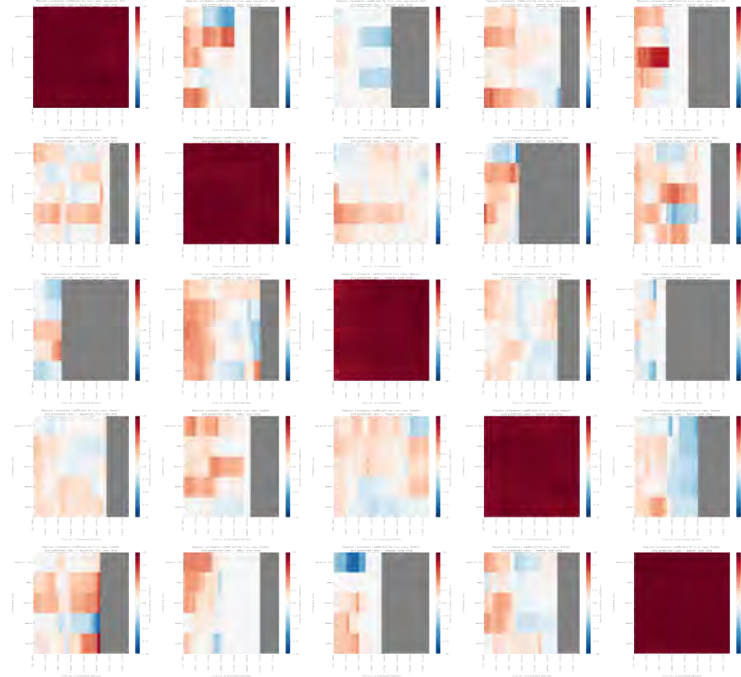


Figure A.10.: CIFAR-100: Windowed Pearson correlation coefficients per true, predicted and confidence class of member vs. non-member time series of aggregated confidence values for confidence, true and predicted classes "apple", "aquarium_fish", "baby", "bear", "beaver", "bed", "bee", "beetle", "bicycle", "bottle"

