2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) | 979-8-3503-6547-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPRW63382.2024.00362

A Comprehensive Analysis of Factors Impacting Membership Inference

Daniel DeAlcala Gonzalo Mancera Aythami Morales Julian Fierrez
Ruben Tolosana Javier Ortega-Garcia
Biometrics and Data Pattern Analytics Lab, Universidad Autonoma de Madrid, Spain
{ daniel.dealcala gonzalo.mancera aythami.morales
julian.fierrez ruben.tolosana javier.ortega } @uam.es

Abstract

We analyze various factors affecting the proper functioning of MIA and MINT, two research lines aimed at detecting data used for training. The difference between these lines lies in the environmental conditions, while the fundamental bases are similar for both. As evident in the literature, this detection task is far from straightforward and poses an ongoing challenge for the scientific community. Specifically, in this work, we conclude that factors such as the number of times data passes through the original network, the loss function, or dropout significantly impact detection outcomes. Therefore, it is crucial to consider them when developing these methods and during the training of any neural network, both to avoid (MIA) and to enhance (MINT) this detection. We evaluate the AdaFace facial recognition model using five databases with over 22 million images, modifying the different factors under analysis and defining a suitable protocol for their examination. State-of-the-art accuracy reaching up to 87% is achieved, surpassing existing methods.

1. Introduction

The detection of data used to train a model is an open field with various applications. The traditional application, known as Membership Inference Attacks (MIA) [31], involves an attacker attempting to obtain confidential information about the data used for training, such as medical information about patients used to train a model. The purpose of MIA is malicious, aiming to steal information. The literature aims to demonstrate that this is possible and proposes solutions to prevent such detection. A new perspective called Membership Inference Test (MINT) [4] seeks to perform this detection on models with the goal of identifying the unauthorized use of data without users' consent in line with new international regulations such as the EU AI act[1]. MINT audits trained AI/ML models to provide transparency and explainability to users [1, 33], by allowing people to know if given data (e.g., their private [9] or some other sensitive data [23]), was used in the development of those models. Both MIA and MINT work on the same task (membership inference) but under different environmental conditions. MIA studies attacks on models, and thus, access to the original model is not be assumed, although access to information to train similar models is possible. In contrast, MINT assumes access to the original model trained by the developer since it is not an attack but a tool for auditing.

These two research lines pose a challenge to the scientific community, not being trivial at all [28]. Despite the distinct purposes of MIA and MINT, it is crucial for both lines to understand the factors and parameters influencing their outcomes. This understanding is essential, whether to prevent considered in MIA or to enhance the membership analysis considered in MINT.

In [4], the authors proposed a MINT model elevating the detection of training data usage up to 90%. This prompts us to explore the factors influencing the performance of that result. In the present study, we utilize the experimental protocol proposed in [4] and investigate factors that may alter MINT detection in facial recognition models. The main contributions can be summarized as follows:

- We analyze in a comprehensive way the performance of MINT models applied to face recognition systems and propose novel architectures to identify data used during the training process of these systems.
- We identify various factors influencing the detection of data used for training and provide a comprehensive evaluation of these factors in a controlled environment.
- By making specific modifications to the facial recognition model, we achieve a significant improvement (23% relative decrease in membership detection errors) over the results presented in existing approaches without compromising the face recognition model's performance.

[1] https://artificialintelligenceact.eu/

2160-7516/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPRW63382.2024.00362

The is structured as follows: the Sec. 2 provides a summary of realted works. In Sec. 3, we describe our methods and datasets, outlining the factors considered in the experiments. Sec. 4 presents our experiments. Finally, Sec. 5 and Sec. 6 present the discussion and conclusions of our study.

2. Related Works

MIA was presented in 2017 [31] that has spawned numerous related works. MINT on the other hand is a recently introduced concept with no existing literature as of now. In this section, we will introduce the key concepts and relevant works on MIA, along with the foundational work on MINT that serves as the basis for our experiments.

2.1. Membership Inference Attacks (MIA)

Membership Inference Attacks, initially conceptualized in [31], involve attempts to extract information used in the training of a model. This information, encompassing areas such as health and shopping preferences, can be sensitive if disclosed. In MIA, the objective is to obtain this information through adversarial approaches without access to the original model, assuming developers won't willingly provide sensitive information [23]. Shokri *et al*. employed "shadow model" in their approach [31], mimicking the functionality of the original model. Constructing these models relies on having information about the architecture to replicate the original model and a portion of the training database, along with some statistics. In this manner, attackers have complete control over these shadow models and are well-informed about the data used in their training. They forward images through the shadow models, obtain embeddings, and train a binary classifier on whether they were used in training or not.

Following the work of Shokri *et al*., numerous studies have been published. Initially, some works attempted to enhance these results by utilizing metric values and thresholds to differentiate between embeddings of data used in training or not. Yeom *et al*. in [36] sets a threshold in the loss function, while authors in [29, 32] use the prediction value instead of the loss value for this differentiation.

