Evaluating Query Efficiency and Accuracy of Transfer Learning-based Model Extraction Attack in Federated Learning

Sayyed Farid Ahamed*§, Sandip Roy*‡§, Soumya Banerjee*‡,
Marc Vucovich†, Kevin Choi†, Abdul Rahman†, Alison Hu†, Edward Bowen†, Sachin Shetty*
*Center for Secure & Intelligent Critical Systems, Old Dominion University, Virginia, USA
‡School of Cybersecurity, Old Dominion University, Virginia, USA
{saham001, sroy, s1banerj, sshetty}@odu.edu
†Deloitte & Touche LLP
mdvucovich@gmail.com, {kevchoi, abdulrahman, aehu, edbowen}@deloitte.com

Abstract—**Federated Learning (FL) is a collaborative learning framework designed to protect client data, yet it remains highly vulnerable to Intellectual Property (IP) threats. Model extraction (ME) attack poses a significant risk to Machine-Learning-as-a-Service (MLaaS) platforms, enabling attackers to replicate confidential models by querying Black-Box (without internal insight) APIs. Despite FL's privacy-preserving goals, its distributed nature makes it particularly susceptible to such attacks. This paper examines the vulnerability of the FL-based victim model to two types of model extraction attacks. For various federated clients built under NVFlare platform, we implemented ME attack across two deep-learning architectures and three image datasets. We evaluate the proposed ME attack performance using various metrics, including accuracy, fidelity, and KL divergence. The experiments show that for various FL clients, the accuracy and fidelity of the extraction model are closely related to the size of the attack query set. Additionally, we explore a transfer learning-based approach where pre-trained models serve as the starting point for the extraction process. The results indicate that the accuracy and fidelity of the fine-tuned pre-trained extraction models are notably higher, particularly with smaller query sets, highlighting potential advantages for attackers.**

Index Terms—**Model extraction attack, Federated learning, Machine-Learning-as-a-Service (MLaaS), Transfer learning, Fidelity, Security.**

I. INTRODUCTION

Recently, Federated Learning (FL) has gained popularity as a privacy-focused machine learning (ML) approach that enables multiple clients to collaboratively create a consolidated model while safeguarding their training data [1]. Unlike traditional centralized ML, FL eliminates the need for clients to transfer their raw data to a central server, thus protecting user privacy and security. The process typically involves training local models on client-specific data, sharing model updates among clients, and constructing a unified model accessible to all participants [2]. Since FL avoids data sharing, it effectively addresses privacy and security concerns commonly associated with centralized ML [3], [4].

Although FL is intended to safeguard personal data, recent studies reveal that FL models are at risk of different attacks

§The authors have equal contributions.

that can disclose sensitive information from training datasets [5], [6]. These vulnerabilities include model extraction (ME), reconstruction, membership inference, and model inversion (MI) attacks [7]. An ME attack occurs when an adversary replicates the functionality of a victim model by querying an Application Programming Interface (API) without internal insight, leading to an extracted model that approximates the original model without direct access to its parameters or training data [8]. In reality, an ME attacker amis for an approximate extraction that focuses on constructing an extracted or piracy model that closely resembles the victim model [9], which can either achieve comparable performance to the victim model (measured by accuracy) or exhibit similar behavior (measured by fidelity). Figure 1 illustrates an overview of the ME attack process utilizing the predictive API on a MLaaS platform.
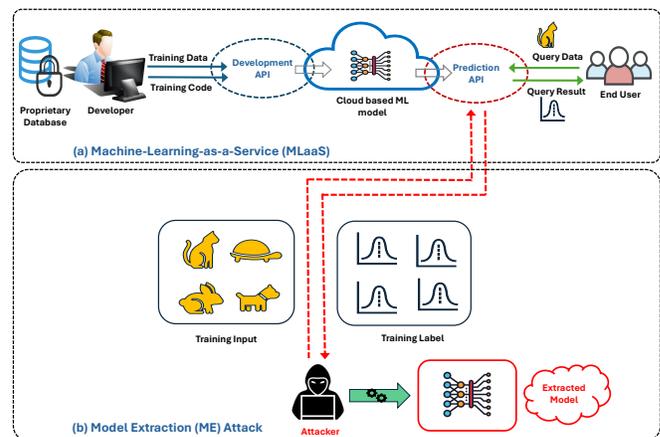

Fig. 1: ME attack executed through predictive API in a Machine Learning as a Service (MLaaS) platform.