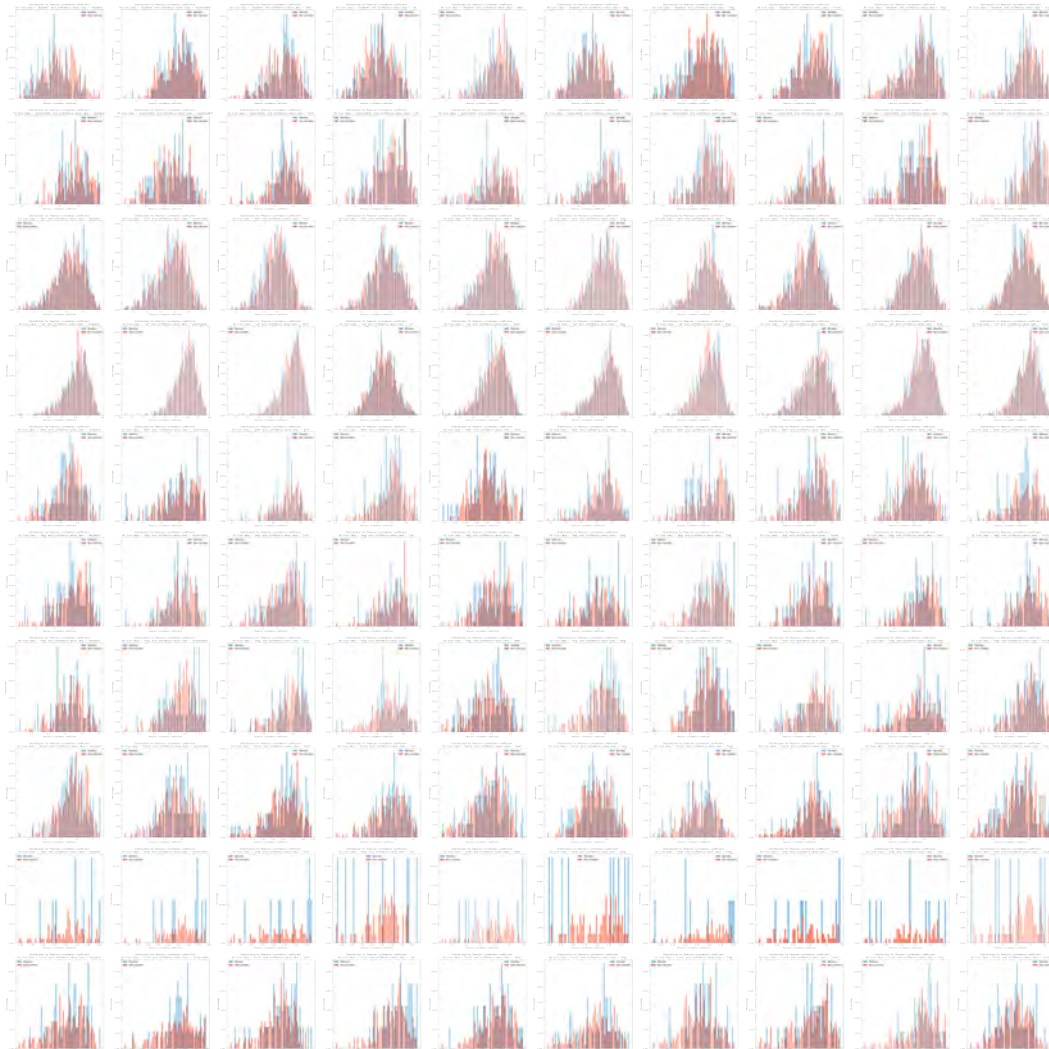


Figure A.11.: CIFAR-10: Histogram of Pearson correlation coefficients per confidence class and true class of member and non-member time series of aggregated, incorrectly classified confidence values with max. epoch 120