Additionally, Nasr *et al*. [25] endeavored to leverage more information from the model beyond the output embeddings. They introduced the terminology of black-box and white-box, where black-box assumes access only to the output embeddings, as proposed by Shokri *et al*. and continued by [2, 32, 36] among others, and white-box assumes access to activations, losses, and gradients. However, in their work, they did not achieve promising results with white-box access. Activations and losses did not improve black-box results significantly, and the limited improvement in black-box outcomes from gradients, coupled with the challenges in accessing this information by an attacker, rendered it of limited utility. Subsequent studies [3, 28] also demonstrated that the use of activations was not beneficial in this detection.

The challenges in this line of research are highlighted in these mentioned studies and reinforced in [28], where the authors focus on the inherent difficulty of the task. They emphasize that the results that can be expected in the practice are worse than the ones indicated in the literature due to questionable evaluation approaches. In that work [28], the authors once again demonstrated that the white-box scenario did not yield significant benefits over black-box.

Numerous studies have sought to elucidate the mechanisms behind MIA, primarily attributing its operation to model memorization [15, 34], a phenomenon linked to overfitting. Consequently, several works propose counteracting overfitting as a potential solution [12–14, 17, 18]. Tonni *et al*. [34] scrutinize database and model factors pertinent to MIA detection. In their case, databases with vector information and fully connected models are employed. Showcasing the impact of factors such as the quantity of data used to train shadow models or the class distribution on performance. Although this related work bears similarity to ours, the analyzed factors differ significantly, with our emphasis on the original model—within the developer's purview. Moreover, our experiments are conducted in a real and challenging context, specifically facial recognition.

2.2. Membership Inference Test (MINT)

Membership Inference Test (MINT), as presented in [4], serves as an auditing tool designed to detect if given data was used to train a model. This distinctive approach holds significant promise for providing transparency to users and aligns with new European legislation [19], imposing obligations to safeguard citizens' rights. Unlike Membership Inference Attacks (MIA), MINT does not require the training of shadow models, it directly infers information from the original model. The authors in [4] conduct experiments in both white-box and black-box scenarios, revealing that various factors significantly impact the performance of the "MINT Mode" in detecting training data, e.g., the amount of available training data notably influences detection. Moreover, they demonstrate that a white-box approach yields superior results compared to black-box methods, contrary to the results in the MIA literature. Lastly, the paper proposes several new architectures for the MINT Model, optimizing information utilization and enhancing overall performance.

3. Methods and datasets

3.1. Key terms and architecture

MINT is a tool designed to detect the data used in training a neural network. In MINT, we assume the role of an auditor, which involves having access to either: 1) the original

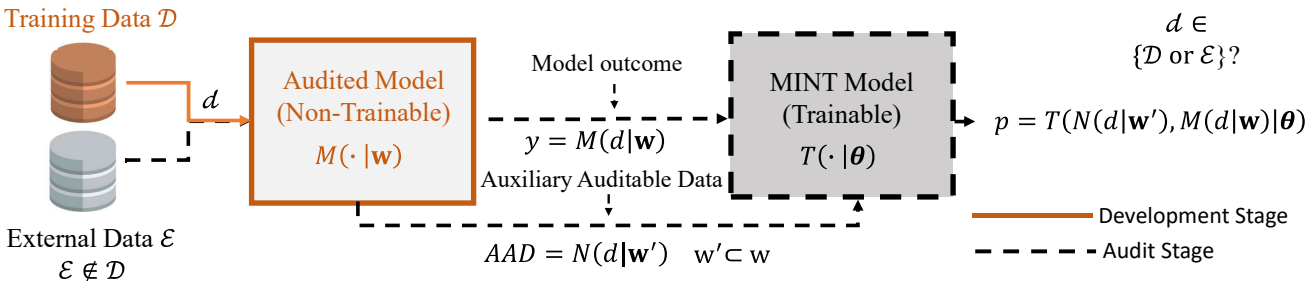

Figure 1. The Membership Inference Test (MINT) Model (T) is trained to predict if a specific data (d) was used during the training process of an Audited AI/ML Model (M), which was previously trained with a database (D). The input of the MINT Model is Auxiliary Auditable Data (e.g., activations maps for data samples d) and/or the model outcome obtained from M [4].

model, 2) a portion of it, or 3) inside information as certain data pass through the network. Therefore, the use of 'shadow models' is not necessary in this technology. In the original work, the authors consider different scenarios for training what they call MINT Models, which are models for detecting data used in training. Specifically, they take into account both black-box and white-box scenarios, as well as the use of varying amounts of data to train these MINT Models. All of this is done, without loss of generality, in a realistic setting based on Face Recognition (FR). Note that the same methods and methodology can be applied to any other AI/ML problem involving learned models. Next, we mention the key terms in MINT approaches (see Fig. 1):