As FL models are deployed in MLaaS platforms, attackers can exploit predictive API queries to obtain insights about the victim model, effectively infringing on intellectual property (IP) and undermining the security measures intended to safeguard proprietary algorithms [10]. The collaborative exchange

of model updates in FL can inadvertently expose sensitive information, making FL systems particularly susceptible to ME attacks, which threaten both data privacy and IP [11]. However, there has been only limited exploration of the impact of ME attacks in a scalable FL environment, particularly concerning accuracy and fidelity analysis, leaving this as an open area for further investigation. MLaaS platforms implement strong authentication mechanisms, such as API keys, OAuth, and multi-factor authentication, to restrict model access to authorized users [12]. However, an authenticated adversary can still execute model extraction by issuing an excessive number of queries. In this paper, we investigate three key research questions. **First,** How vulnerable is an FL-based victim model to ME attacks, particularly with the size of the attack query set and the number of FL clients? **Second,** under an FL environment, how do various deep learning architectures (DL) affect the fidelity and accuracy of extracted models? **Finally,** how does the use of transfer learning (TL) with pre-trained models influence the effectiveness of ME attacks compared to models trained from scratch?

The main contributions of this paper are as follows:

- This paper evaluates the vulnerability of FL-based victim models to ME attacks, focusing on how query set size, FL clients, and deep learning architectures impact the fidelity and accuracy of the extracted models.
- We investigate the use of pre-trained models as a starting point for extraction instead of training from scratch. This approach allows us to evaluate the impact of pre-trained models on attack accuracy and query efficiency, revealing potential advantages for attackers.
- We demonstrate that the TL-based ME attack approach enables the extraction model to surpass the best accuracy (training from scratch) with fewer query samples. This method allows the extraction model to closely replicate the victim model's performance, nearly matching its accuracy.

The structure of this paper is as follows: Section II outlines the threat model, detailing the attacker's objectives, knowledge, and capabilities. Section III describes the framework for executing the ME attack within the FL environment. In Section IV, we introduce and explain the proposed algorithm for the ME attack in FL. Section V provides the experimental results along with an analysis and discussion of the findings. Lastly, we conclude our work and discuss a few future research thoughts in Section Section VI.

II. THREAT MODEL

In this section, we outline the threat model, detailing the adversary's knowledge, goals, capabilities, and the scope of the proposed ME attack in FL [13], [14]. Machine learning models deployed in critical infrastructures are increasingly targeted by adversarial threats, including ME attack [15].

Adversary's objective: The adversary $\mathcal{A}$ aims to create an extracted model $\mathcal{M}_e$ that closely replicates the functionality and/or performance of the MLaaS backend model, referred to as the victim model $\mathcal{M}_v$. The similarity between $\mathcal{M}_e$ and

$\mathcal{M}_v$ is evaluated based on either accuracy or fidelity using a test dataset $\mathcal{D}^*$. The attacker does not modify the model's parameters and requires no extra information [8].

Adversary's knowledge: We consider a scenario where $\mathcal{A}$ possesses minimal information about the victim model $\mathcal{M}_v$, such as its architecture, hyperparameters, or the exact dataset used for training. However, the adversary does have access to an unlabeled reference dataset $\mathcal{D}$. While the training goals and model architecture are known to all FL participants, the adversary lacks any insight into the global training process (whether centralized or FL) and the distribution of the training data across clients.

Adversary's capability: $\mathcal{A}$ can interact with $\mathcal{M}_v$ via the MLaaS API, which returns the prediction $\mathcal{M}_v(x)$ for any given input query x. These queries are not limited to real-world data and may also include synthetic or adversarial inputs. The attacker lacks the ability to modify the model's parameters and does not require additional information. Furthermore, as MLaaS typically operates on a pay-per-query basis, we assume the adversary is constrained by a limited query budget n_{query} [16]. $\mathcal{A}$ samples n_{query} inputs query dataset $\mathcal{Q}$ to MLaaS prediction API and use all the query-response pairs $\{x, \mathcal{M}_v(x)\}$ to train the extracted model $\mathcal{M}_e$ by minimizing the cross-entropy loss [8].