A. Appendix

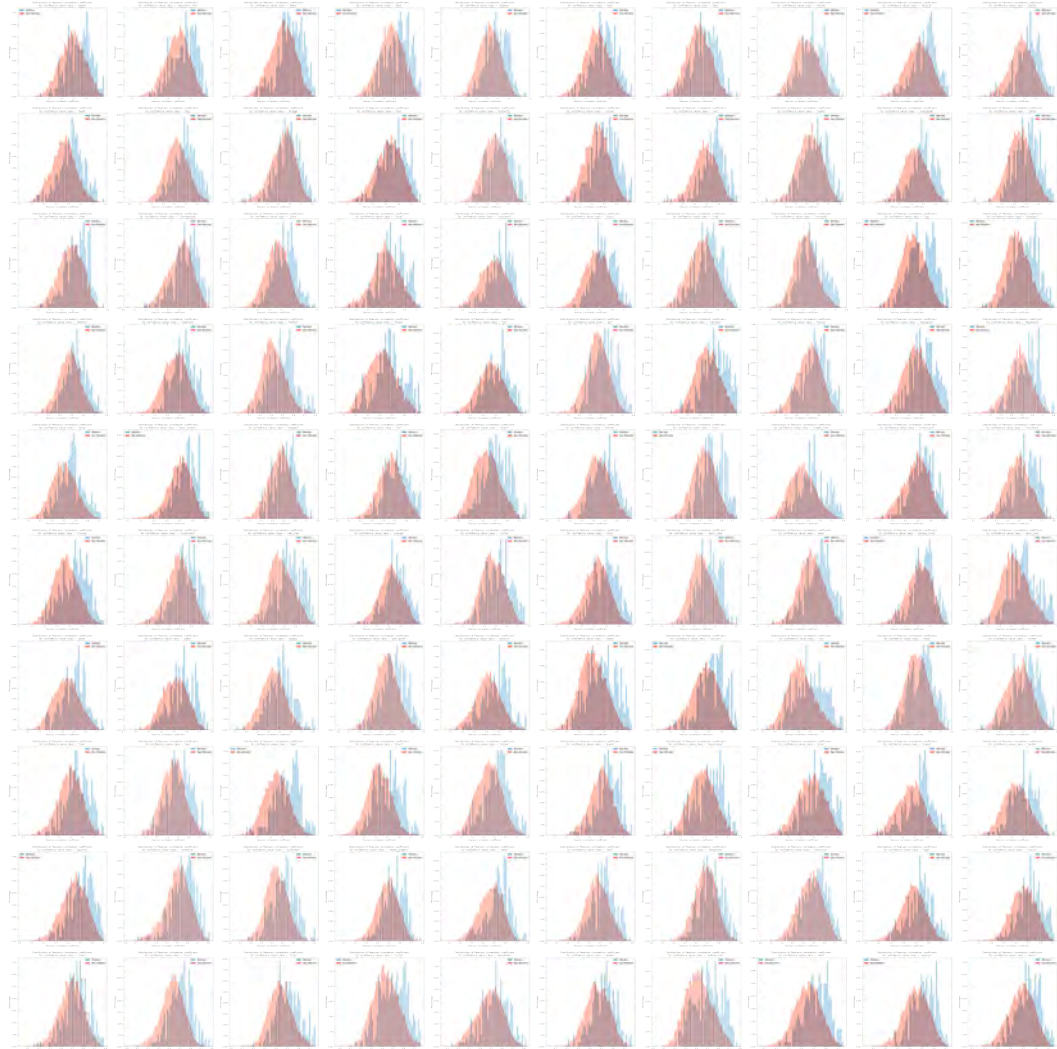


Figure A.12.: CIFAR-100: Histogram of Pearson correlation coefficients per confidence class of member and non-member time series of aggregated, incorrectly classified confidence values