- Audited Model (M): This is the Facial Recognition (FR) model being audited. The model is defined by its architecture and the set of parameters $\mathbf{w}$. The model has been trained by the developer and cannot be modified by the auditor.
- Training Dataset ($\mathcal{D}$): This is the dataset used (by the developer) to train the Audited Model (M).
- External Dataset ($\mathcal{E}$): External data (face images in our case study) not used in the training of the Audited Model. It is crucial that this data is entirely independent of $\mathcal{D}$, ensuring proper training of the MINT Model.
- Collection of Samples (d): These are the samples used to train the MINT Model. They come from both $\mathcal{D}$ and $\mathcal{E}$ ($d \in \mathcal{D} \cup \mathcal{E}$)
- Model Outcome ($y = M(d|\mathbf{w})$): The Model Outcome y is the output of the model M for a sample d. This is the final output of the model and, therefore, is the information available in a black-box context.
- Auxiliary Auditable Data ($AAD = N(d|\mathbf{w}')$): AAD consists of intermediate outputs from the model M for a sample d. N represents the model formed by a part of M corresponding to $\mathbf{w}'$, which is a subset of $\mathbf{w}$. $N = M$ in the case where $\mathbf{w}' = \mathbf{w}$. This intermediate information is available in a white-box scenario.
- MINT Model (T): This is the model that detects whether specific data was used or not in training the Audited Model M. This model is defined by an architecture and parameters θ that are adjusted by the auditors. It takes as input the model outcome y and/or the Auxiliary Auditable Data AAD from samples d in sets $\mathcal{D}$ and $\mathcal{E}$.

3.2. MINT Performance: Impact Factors

The aim of the present paper is to study the main factors that influence the detection performance of the MINT Model T. The authors of the original work [4] present several findings in this regard that merit discussion. In their research, they assume possession of an Audited Model M, trained by the developer, from which they can extract information for MINT detection purposes. By acquiring insights from the model, they proceed to train the MINT Model T. The outcome varies depending on the nature of the information used from the Audited Model. Three pivotal factors influencing the outcomes can be identified:

- Number of data. The quantity of data available for training the MINT Model significantly influences the outcome; more data leads to better results. However, the availability of data is constrained by the information provided by the developer to the auditors.
- Available environment. Results exhibit substantial variations between a black-box and a white-box environment. White-box depends heavily on the depth at which Auxiliary Auditable Data (AAD) is obtained.
- Architecture type. Lastly, the results are tied to the architecture of the MINT Model. Certain architectures can more effectively leverage information, yielding superior outcomes.

In their work, DeAlcala *et al.* assumed the role of an auditor who is provided with a pre-trained model, denoted as the audited Model M, without the ability to modify it, aiming to achieve the best possible results within these constraints. We refer to this perspective as the 'auditor-centric perspective'. Under this perspective, modifications could be made to everything corresponding to the audit stage (black dashed lines in Figure 1), which is the focus of the authors in their work. In contrast to the auditor-centric perspective,

our study aims to explore the actions that developers can undertake to influence this detection process. This involves modifying aspects related to the model trained by the developer, including all components concerning the data utilized for training, their incorporation into the model, and the model itself (development stage depicted in orange in Figure 1). This analysis is presented in Sec. 4

3.3. Datasets and Models

For this work, Facial Recognition (FR) models play a crucial role, and it is essential to have complete control over them. This enables us to modify specific parameters of these models and observe their impact on detection. In this study, we will utilize AdaFace's code [22] to train FR models, adjusting as needed to meet our goals. The original AdaFace model we will use possesses the following characteristics: a ResNet50 network trained on the MS1MV3 database with AdaFace Loss function. Subsequently, we will adjust the parameters, as explained in Sec. 4.

The MINT Model will be based on an MLP architecture, consistently employing the same architecture to ensure accurate comparisons of results when varying parameters of the facial recognition model. The architecture of the MINT Model depends on the data used for training. In this study, we will consider two types of scenarios: a black-box scenario where the available data includes only the model outcome y and a white-box scenario where access is also granted to the Auxiliary Auditable Data AAD. In the case of the basic AdaFace model [22], the Model Outcome is a vector of size $N = 512$. For the AAD, we will adopt an approach similar to the original work [4]: extracting activation maps from four points in the Audited Model (AdaFace in our case study) and selecting the maximum value from each map, thus obtaining a vector for each selected model point. We can concatenate the four vectors from the four selected points (note that this number of points $P = 4$ where the Audited Model is inspected is in general tunable), resulting in a vector of size $L = 960$. This MINT Model is called the Vanilla MINT Model, and the details are elaborated in Figure 2.

The Vanilla MINT Model is based on two fully connected layers with ReLU activation, followed by a batch normalization layer and a dropout layer at the end with a rate of 0.3. It is trained for 20 epochs using the Adam optimizer with a learning rate of 0.001, and the remaining parameters follow the default optimizer settings.