III. FRAMEWORK FOR PROPOSED ME ATTACK IN FL

A. Attack Overview

In an ME attack scenario within an FL environment, $\mathcal{A}$ replicates a global victim model $\mathcal{M}_v$ by exploiting its MLaaS API. $\mathcal{A}$ submits API queries without direct access to the model's internal structure or training data. By collecting sufficient input-output pairs, $\mathcal{A}$ trains an extraction or surrogate model $\mathcal{M}_e$ that closely mimics $\mathcal{M}_v$. The attack success of $\mathcal{A}$ is determined by the $\mathcal{M}_e$'s fidelity, or how accurately it replicates the victim model's behavior, and attack accuracy, reflecting how well its predictions align with the original model's outputs. Even with restricted access, a well-executed query strategy can achieve high fidelity in FL and MLaaS setups.

B. Victim Model Built in FL

The victim model $\mathcal{M}_v$ in FL is the global model formed by aggregating updates from multiple participants. In MLaaS environments, this model is accessible through an API, allowing external users to query it for predictions without exposing its internal workings. The attacker leverages this API access to collect numerous input-output pairs. Despite lacking knowledge of the model's architecture or training data, repeated queries enable the adversary to approximate the model's decision boundaries, ultimately building a highly accurate $\mathcal{M}_e$ that replicates the global model's predictive performance.

C. Extracted (Surrogate) Model

The extracted (surrogate) model $\mathcal{M}_e$ is the attacker's replication of $\mathcal{M}_v$, trained on input-output pairs gathered from
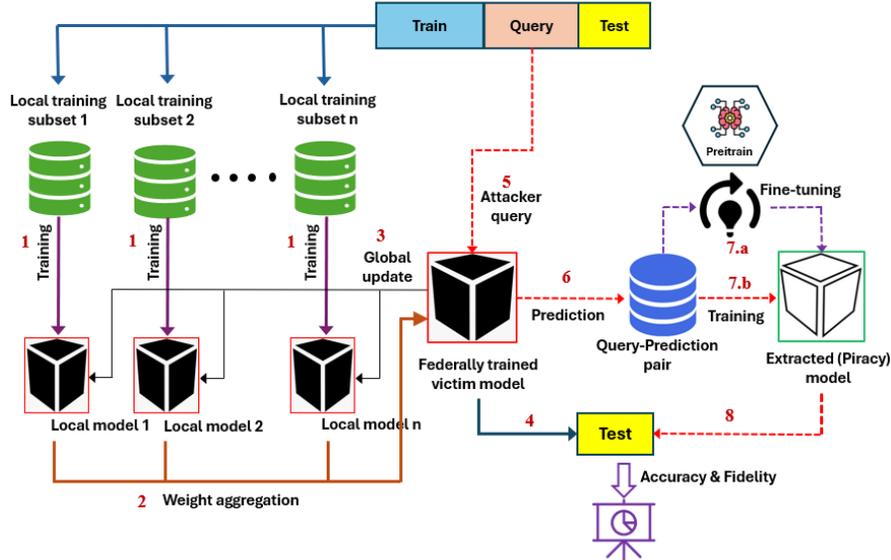
Fig. 2: Framework of the proposed TL-based ME attack executed in an FL environment.

querying the global model. The fidelity of the surrogate indicates how closely it mirrors the original model, while attack accuracy measures its predictive performance against the victim model. Once trained, $\mathcal{A}$ can utilize $\mathcal{M}_e$ to bypass service restrictions or exploit the proprietary model's functionality without authorization. $\mathcal{M}_e$ can also facilitate additional attacks, all while maintaining no direct access to the original training data or model parameters.

IV. PROPOSED ME ATTACK IN FL

In this section, we explain the proposed TL-based ME attack in an FL environment. We first briefly describe the attack environment of datasets, extracted models, FL environments, and attack evaluation metrics.

Figure 2 presents the proposed framework for executing a TL-based ME attack within a FL environment.

Datasets – In this experiment, we utilize three benchmark datasets: CIFAR-10, FashionMNIST, and MNIST .Each dataset is partitioned into training and testing sets, where the test sets remain constant throughout the experiment to evaluate the performance of both the victim and extraction models. The training set is evenly split, with one half used to train the $\mathcal{M}_v$ and the other half designated for generating the query dataset used in the model extraction process.