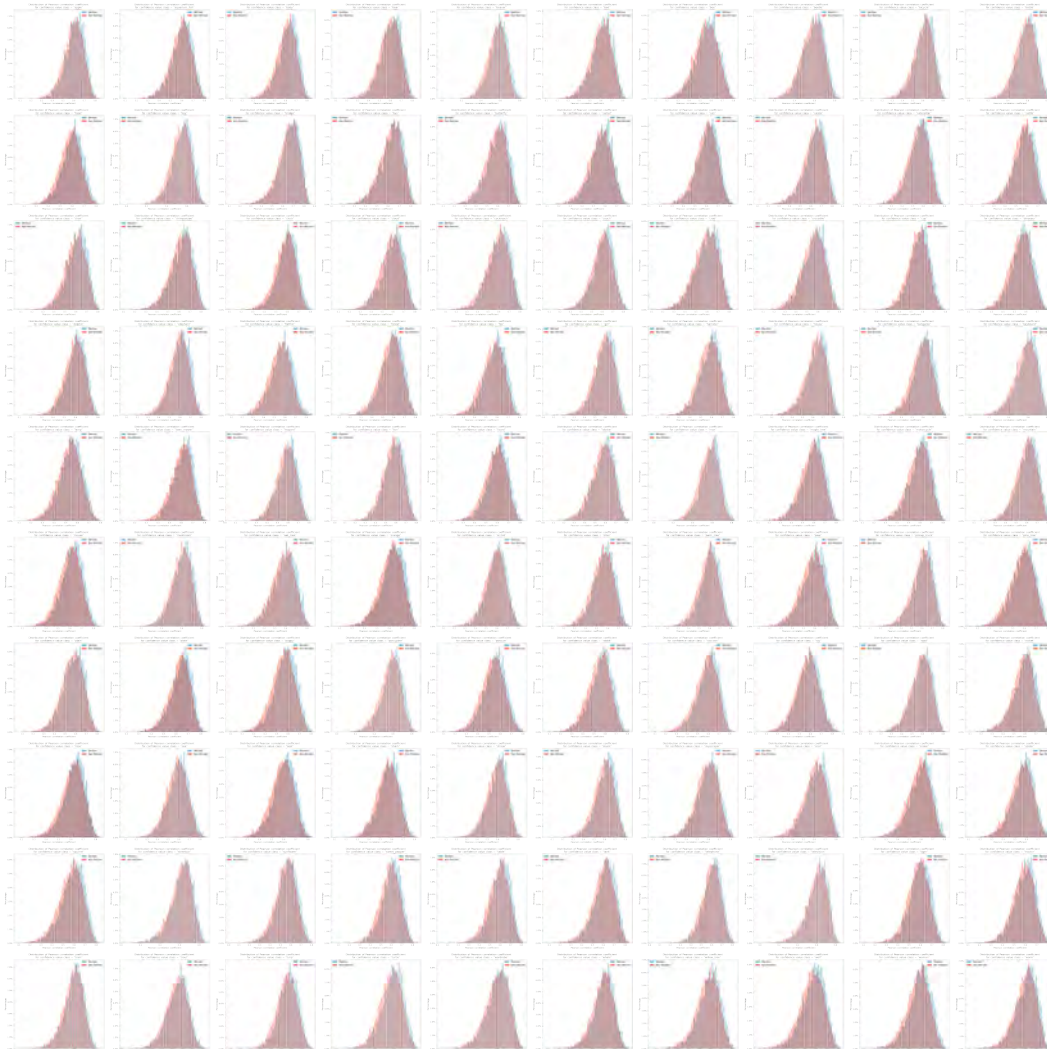


Figure A.13.: CIFAR-100: Histogram of Pearson correlation coefficients per confidence class member and non-member time series of aggregated, incorrectly classified confidence values with max. epoch 120

