TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 16/23 pp2204–2220
DOI: 10.26599/TST.2024.9010243
Volume 31, Number 4, August 2026

Research Article

From Model Parameters to Data Quality: Implicit Factor Evaluation of Model Extraction Attacks

Anli Yan, Lang Li*, Kanghua Mo, Peigen Ye, Shaowei Wang, Li Hu, Hongyang Yan, and Jin Li

Abstract: Model extraction attacks (MEAs) pose a significant threat to deep learning (DL) models, where adversaries aim to steal the decision behavior of targeted DL models. While several works have shown the ability of a surrogate model to mimic the target DL model, the underlying factors that make a DL model vulnerable to MEAs are unclear. Analyzing these underlying factors is the key to enhancing the security of DL systems. This involves exploring MEAs in diverse scenarios to understand the relationship between their success and the features of DL systems. In this paper, we evaluate the underlying factors influencing MEAs from two crucial perspectives: the model's intrinsic parameters and the quality of the data used. For the model's intrinsic parameters, we focus on how the batch size, learning rate, and optimizer influence the effectiveness of MEAs. Regarding data quality, we conduct an in-depth analysis of how data annotation and selection affect MEAs' success. Our study includes analyzing variations in batch size, five learning rates, eight optimizers, the impact of varying proportions of dirty data, and the effects of subtle changes in data richness. The results of our research reveal a diverse range of susceptibilities to MEAs.

Key words: model extraction attacks (MEAs); model parameters; data quality; impact factors

1 Introduction

Deep learning (DL) models are booming in fields as diverse as healthcare, finance, cybersecurity, and autonomous systems, becoming the backbone of numerous applications[1]. Consequently, DL models, with their complex architecture and profound capabilities, have become valuable intellectual properties, heralding an era of unprecedented possibilities[2]. However, model extraction attacks (MEAs) are a serious threat, casting a shadow over the intellectual property and confidentiality of the model[3, 4]. These attacks enable massive unauthorized access and usage of proprietary models, resulting in the theft of valuable intellectual property and the unique parameters that define a model's essence and effectiveness[5, 6]. The consequences of a successful MEA include catastrophic breaches of model confidentiality and the potential for fake models to spread widely across the digital ecosystem[7, 8]. According to statistics, at least 97 models in LiblibAI, the largest open-source community for text and image models, have been illegally copied to a website called "Yizhou Intelligence" for profit without permission.

• Anli Yan is with School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China, and also with School of Artificial Intelligence, Guangzhou University, Guangzhou 510006, China. E-mail: anli_yan2021@163.com.
• Lang Li, Kanghua Mo, Shaowei Wang, Li Hu, Hongyang Yan, and Jin Li are with School of Artificial Intelligence, Guangzhou University, Guangzhou 510006, China. E-mail: lilang@gzhu.edu.cn; mokanghua@gmail.com; wangsw@gzhu.edu.cn; lihu6479@gmail.com; hyang_yan@gzhu.edu.cn; lijin@gzhu.edu.cn.
• Peigen Ye is with School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China. E-mail: ypgmhxy@gmail.com.
* To whom correspondence should be addressed.
Manuscript received: 2024-07-19; revised: 2024-09-27; accepted: 2024-12-08

Furthermore, the MEA paves the way for a host of malicious activities, such as model inversion and adversarial attacks[9], each of which has its own set of harmful effects. Hence, it is vital to explore the underlying factors that make models vulnerable to MEAs, including their architecture, training process, and deployment environment[10]. This awareness is critical to cultivating robust defense strategies to ensure our technology is protected from malicious threats that compromise model confidentiality and infringe intellectual properties[11].

The current research on the impact factors contributing to MEAs in the field of DL is both intense and enlightening, yet it also reveals critical gaps in our understanding[12, 13]. Significant efforts have been focused on the strategies used by adversaries to shed light on popular and effective methods for illegally extracting model information[14, 15]. This research delves into various aspects, including how the adversary selects specific data and model architectures, aiming to uncover their methods and technologies in detail[16]. For example, adversaries do not need to select a model with a complex architecture as a surrogate model to achieve excellent MEAs without knowing the target model architecture[17]. Analyzing the motivations and impacts behind these choices aims to reveal their attack methods and technologies and provide a comprehensive understanding of current attack methods. However, current research shows a clear gap, as it pays insufficient attention to the intrinsic parameters of the model and the impact of training data. Exploring and evaluating the intrinsic parameters of DL models, along with the nature of their training data, is paramount for gaining insights and enhancing defenses against MEAs[18]. The intrinsic parameters of the model, such as parameter optimization method, learning rate, and batch size, directly affect the way the model processes input data and its sensitivity to MEAs. The quality, diversity, and annotation accuracy of training data directly affect the model's generalization ability and resistance to MEAs.