Extracted Models – We consider two machine learning models, basic CNN and ResNet, to evaluate the effectiveness of the ME attacks. In the default setting, the model parameters are randomly initialized, meaning the models are trained from scratch. To further enhance the performance of the ME attack, we also include pre-trained models, which are subsequently fine-tuned on the extraction dataset.

FL Environments – We design and implement an FL architecture to evaluate the ME attacks, incorporating two configurations: one with five clients and another with ten clients. In both setups, the FedAvg algorithm is employed to aggregate the model updates from the clients at the central

server, creating a global model [17]. For this experiment, we utilize the NVFlare (NVIDIA) library to develop and execute the FL architecture on GPU, ensuring efficient distributed training across the clients.

Metrics – To assess the effectiveness of ME attacks, we focus on three key metrics:

Accuracy represents the proportion of inputs from the test set that are correctly classified by $\mathcal{M}_e$.

Fidelity quantifies the proportion of inputs from the test set that are classified identically by both $\mathcal{M}_v$ and $\mathcal{M}_e$. Formally,

$$Fidelity(\mathcal{M}_e) = \frac{\sum_{(x,y)\in\mathcal{D}^\star} \mathbb{1}\{\mathcal{M}_v(x) = \mathcal{M}_e(x)\}}{|\mathcal{D}^\star|} \quad (1)$$

Kullback-Leibler (KL) divergence quantifies the difference between two probability distributions. In ME attacks, it measures how the output probability distributions of $\mathcal{M}_v$ and $\mathcal{M}_e$ differ.

A. The Overall Attack Paradigm

This subsection presents the fundamental steps and pseudo-code-based algorithm for the proposed TL-based ME attack implementation (Algorithm 1).

1) Prepare the query dataset by generating a set of inputs.
2) Query $(\mathcal{M}_v)$ by sending each input (q_i) to $(\mathcal{M}_v)$ and store the predictions (p_i) for each query (q_i) to create input-output pairs (q_i, p_i).
3) Construct the extracted dataset by collecting all the input-output pairs (q_i, p_i) gathered during the querying phase. This dataset (D) serves as the training data for the extracted model, enabling it to learn and replicate the behavior of the victim model.
4) Measure accuracy, fidelity, and KL divergence to validate the extracted model $(\mathcal{M}_e)$ by utilizing the test dataset $\mathcal{D}^*$ to evaluate the performance of both $(\mathcal{M}_v)$ and $(\mathcal{M}_e)$.

We utilize the FL architecture to train the $\mathcal{M}_v$, while the extraction model is trained using conventional machine learning methods. First, the server initializing global model parameters and sending them to all clients. Each client updates the model locally with their data and sends the updated parameters back to the server, which aggregates these updates over multiple communication rounds to produce a trained global victim model.

Algorithm 1 TL-based ME Attack

Input: Query dataset $\mathcal{Q}$, Victim model $\mathcal{M}_v$, Test dataset $\mathcal{D}^*$, Pre-train flag pre_train

Output: Extracted Model $\mathcal{M}_e$

1: Set $\mathcal{D} = []$ ▷ *Extracted Dataset*
2: Initialize $\mathcal{M}_e \leftarrow$ Model Architecture
3: **for** each q_i in $\mathcal{Q}$ **do**
4:     Send q_i to the $\mathcal{M}_v$
5:     Record predictions $p_i \leftarrow \mathcal{M}_v(q_i)$
6:     Update $\mathcal{D} \leftarrow (q_i, p_i)$
7: **end for**
8: **if** pre_train **then**
9:     Load pre-trained $\mathcal{M}_e$
10:    Fine-tune $\mathcal{M}_e$ on $\mathcal{D}$
11: **else**
12:    Train $\mathcal{M}_e$ on $\mathcal{D}$
13: **end if**
14: Compute $Accuracy(\mathcal{M}_e)$
15: Compute $Fidelity(\mathcal{M}_e)$, using Eq. (1)
16: Compute $KL\ Divergence(\mathcal{M}_v \| \mathcal{M}_e)$
17: **return** $\mathcal{M}_e$