Regarding the databases, we have the MS1Mv3 [11] database, consisting of 5.2 million images from 91K identities. This will serve as the Training Dataset D. As for External Datasets $\mathcal{E}$, we include IJB_C [20] (3.5K identities and 31K face images), FDDB [16] (5.2K face images), GANDiffFace [21] (10K identities and 500K face images), and Adience [8] (2.2K identities and 26.5K face images).

4. Experiments

As mentioned earlier, we will modify the facial recognition model (development stage Fig. 1) to compare the results for key parameters in the developer's hands. In each subsection, we will detail the changes made to the original audited model. Changes to the audited model may result in a significant loss of performance, which is essential to note. In each subsection, we analyze the performance differences between the Audited Model M and the MINT Model T compared to the original performance.

4.1. Evaluation Protocols and Metrics

The evaluation of the Audited Models aligns with the methodology outlined in the original AdaFace GitHub [22]. Specifically, the evaluation is performed on the High-Quality [30] Image Validation Set, yielding verification accuracy across five distinct databases (LFW, CFPFP, CPLFW, AGEDB). We will present the mean evaluation value across these databases.

The evaluation of the MINT Models is structured as follows. We have five databases, one used for training, representing the positive class, and four others that were not used for training, representing the negative class in MINT Model training. We utilize three of the four negative class databases (IJB_C, GANDiffFace, FDDB) for MINT Model training, reserving the fourth (Adience) for evaluation. The positive class database is divided into 66% for training and 33% for evaluation. For more detailed information on this evaluation, please refer to the original paper [4].

As explained in Sec. 3.3, the results are presented for the Vanilla MINT Model, trained using different information sources. Specifically, this model is trained using data from intermediate layer outputs (as detailed in Fig. 2) as well as the model outcome. The tables in this section include results for 'Conv Layer X', where lower values of X indicate AAD obtained from an activation layer closer to the model's input (only that layer indicated by X). 'All Conv Layers' denotes the concatenation of these four AAD, while 'Model Outcome' refers to the model's output (y in Sec. 3.1). All conv layers exclude the model outcome to properly distinguish between white-box and black-box scenarios. For further details, please refer to Sec. 3.3 and the original paper [4]. The MINT Model is trained using 100K samples, a parameter previously adjusted by the authors in [4]. In this investigation, our aim is to delve into the nuances of factors within the purview of the original model developer (Development Stage in Fig. 1). Hence, we deliberately refrain from altering this specific parameter, focusing our inquiry instead on discerning how manipulations made by the developer impact the model's efficacy. For a deeper understanding of how varying training samples influences MINT Model behavior, we refer readers to the comprehensive analysis provided in the original publication [4].

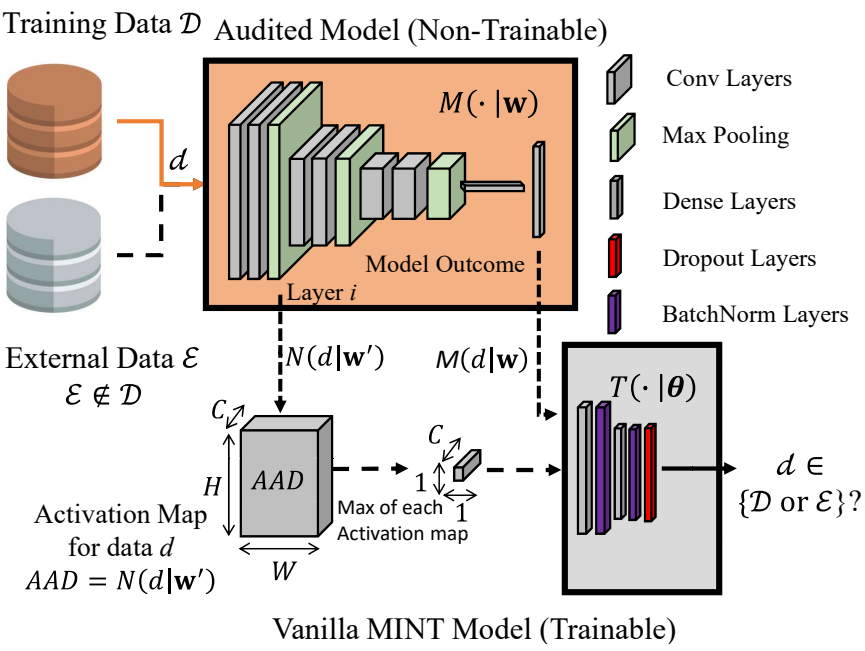

Figure 2. Learning framework of the Vanilla MINT Model trained with the Auxiliary Auditable Data obtained from the Convolutional Layer i [4]. In general we consider P different Layers across the Audited Model where the model is inspected. The input to T is therefore of size $L = C \times P$ ($P = 4$ in our case study).

Auditable Data	Baseline
Conv Layer #1	0.76
Conv Layer #2	0.70
Conv Layer #3	0.65
Conv Layer #4	0.74
Model Outcome	0.81
All Conv Layers	**0.84**

Table 1. Classification accuracy for the various AAD and Model Outcome configurations using the **Baseline** Vanilla MINT Model. MINT model trained with $100K$ samples.