A. Appendix

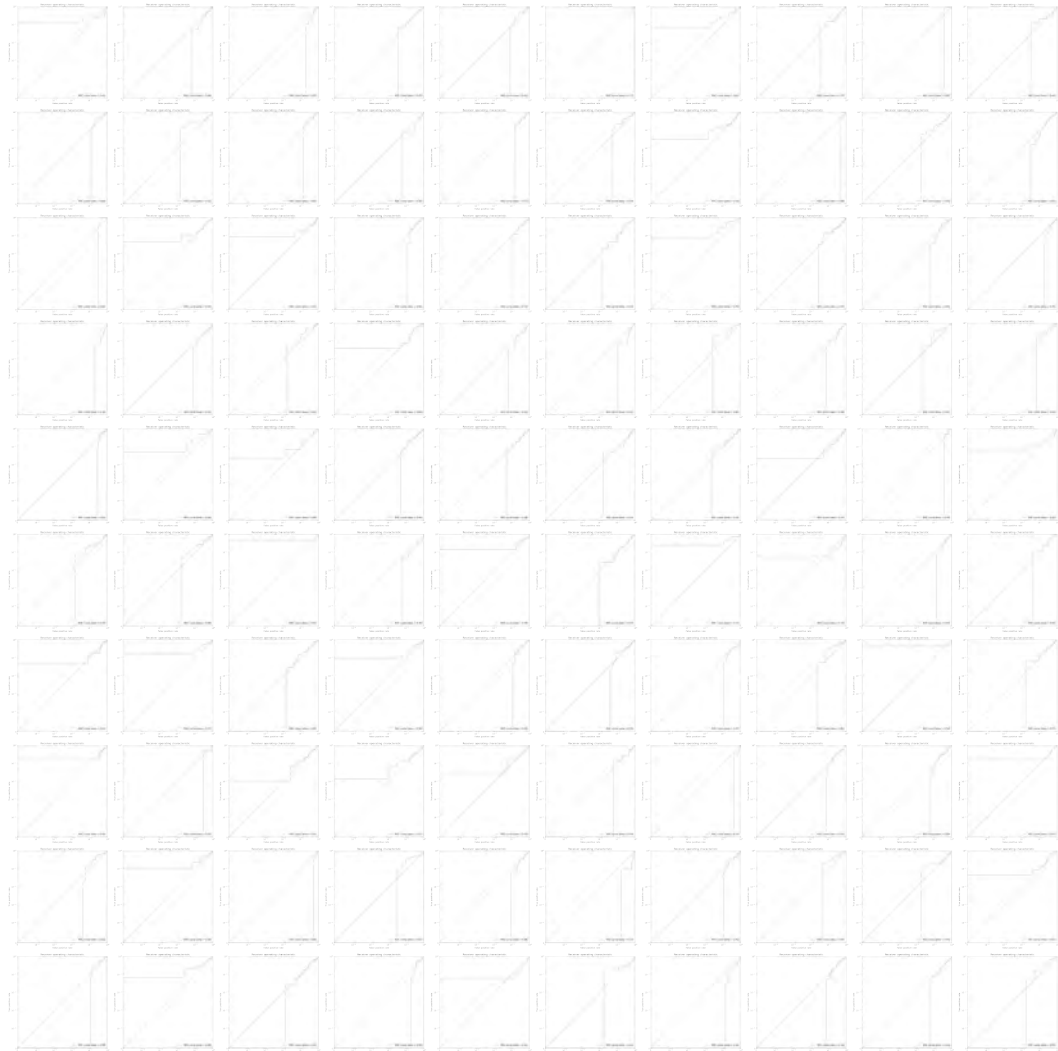


Figure A.14.: CIFAR-100: ROC curve of membership inference attack based on Pearson correlation coefficients per true class of member and non-member time series of aggregated, incorrectly classified confidence values with max. epoch 120

Bibliography

- [1] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. “A Training Algorithm for Optimal Margin Classifiers”. In: COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, pp. 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401. URL: <https://doi.org/10.1145/130385.130401>.
- [2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. “Membership Inference Attacks From First Principles”. en. In: *arXiv:2112.03570 [cs]* (Dec. 2021). arXiv: 2112.03570. URL: <http://arxiv.org/abs/2112.03570> (visited on 03/22/2022).
- [3] C. Cortes and V. Vapnik. “Support-vector networks”. In: *Mach Learn* 20 (1995), pp. 273–297. DOI: 10.1007/BF00994018.
- [4] David A. Dickey and Wayne A. Fuller. “Distribution of the Estimators for Autoregressive Time Series With a Unit Root”. In: *Journal of the American Statistical Association* 74.366 (1979), pp. 427–431. ISSN: 01621459. URL: <http://www.jstor.org/stable/2286348> (visited on 12/02/2022).
- [5] W. H. Greene. *Econometric Analysis*. English. 5th. New Jersey: Prentice Hall, 2002.
- [6] Robin J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. English. 2nd. Australia: OTexts, 2018.
- [7] Werner Kinnebrock. *Neuronale Netze : Grundlagen, Anwendungen, Beispiele / Werner Kinnebrock*. ger. 2., verbesserte Auflage. Reprint 2018. Berlin ; Oldenbourg Wissenschaftsverlag, 2018. ISBN: 9783486786361.
- [8] “Pearson’s Correlation Coefficient”. In: *Encyclopedia of Public Health*. Ed. by Wilhelm Kirch. Dordrecht: Springer Netherlands, 2008, pp. 1090–1091. ISBN: 978-1-4020-5614-7. DOI: 10.1007/978-1-4020-5614-7_2569. URL: https://doi.org/10.1007/978-1-4020-5614-7_2569.
- [9] Alex Krizhevsky. “Learning Multiple Layers of Features from Tiny Images”. en. In: (), p. 60.
- [10] Yunhui Long, Lei Wang, Diyu Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. “A Pragmatic Approach to Membership Inferences on Machine Learning Models”. In: *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2020, pp. 521–534. DOI: 10.1109/EuroSP48549.2020.00040.