The TL-based ME attack algorithm involves two primary stages: generating the query dataset and applying transfer learning presented in Algorithm 1. In the first part (lines 1-7), the algorithm queries the $\mathcal{M}_v$ using each sample from the query dataset $\mathcal{Q}$, collecting the $\mathcal{M}_v$'s predictions to form an extracted dataset $\mathcal{D}$. In the second part (lines 8-12), if pre-train is enabled, a pre-trained model is loaded and fine-tuned on the extracted dataset. Otherwise, the extraction model $\mathcal{M}_e$ is trained from scratch. The algorithm then evaluates $\mathcal{M}_e$'s performance based on accuracy, fidelity, and KL divergence before returning $\mathcal{M}_e$.

V. RESULT ANALYSIS AND DISCUSSION

This section presents the experimental setup, demonstrates the ME attack on both model types, and introduces an enhanced ME attack with pre-trained surrogate models and a TL approach.

A. Experimental Setup

The proposed ME attack is divided into two subsection: victim model training and extraction model training. For victim model training, we employ both centralized and federated training approaches.

For the federated training approach, we outline an FL architecture designed to train ML models across various datasets.

In FL architecture, the datasets are divided into N distinct subsets and distributed into N clients. Each client then trains a local instance of the model on its respective dataset. After local training, the clients securely transmit their model weights to a central server. The server aggregates these weights using a federated averaging method to create a global model, which is then redistributed to the clients [17]. In subsequent rounds, the clients perform another epoch of local training on the updated global model and share the new weights with the server. This process is repeated for T rounds to finalize the global model $\mathcal{M}_v$.

For the extraction (or surrogate) model, we utilize four distinct sample query datasets, consisting of 5k, 10k, 15k, and 20k queries. These query sets are used to generate query-prediction pairs, which are subsequently used to train $\mathcal{M}_e$. In this study, we mainly focus on three datasets- CIFAR-10, MNIST, and FashionMNIST, and two deep learning (DL) models, basic CNN and ResNet to evaluate the ME attack. Initially, $\mathcal{M}_e$ is trained from scratch without utilizing pre-trained model parameters. We then present a TL-based ME attack approach, where pre-trained model parameters are utilized for $\mathcal{M}_e$ to enhance the ME attack performance.

B. ME Attack without Pre-trained Model

In this section, we demonstrate the effectiveness of the ME attack on machine learning models trained using both centralized and federated approaches with 5 and 10 clients. Table I summarize the accuracy, fidelity, and KL divergence of the ME attack for CIFAR-10, MNIST, and FashionMNIST. The accuracy of $\mathcal{M}_v$ serves as the baseline accuracy, which we aim to closely match. In this experiment, we apply both centralized and federate training approaches to the victim model, while for $\mathcal{M}_e$, we exclusively employ centralized training.

For example, in the case of the CIFAR-10 dataset using a basic CNN model, the baseline accuracy of $\mathcal{M}_v$ is approximately 80.19%. Our objective is to closely approach this baseline accuracy in the extraction model. We observe that the accuracy of $\mathcal{M}_e$ is directly correlated with the size of the query set. A query dataset consisting of 25k samples typically yields the highest $\mathcal{M}_e$ accuracy in both centralized and federated architectures. Similarly, for the ResNet model, accuracy is also strongly influenced by the size of the query set.

Similar trends are observed across other datasets, such as FashionMNIST and MNIST, where accuracy improves consistently with increasing query set size for both the CNN and ResNet models. Moreover, the $\mathcal{M}_v$'s performance also affects the accuracy of the extraction model. For instance, in the FashionMNIST dataset, the highest baseline accuracy is achieved using the basic CNN model with training distributed across five clients, which consequently results in the best extraction model accuracy for that setup.