4.2. Baseline Performance

We first display the outcomes for the basic AdaFace FR model. This model consists of a ResNet50 trained using the MS1Mv3 dataset alongside the AdaFace loss function, as described in Sec. 3.3. These outcomes serve as our baseline for subsequent comparisons. The results are tabulated in Tab. 1. The FR Model M verification accuracy in the High-Quality Image Validation Set considered is 96.3%.

4.3. Scenario 1: Augmented Data Frequency

During the model training process, the common practice involves processing all training data once per epoch. While alternative approaches exist, such as assigning probabilities to data to determine their processing frequency, the prevailing and typically most effective method is for all data to undergo a single pass per epoch [10]. This approach is also employed in AdaFace training, meaning that if the AdaFace model is trained for 20 epochs, each image is presented to the model 20 times. This raises the question: Will increasing the number of exposures of images during training result in more effective memorization of the data and consequently impact the MINT detection capability?

To address this question, we introduce the procedure outlined in Fig. 3. On the left, we depict the baseline scenario, where each data point is encountered once by the model in each epoch. On the right, we illustrate the scenario with increased frequency of image exposure, which we term the Augmented Data Frequency scenario. Training has been adjusted such that 25% of the dataset is encountered only once per epoch, mirroring the baseline scenario. Another 25% of the dataset is repeated 10 times per epoch, followed by 20 times for the subsequent 25%, and finally, the last 25% is repeated 30 times per epoch. After training the FR model in this manner, we can analyze the results of the MINT Model. We train a MINT Model T using each of the four sections presented to examine the differences in results among them. The detection results for each MINT Model are presented in Tab. 2.

The performance of the Augmented Data Frequency FR Model M is 98.2 %. Compared to the baseline model, we have increased performance by nearly 2%, indicating that

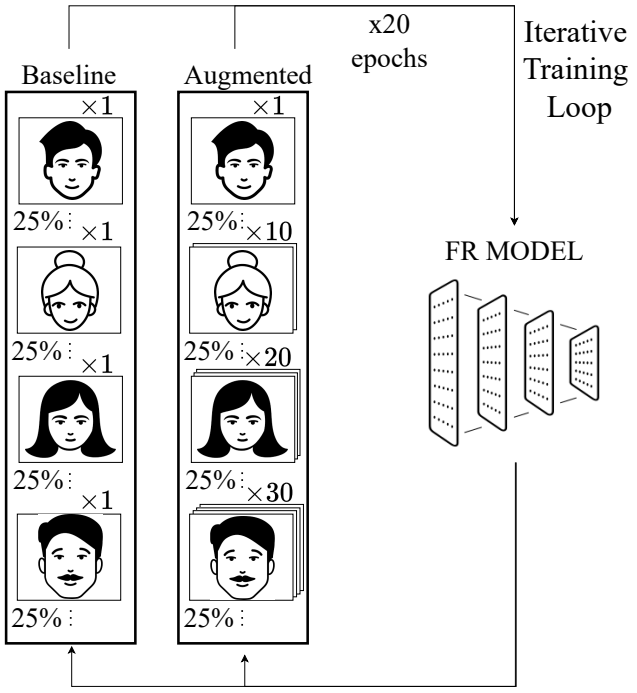

Figure 3. On the left, we have the baseline training method where all images pass through the model once in each epoch. On the right is our approach with Augmented Frequency Data where 25% of the images pass once in every epoch, another 25% pass 10 times, the next 25% pass 20 times, and the last 25% pass 30 times.

this modification in training is indeed beneficial for the FR Model. However, the improvement comes at the expense of significantly increased computational resources required for model training.

We can observe the MINT detection results in Tab. 2, where we analyze the outcomes for different Augmented Data Frequency scenarios. Firstly, **we can conclude that the number of times the data is observed when training the Audited Models is a significant factor in MINT detection**. In the last three rows, we can observe that the MINT model performs worse as the data is seen less when training the Audited Model. Conversely, for the layers closer to the input (Conv Layers #1, #2, and #3), the MINT detection accuracy remains stable independent of the training data frequency. Another noteworthy aspect is evident when comparing Tab. 1 and Tab. 2. Let's focus on the "All Conv Layers" row, as it yields the best results. The baseline model's value is 0.84 (see Tab. 1), whereas in the augmented frequency model, there are values both above (for data augmented $\times 20$ and $\times 30$ times) and below (for data augmented $\times 10$ and $\times 1$). This underscores an important conclusion: **developers have the power to dictate which data the MINT Model detect better or worse by controlling the training data frequency without compromising the overall performance of their model. Consequently, they could conceal illicitly the usage of certain training data through appropriate training procedures, at least to some extent.**

Auditable Data	$\times 1$	$\times 10$	$\times 20$	$\times 30$
Conv Layer #1	0.72	0.74	0.73	0.72
Conv Layer #2	0.66	0.67	0.67	0.66
Conv Layer #3	0.68	0.67	0.69	0.68
Conv Layer #4	0.64	0.73	0.82	0.84
Model Outcome	0.75	0.76	0.80	0.85
All Conv Layers	**0.76**	**0.81**	**0.85**	**0.87**

Table 2. Classification accuracy for the various AAD and Model Outcome configurations using the Vanilla MINT Model for the **Augmented Data Frequency** scenario. MINT model trained with $100K$ samples.