Bibliography

- [11] *Original Wide Residual Networks Implementation*. <https://github.com/szagoruyko/wide-residual-networks>. Accessed: 2022-12-02.
- [12] Karl Pearson. “Note on Regression and Inheritance in the case of Two Parents”. In: *Proceedings of the Royal Society of London*. Taylor & Francis, June 1895, pp. 240–242.
- [13] Jens K. Perret. “Wann verwendet man was?” In: *Arbeitsbuch zur Statistik für Wirtschafts- und Sozialwissenschaftler : Theorie, Aufgaben und Lösungen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 631–634. ISBN: 978-3-658-26148-1. DOI: 10.1007/978-3-658-26148-1_12. URL: https://doi.org/10.1007/978-3-658-26148-1_12.
- [14] *Pytorch Implementation of Sergey Zagoruyko’s Wide Residual Networks*. <https://github.com/meliketoy/wide-resnet.pytorch>. Accessed: 2022-12-02.
- [15] Shahbaz Rezaei and Xin Liu. “On the Difficulty of Membership Inference Attacks”. en. In: *arXiv:2005.13702 [cs, stat]* (Mar. 2021). arXiv: 2005.13702. URL: <http://arxiv.org/abs/2005.13702> (visited on 03/22/2022).
- [16] Alexandre Sablayrolles, Matthijs Douze, Yann Ollivier, Cordelia Schmid, and Hervé Jégou. *White-box vs Black-box: Bayes Optimal Strategies for Membership Inference*. 2019. DOI: 10.48550/ARXIV.1908.11229. URL: <https://arxiv.org/abs/1908.11229>.
- [17] Said E. Said and David A. Dickey. “Testing for unit roots in autoregressive-moving average models of unknown order”. In: *Biometrika* 71.3 (Dec. 1984), pp. 599–607. ISSN: 0006-3444. DOI: 10.1093/biomet/71.3.599. eprint: <https://academic.oup.com/biomet/article-pdf/71/3/599/719376/71-3-599.pdf>. URL: <https://doi.org/10.1093/biomet/71.3.599>.
- [18] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. “ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models”. In: Jan. 2019. DOI: 10.14722/ndss.2019.23119.
- [19] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. “Membership Inference Attacks against Machine Learning Models”. en. In: *arXiv:1610.05820 [cs, stat]* (Mar. 2017). arXiv: 1610.05820. URL: <http://arxiv.org/abs/1610.05820> (visited on 03/22/2022).
- [20] Robert Shumway and David Stoffer. *Time Series and Its Applications*. Jan. 2011. ISBN: 978-1-4757-3263-4. DOI: 10.1007/978-1-4757-3261-0.
- [21] Liwei Song, Reza Shokri, and Prateek Mittal. “Privacy Risks of Securing Machine Learning Models against Adversarial Examples”. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Nov. 2019. DOI: 10.1145/3319535.3354211. URL: <https://doi.org/10.1145/3319535.3354211>.

- [22] Lauren Watson, Chuan Guo, Graham Cormode, and Alex Sablayrolles. *On the Importance of Difficulty Calibration in Membership Inference Attacks*. 2021. DOI: 10.48550/ARXIV.2111.08440. URL: <https://arxiv.org/abs/2111.08440>.
- [23] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. “Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting”. en. In: *arXiv:1709.01604 [cs, stat]* (May 2018). arXiv: 1709.01604. URL: <http://arxiv.org/abs/1709.01604> (visited on 03/22/2022).
- [24] Sergey Zagoruyko and Nikos Komodakis. “Wide Residual Networks”. en. In: *arXiv:1605.07146 [cs]* (June 2017). arXiv: 1605.07146. URL: <http://arxiv.org/abs/1605.07146> (visited on 03/29/2022).