When comparing the accuracy between the basic CNN and ResNet models, the basic CNN outperforms the ResNet model on both the CIFAR-10 and FashionMNIST datasets. However,

	Metrics	No. of FL Clients	Basic CNN					ResNet				
			Victim model	Extraction model				Victim model	Extraction model			
				5K	10K	20K	25k		5K	10K	20K	25K
CIFAR-10	Accuracy (%)	Centralized	80.19	59.99	63.52	70.72	73.7	76.22	52.40	58.73	65.24	68.36
		5	79.98	56.49	63.75	70.67	71.86	81.05	52.55	58.65	65.9	68.07
		10	76.57	53.13	63.83	70.05	73.05	82.39	52.12	59.19	66.03	68.94
	Fidelity (%)	Centralized	N/A	61.59	62.44	71.38	73.52	N/A	51.74	57.9	63.44	66.87
		5		58.3	65.02	71.67	72.57		52.74	58.71	66.04	68.15
		10		56.38	65.23	70.16	73.4		52.82	58.82	66.64	68.38
	KL Divergence	Centralized	N/A	0.0322	0.02744	0.02265	0.02069	N/A	0.000486	0.000416	0.000348	0.000318
		5		0.0344	0.02837	0.022459	0.01899		0.000567	0.000458	0.000374	0.0003436
		10		0.03076	0.02349	0.01863	0.0198		0.000577	0.000493	0.000368	0.000347
MNIST	Accuracy (%)	Centralized	98.94	98.53	99.06	99.33	99.39	99.08	98.36	98.57	99.17	99.24
		5	99.46	98.62	99.02	99.12	99.33	99.46	97.83	98.76	99.12	99.01
		10	99.39	98.68	99.14	99.37	99.32	99.48	97.48	99.04	99.02	99.17
	Fidelity (%)	Centralized	N/A	98.48	98.82	98.85	98.87	N/A	97.25	97.87	98.45	98.59
		5		98.58	99.05	99.2	99.3		97.51	98.79	98.41	98.98
		10		98.70	99.04	99.3	99.3		97.61	98.83	98.52	98.79
	KL Divergence	Centralized	N/A	0.0013	0.001078	0.00103	0.0011697	N/A	3.60E-05	2.62E-05	1.77E-05	1.52E-05
		5		0.00146	0.000926	0.00075	0.000577		3.04E-05	1.54E-05	2.35E-05	1.18E-05
		10		0.00129	0.000919	0.00056	0.000572		3.0379E-05	2.38E-05	1.72E-05	1.26E-05
Fashion-MNIST	Accuracy (%)	Centralized	92.01	88.16	88.7	90.74	91.23	89.65	83.62	86.94	88.98	89.49
		5	92.21	87.96	89.57	91.22	91.39	91.78	85.30	87.25	89.43	88.88
		10	91.58	87.64	89.31	91.04	91.16	91.85	85.97	87.03	89.29	89.7
	Fidelity (%)	Centralized	N/A	89.67	90.83	91.68	92.05	N/A	83.15	86.85	88.12	89.01
		5		89.53	91.29	92.49	92.72		86.83	87.2	89.71	89.95
		10		89.88	91.03	91.5	91.48		85.99	87.09	89.66	89.65
	KL Divergence	Centralized	N/A	0.0135	0.006298	0.00677	0.00628	N/A	0.000125	9.50E-05	8.84E-05	8.71E-05
		5		0.00789	0.006897	0.005633	0.00572		0.000134	1.20E-04	1.05E-04	9.13E-05
		10		0.00693	0.00653	0.006176	0.00578		0.0001416	1.14E-04	9.32E-05	9.23E-05

TABLE I: ME atatck accuracy and fidelity across various FL clients on CIFAR-10, MNIST, and FashionMNIST datasets, evaluated on different DL model architectures.

when employing a pre-trained ResNet model, we achieve significantly better results (as discussed later).

Since the MNIST datasets are less complex than CIFAR-10, the extraction model can achieve high accuracy with fewer query samples. For example, using the FashionMNIST dataset with only a 5k query set, we achieve an extraction accuracy of approximately 88.16% with the basic CNN model, compared to a baseline accuracy of around 92%. However, under the same conditions with the ResNet model, the accuracy is significantly lower. Thus, we hypothesize that the extraction model's accuracy is strongly influenced by both the size of the query set and the performance of $\mathcal{M}_v$.

To further evaluate the ME attack's effectiveness, we assess fidelity and KL divergence to measure how well the extraction model approximates $\mathcal{M}_v$. Similar to accuracy, these metrics are influenced by the query dataset size, larger datasets generally yield higher extraction accuracy and, consequently, better fidelity and lower divergence. Since fidelity is inherently tied to extraction model accuracy, improvements in accuracy enhance fidelity, while reductions degrade both fidelity and KL divergence. These findings underscore the critical role of query set size and victim model performance in shaping ME attack success.