4.4. Scenario 2: Reduced Batch Size

The batch size can also be an important factor in our setup. Batch training involves processing sets of data together, which leads to more stable training and prevents weight updates from depending on a single sample. Using an appropriate batch size is crucial for achieving proper model training. Changing the batch size has a direct effect on the performance of a model M. The question here is how does this batch size affect the performance of the MINT Model T? The initial hypothesis is that the smaller the batch size, the more influence each sample has on the modification of the model's weights, and therefore it could be more easily memorized by the network.

In our setup, the baseline model referenced in Sec. 4.2 is trained with a batch size of 256. The experiments here reduce this size to observe its impact on detection. The initial goal was to reach a batch size of 1 to examine the effects; however, training with such small batch sizes resulted in the Audited Model M—in this instance, the AdaFace model—not learning effectively, reducing its recognition accuracy almost to random guess. The smallest batch size that achieved a decent Audited Model performance was 8. Tab. 3 displays the results with batch sizes of 8, 32, 64, and 256 (baseline). The performances of the Reduced Batch Size FR Models M are 78.6%, 90.5%, and 94.0%, compared to the baseline model's 96.3%.

In Tab. 3, we can observe the results of the MINT Model T for the various Audited Models M trained with different batch sizes. In this case, it is evident that reducing the batch size does not aid in improving MINT detection T; in fact, with a very small batch size, the results are even worse. Therefore, **reducing the batch size deteriorates the performance of the Audited Model, offering no clear advantages for MINT detection.**

Auditable Data	8 batch	32 batch	64 batch	256 batch
Conv Layer #1	0.73	0.77	0.75	0.76
Conv Layer #2	0.70	0.71	0.70	0.70
Conv Layer #3	0.61	0.65	0.65	0.65
Conv Layer #4	0.69	0.71	0.73	0.74
Model Outcome	0.80	0.82	0.82	0.81
All Conv Layers	**0.79**	**0.83**	**0.84**	**0.84**

Table 3. Classification accuracy for the various AAD and Model Outcome configurations using the Vanilla MINT Model for the **Reduced Batch Size** scenario. MINT model trained with $100K$ samples.

Auditable Data	with Dropout	without Dropout
Conv Layer #1	0.76	0.79
Conv Layer #2	0.70	0.75
Conv Layer #3	0.65	0.65
Conv Layer #4	0.74	0.74
Model Outcome	0.81	0.80
All Conv Layers	**0.84**	**0.87**

Table 4. Classification accuracy for the various AAD and Model Outcome configurations using the Vanilla MINT Model for the **no-Dropout** scenario. MINT model trained with $100K$ samples.

4.5. Scenario 3: No Dropout

Dropout is a technique designed to prevent memorization, and its beneficial use in network training is well-established [10]. However, the goal of MINT is for the network to detect if given data was used in training, thus the use of dropout may be counterproductive. In this section, we study: can the elimination of Dropout improve MINT detection?

We first train the baseline FR model in the same manner as before but removing the dropout layer and compare the results of these two models (with or without Dropout). The performance of the no-Dropout FR Model M is 96.2%, virtually identical to the baseline model (96.3%). Observing the results in Tab. 4, it is evident that **eliminating Dropout is beneficial for MINT detection, increasing the detection accuracy by 3%, from 0.84 to 0.87 (i.e., 23% relative decrease in MINT detection errors), without compromising the performance of the Audited Model.**

4.6. Scenario 4: Different Loss Function

The loss function defines the learning of a model [24], directly influencing the model's memorization capabilities. The question here is: can certain loss functions favor MINT detection over others?

In Tab. 5, we present the experiments for the Different Loss Function scenario. The same model was trained using three distinct loss functions: ArcFace [6], CosFace [35], and AdaFace [22]. **The overarching conclusion is that the loss function is indeed significant in detection, altering the MINT Model's detection outcomes. Upon closer examination, we observe that depending on the loss function, the MINT Model performs better with information extracted from different parts of the network.** For instance, with the ArcFace function, the MINT Model performs worse with information from the initial convolutional layers (Conv Layers #1 and #2) than with the latter ones (Conv Layers #3 and #4). Conversely, with the AdaFace function, the best-performing MINT Model is the one trained with Conv Layer #1, and the least effective is with Conv Layer #3. The CosFace function presents an intermediate case between the two aforementioned functions.