C. ME Attack with Pre-trained Model

We leverage the TL approach to enhance the performance of the ME attack across multiple datasets [18], [19]. In this experiment, we employ a pre-trained ResNet model as the extraction model, which is subsequently fine-tuned on the extracted dataset. The ME attack performance for all query sets on the victim ResNet model, trained on the CIFAR-10 and FashionMNIST datasets, is presented in Table II and Table III, respectively. Across all query sets, the pre-trained

TABLE II: ME attack performance using ResNet pre-trained model on CIFAR-10.

Metric	N	Training-from-scratch				Using pre-trained model			
		5k	10k	20k	25k	5k	10k	20k	25k
Accuracy	0	52.4	58.73	65.24	68.66	63.75	70.94	73.67	75.77
	5	52.55	58.65	65.9	68.07	67.27	70.27	73.89	75.61
	10	52.12	59.19	66.03	68.94	66.88	69.70	73.65	76.12
Fidelity	0	51.74	57.9	63.44	66.87	62.52	68.57	70.69	70.88
	5	52.74	58.71	66.04	68.15	68.3	70.3	73.67	74.69
	10	52.82	58.82	66.64	68.38	68.98	69.94	73.45	74.16

TABLE III: ME attack performance using the ResNet pre-trained model on FashionMNIST.

Metric	N	Training-from-scratch				Using pre-trained model			
		5k	10k	20k	25k	5k	10k	20k	25k
Accuracy	0	83.62	86.94	88.98	89.49	87.95	88.86	89.73	90.44
	5	85.3	87.25	89.43	88.88	87.72	88.41	90.41	90.17
	10	85.97	87.03	89.29	89.7	88.33	89.24	90.27	90.8
Fidelity	0	83.15	86.85	88.12	89.01	88.57	88.08	88.94	90.23
	5	86.83	87.2	89.71	89.95	89.19	89.64	91.3	91.28
	10	85.99	87.09	89.66	89.65	89.46	90.72	90.82	90.44

model consistently surpasses the original extraction model in both accuracy and fidelity metrics.

For instance, in the CIFAR-10 dataset, the highest accuracy achieved by the basic CNN model with a 25k query set is around 73%. However, when applying the TL approach on the same victim model, this accuracy is surpassed with a smaller query set (20k). The highest recorded extraction model accuracy for CIFAR-10 is around 76.12%, closely matching the baseline accuracy of 76.52%, effectively replicating the victim model's performance. A similar trend is observed in

the FashionMNIST dataset, where TL leads in both accuracy and fidelity metrics across various query sets.

Figure 3 and Figure 4 illustrate the notable improvements in both accuracy and fidelity for the CIFAR-10 dataset. These results clearly indicate that the performance of the ME attack is strongly tied to the extracted model's parameters. By incorporating pre-trained parameters, the extraction model achieves significantly better accuracy and fidelity, particularly with smaller query sets, thereby improving the overall effectiveness of the ME attack. For example, with a 10k query set on the CIFAR-10 dataset, the TL approach results in approximately 12.21% higher accuracy compared to the original extraction model accuracy.
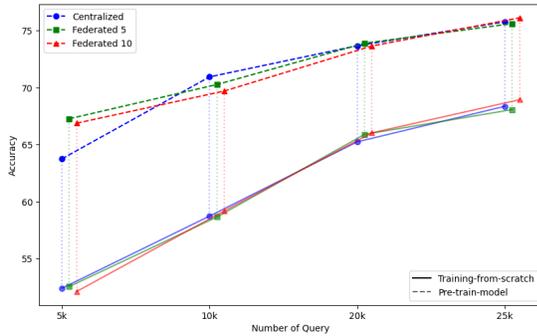

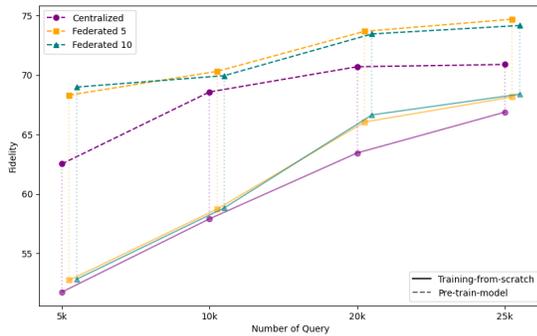Fig. 3: ME attack accuracy with ResNet pre-trained model on CIFAR-10.

Fig. 4: ME attack fidelity with ResNet pre-trained model on CIFAR-10.

VI. Conclusion and future Scope

In this study, we examine the vulnerability of FL-based models to ME attacks, showing that the accuracy and fidelity of extracted models are significantly affected by factors such as query set size, model architecture, and training datasets. Further, incorporating transfer learning (TL) into the extraction process notably enhances attack performance, especially with smaller query sets.

In the future, we plan to focus on developing defenses against ME attacks, including techniques like noise injection in API responses and other stronger privacy-preserving methods in FL environments. Extending the research to other data types and exploring larger, more diverse federated networks will offer further insights into mitigating these risks.

References

[1] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, vol. 2, no. 2, 2016.

[2] M. Vucovich, A. Tarcar, P. Rebelo, N. Gade, R. Porwal, A. Rahman, C. Redino, K. Choi, D. Nandakumar, R. Schiller *et al.*, "Anomaly detection via federated learning," *arXiv preprint arXiv:2210.06614*, 2022.

[3] J. Chen, X. Zhang, R. Zhang, C. Wang, and L. Liu, "De-pois: An attack-agnostic defense against data poisoning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3412–3425, 2021.

[4] D. Thakur, S. Roy, S. Biswas, E. S. Ho, S. Chattopadhyay, and S. Shetty, "A novel smartphone-based human activity recognition approach using convolutional autoencoder long short-term memory network," in *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2023, pp. 146–153.

[5] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang, "Source inference attacks in federated learning," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1102–1107.

[6] S. Banerjee, S. Roy, S. F. Ahamed, D. Quinn, M. Vucovich, D. Nandakumar, K. Choi, A. Rahman, E. Bowen, and S. Shetty, "Mia-bad: An approach for enhancing membership inference attack and its mitigation with federated learning," in *2024 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2024, pp. 635–640.

[7] C. Ma, J. Li, M. Ding, H. H. Yang, F. Shu, T. Q. Quek, and H. V. Poor, "On safeguarding privacy and security in the framework of federated learning," *IEEE network*, vol. 34, no. 4, pp. 242–248, 2020.

[8] J. Liang, R. Pang, C. Li, and T. Wang, "Model extraction attacks revisited," in *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, 2024, pp. 1231–1245.

[9] X. Gong, Y. Chen, W. Yang, G. Mei, and Q. Wang, "Inversenet: Augmenting model extraction attacks with training data inversion." in *IJCAI*, 2021, pp. 2439–2447.

[10] M. Kesarwani, B. Mukhoty, V. Arya, and S. Mehta, "Model extraction warning in mlaas paradigm," in *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018, pp. 371–380.

[11] J. Li, A. S. Rakin, X. Chen, L. Yang, Z. He, D. Fan, and C. Chakrabarti, "Model extraction attacks on split federated learning," *arXiv preprint arXiv:2303.08581*, 2023.

[12] A. Vangala, S. Roy, and A. K. Das, "Blockchain-based lightweight authentication protocol for iot-enabled smart agriculture," in *2022 International Conference on Cyber-Physical Social Intelligence (ICCSI)*. IEEE, 2022, pp. 110–115.

[13] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction {APIs}," in *25th USENIX security symposium (USENIX Security 16)*, 2016, pp. 601–618.

[14] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1345–1362.

[15] A. K. Das, S. Roy, E. Bandara, and S. Shetty, "Securing age-of-information (aoi)-enabled 5g smart warehouse using access control scheme," *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 1358–1375, 2022.

[16] J.-B. Truong, P. Maini, R. J. Walls, and N. Papernot, "Data-free model extraction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4771–4780.

[17] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[18] S. F. Ahamed, P. Aggarwal, S. Shetty, E. Lanus, and L. J. Freeman, "Attl: An automated targeted transfer learning with deep neural networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–7.

[19] S. F. Ahamed, K. A. Islam, and S. Shetty, "Targeted transfer learning: Leveraging optimal transport for enhanced knowledge transfer," in *2024 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2024, pp. 506–510.