Auditable Data	AdaFace	CosFace	ArcFace
Conv Layer #1	0.76	0.74	0.67
Conv Layer #2	0.70	0.68	0.62
Conv Layer #3	0.65	0.69	0.72
Conv Layer #4	0.74	0.74	0.75
Model Outcome	0.81	0.77	0.78
All Conv Layers	**0.84**	**0.82**	**0.81**

Table 5. Classification accuracy for the various AAD and Model Outcome configurations using the Vanilla MINT Model for the **Different Loss Function** scenario. MINT model trained with $100K$ samples.

5. Discussion

As we have seen, the developer has the ability to positively or negatively influence MINT detection. This has two main applications.

- The first key application of our research is that an auditing or regulatory body (such as the European Union) establishes training guidelines that developers must follow to achieve the best possible MINT detection. In this case, this could involve avoiding the use of dropout, utilizing loss functions that favor this detection the most, and/or ensuring that data passes through the model a minimum or maximum number of times. This would result in greater model transparency and, therefore, user trustworthiness in modern AI/ML systems [7]. Moreover, this can also be subject to minimum conditions regarding the decrease in performance of the developer's model M, meaning that ML strategies to help auditing processes do not result in a performance decrease of inspected models.
- On the other hand, there is the possibility that developers exploit our findings to conceal that certain data was used for training. This option is dangerous, specially in the Augmented Data Frequency scenario, where it has been observed that the developer can reduce the MINT detection capability of certain data used in training while favor-

ing others. The second main application of this work is to make this possibility visible to encourage the scientific community to develop new techniques designed to prevent (or control) the illicit hiding of training data or other training details that should be visible because of legal requirements or other reasons.

6. Conclusions

We have analyzed potential factors in the development stage of AI/ML models (see Fig. 1) that impact MINT detection (i.e., detecting that given data was used or not in the training of AI/ML models). After a general description, then, to better illustrate our methods and methodology, and without loss of generality (our methods and methodology can be applied to any other domain where AI/ML models are trained with data), we have developed and explored a case study in face recognition. Through experiments involving the modification of certain training parameters of the Audited Model M, we have explored their effects on MINT detection, presenting a diverse range of scenarios and results. Our findings suggest that certain parameters, such as dropout elimination, the utilization of specific loss functions, or increasing the number of times data passes through the network, can favor MINT detection, resulting in improvements of up to 23% relative decrease in MINT detection errors. Furthermore, we have demonstrated that developers can manipulate the MINT detection difficulty of certain data by adjusting their frequency of exposure within the Audited Model training process.

Our future work in this line will: 1) improve the MINT detection accuracy with improved learning architectures, 2) exploit general AI models to help in that regard in specific AI/ML domains of our interest such as biometrics [5], 3) introduce multimodal methods [26] to improve the proposed auditing processes, and 4) investigate how certain human feedback [27] can help to curate and improve the proposed methods.

Acknowledgement

This work has been supported by projects BBfor-TAI (PID2021-127641OB-I00 MICINN/FEDER), Cátedra ENIA UAM-VERIDAS en IA Responsable (NextGenerationEU PRTR TSI-100927-2023-2), and Comunidad de Madrid (ELLIS Unit Madrid). The work of D. deAlcala is supported by a FPU Fellowship (FPU21/05785) from the Spanish MIU. A. Morales is supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with Universidad Autónoma de Madrid in the line of Excellence for the University Teaching Staff in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

References

[1] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. 1

[2] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *Proceedings of the International Conference on Machine Learning*, pages 1964–1974. PMLR, 2021. 2

[3] Ana-Maria Cretu, Daniel Jones, Yves-Alexandre de Montjoye, and Shruti Tople. Re-aligning Shadow Models can Improve White-box Membership Inference Attacks. *arXiv preprint arXiv:2306.05093*, 2023. 2

[4] Daniel DeAlcala, Aythami Morales, Gonzalo Mancera, Julian Fierrez, Ruben Tolosana, and Javier Ortega-Garcia. Is my Data in your AI Model? Membership Inference Test with Application to Face Images. *arXiv preprint arXiv:2402.09225*, 2024. 1, 2, 3, 4, 5

[5] Ivan Deandres-Tame, Ruben Tolosana, Ruben Vera-Rodriguez, Aythami Morales, Julian Fierrez, and Javier Ortega-Garcia. How Good Is ChatGPT at Face Biometrics? A First Look Into Recognition, Soft Biometrics, and Explainability. *IEEE Access*, 12:34390–34401, 2024. 8

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 7

[7] Natalia Díaz-Rodríguez, Javier Del Ser, Mark Coeckelbergh, Marcos López de Prado, Enrique Herrera-Viedma, and Francisco Herrera. Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation. *Information Fusion*, 99:101896, 2023. 7

[8] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and Gender Estimation of Unfiltered Faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014. 4

[9] Marta Gomez-Barrero, Javier Galbally, Aythami Morales, and Julian Fierrez. Privacy-preserving comparison of variable-length data with application to biometric template protection. *IEEE Access*, 5:8606–8619, 2017. 1

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016. 5, 7

[11] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In *Proceedings of the European Conference on Computer Vision*, pages 87–102. Springer, 2016. 4

[12] Hongsheng Hu, Zoran Salcic, Gillian Dobbie, Yi Chen, and Xuyun Zhang. EAR: An enhanced adversarial regularization approach against membership inference attacks. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2021. 2

[13] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys*, 54(11s):1–37, 2022.

[14] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Practical blind membership inference attack via differential comparisons. In *Proceedings of the Network and Distributed Systems Security Symposium*, 2021. 2

[15] Paul Irolla and Grégory Châtel. Demystifying the membership inference attack. In *Proceedings of the CMI Conference on Cybersecurity and Privacy*, pages 1–7. IEEE, 2019. 2

[16] Vidit Jain and Erik Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, UMass Amherst technical report, 2010. 4

[17] Yigitcan Kaya and Tudor Dumitras. When does data augmentation help with membership inference attacks? In *Proceedings of the International Conference on Machine Learning*, pages 5345–5355. PMLR, 2021. 2

[18] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership inference attacks and defenses in classification models. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, pages 5–16, 2021. 2

[19] Tambiama Madiega. Artificial Intelligence Act. *European Parliament: European Parliamentary Research Service*, PE 698.792 (Updated June 2023), 2023. 2

[20] Brianna Maze, Jocelyn Adams, James A. Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K. Jain, W. Tyler Niggel, Janet Anderson, Jordan Cheney, and Patrick Grother. IARPA Janus Benchmark - C: Face Dataset and Protocol. In *Proceedings of the International Conference on Biometrics*, pages 158–165, 2018. 4

[21] Pietro Melzi, Ruben Tolosana, Ruben Vera-Rodriguez, Minchul Kim, Christian Rathgeb, Xiaoming Liu, Ivan DeAndres-Tame, Aythami Morales, Julian Fierrez, Javier Ortega-Garcia, et al. FRCSyn-onGoing: Benchmarking and comprehensive evaluation of real and synthetic data to improve face recognition systems. *Information Fusion*, 107: 102322, 2024. 4

[22] Minchul Kim. *AdaFace Code (Available at GitHub)*. https://github.com/mk-minchul/AdaFace, 2023. 4, 7

[23] Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Ruben Tolosana. SensitiveNets: Learning Agnostic Representations with Application to Face Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 43(6): 2158–2164, 2021. 1, 2

[24] Aythami Morales, Julian Fierrez, Alejandro Acien, Ruben Tolosana, and Ignacio Serna. SetMargin Loss applied to Deep Keystroke Biometrics with Circle Packing Interpretation. *Pattern Recognition*, 122:108283, 2022. 7

[25] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 739–753. IEEE, 2019. 2

[26] Alejandro Peña, Ignacio Serna, Aythami Morales, Julian Fierrez, Alfonso Ortega, Ainhoa Herrarte, Manuel Alcantara, and Javier Ortega-Garcia. Human-Centric Multimodal Machine Learning: Recent Advances and Testbed on AI-based Recruitment. *SN Comp. Science*, 4(5):434, 2023. 8

[27] Alejandro Peña, Aythami Morales, Julian Fierrez, Javier Ortega-Garcia, Iñigo Puente, Jorge Cordova, and Gonzalo Cordova. Continuous document layout analysis: Human-in-the-loop AI-based data curation, database, and evaluation in the domain of public affairs. *Information Fusion*, page 102398, 2024. 8

[28] Shahbaz Rezaei and Xin Liu. On the difficulty of membership inference attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7892–7900, 2021. 1, 2

[29] 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 *Proceedings of the Annual Network and Distributed System Security Symposium*, 2018. 2

[30] Torsten Schlett, Christian Rathgeb, Olaf Henniger, Javier Galbally, Julian Fierrez, and Christoph Busch. Face image quality assessment: A literature survey. *ACM Computing Surveys*, 10(54):1–49, 2022. 4

[31] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 3–18, 2017. 1, 2

[32] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *Proceedings of the USENIX Security Symposium*, pages 2615–2632, 2021. 2

[33] Javier Tello, Marina de la Cruz, Tony Ribeiro, Julian Fierrez, Aythami Morales, Ruben Tolosana, Cesar Luis Alonso, and Alfonso Ortega. Symbolic AI (LFIT) for XAI to Handle Biases. In *European Conf. on Artificial Intelligence Workshops (ECAIw)*, 2023. 1

[34] Shakila Mahjabin Tonni, Dinusha Vatsalan, Farhad Farokhi, Dali Kaafar, Zhigang Lu, and Gioacchino Tangari. Data and model dependencies of membership inference attack. *arXiv preprint arXiv:2002.06856*, 2020. 2

[35] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 7

[36] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of the IEEE Computer Security Foundations Symposium*, pages 268–282. IEEE, 2018. 2