In this paper, we conduct an evaluation of MEAs by considering multifaceted perspectives, encompassing both the intrinsic parameters of the model and the quality of data. Concerning the intrinsic parameters of the model, it is imperative to evaluate various parameters such as batch size, learning rate, and optimizer. By exploring these model-specific parameters, our goal is to identify potential weaknesses vulnerable to malicious manipulation, enhancing the model's resistance to MEAs. Regarding data quality, we analyze the impact of various elements, such as data annotation and data selection, on MEAs. Our exploration focuses on uncovering potential vulnerabilities arising from these data-centric aspects to better understand their susceptibility to manipulation and exploitation. We conduct extensive empirical analysis to answer the five questions: (1) What is the impact of batch size on MEAs? (2) What is the impact of learning rate schemes on MEAs? (3) What is the impact of optimizers on MEAs? (4) What is the impact of data annotation on MEAs? (5) What is the impact of data selection on MEAs? The analysis of these five questions yields a series of valuable and insightful results. For example, the learning rate with MultiStepLR appears to be more resilient to MEAs; optimizers with Adadelta are more vulnerable to MEAs, which can lead to greater risks of model privacy leaks. Note that our work focuses on image classification. Text generation, object detection, etc. are not within our scope and may be explored in the future. Contributions of this paper are summarized as follows:

- We delve into exploring how the model's intrinsic parameters, coupled with data quality, influence MEAs. This critical exploration illuminates the underlying factors affecting MEAs, laying the groundwork for a deeper understanding of these vulnerabilities.
- Regarding the model's intrinsic parameters, our work uncovers their impact on MEAs through a detailed analysis of key model parameters. Specifically, we concentrate on evaluating elements such as batch size, learning rate, and optimizer, to evaluate their respective vulnerability and resilience against MEAs.
- Concerning the impact of data quality, our evaluation focuses on two key aspects: data annotation and data selection. We evaluate data annotation to understand how inaccurate annotations affect MEAs. Data selection analysis aims to gain insights into how the choice of certain datasets influences the model's vulnerability to MEAs.

Roadmap. Section 2 presents the related work in this paper. Section 3 introduces background. Section 4 presents our threat models. Section 5 describes the experimental design and implementation. Section 6

discusses the experimental results. Section 7 concludes the paper.

2 Related Work

Model extraction attacks have fallen into two camps based on the stolen goal: internal knowledge[19] and decision behavior[20].

Internal knowledge aims to steal the specific parameters of the target model, e.g., weights, network layers, convolutional numbers, etc. For example, Yan et al.[21] found that the reasoning of DL heavily depends on generalized matrix multiplication (GEMM), so they proposed a cache-based model extraction attack against GEMM. Moreover, Jagielski et al.[20] conducted a model extraction attack on neural networks with ReLU activation functions, leveraging the technique of reverse engineering based on the fact that each ReLU network defines a piecewise linear function[22]. Carlini et al.[23] proposed a model extraction attack from the perspective of cryptography, which restores the weight parameters of the target model by selecting a plaintext attack. Canales-Martínez et al.[24] successfully extracted all real-valued parameters of a deep neural network (DNN) based on the ReLU activation function with arbitrarily high accuracy using polynomial query numbers and polynomial time complexity methods. Gao et al.[25] proposed an end-to-end attack method called DeepTheft, which utilizes a power side channel based on run average power limitation (RAPL) to accurately recover complex DNN model architectures on general-purpose processors. Although model extraction attack methods aimed at stealing a target model's internal parameters can achieve more accurate recovery of information, their effectiveness is often contingent on the adversary possessing additional background knowledge or the attacks being conducted in specific, limited scenarios.

Decision behavior aims to imitate the decision behavior of the target model while ignoring the internal knowledge of the target model. For example, Wu et al.[26] proposed a model extraction attack against graph neural networks, which formalizes and characterizes the adversary's knowledge from three perspectives: node attributes, graph structure, and shadow subgraphs. He et al.[15] proposed DRMI, a model extraction attack technique utilizing mutual information technology to choose samples with significant information content. Truong et al.[27] proposed a data-free model extraction attack. In the data-free model extraction attack, the adversary does not need to specifically collect samples but uses the generated model to synthesize data[16]. The model extraction of decision behaviors is suitable for any complex target model and does not need to understand any knowledge of the target model[9]. Rosenthal et al.[14] proposed a new generator training scheme that combines inconsistency loss, diversity loss, and experience replay to enable the generator to generate higher-quality instances, effectively improving the training performance of cloned models. To address the issue of model extraction attacks in real-world scenarios with limited query budgets, Lin et al.[16] proposed QUDA, a novel query-limited data-free model extraction attack. QUDA combines a generative adversarial network (GAN) pre-trained on a publicly irrelevant dataset to provide weak image priors and employs deep reinforcement learning technologies to improve the efficiency of query generation strategies. Zhuang et al.[28] proposed StealGNN, which is the first data-driven model extraction attack framework for graph neural networks (GNNs). StealGNN has driven the development of GNN extraction attacks in three key aspects: completely data-free, full-rank attacks, and the ability to cope with the most challenging hard-label attacks. In this paper, we focus on evaluating the impact factors of model extraction attacks for decision behavior in the field of image-based classification. We select this type of model extraction attack for evaluation because its mechanism aligns more closely with real-world attack scenarios, where the adversary may not possess specific knowledge about the target model.

3 Background

3.1 DL

With the explosion of data resources and the steady increase in computational resources, DL has emerged as a promising research field with enormous potential for a broad range of applications[29, 30]. The excellent performance of DL in fields such as computer vision, natural language processing, speech recognition, and autonomous driving has further attracted significant investments and researchers' attention around the world[31]. Large corporations heavily invested in DL use the Machine Learning as a Service (MLaaS) platform[32] to provide access to their trained models.

They offer services through application programming interfaces (APIs), allowing users to query and receive predictions or responses, which generates substantial economic benefits[33]. As a result, the platforms are expected to continue to play a paramount role in providing the necessary conditions for the deployment of DL models.

However, various studies have highlighted the potential security risks associated with deploying high-performance DL models on MLaaS platforms and offering services to users exclusively through APIs[34]. These security risks include adversarial attacks and privacy leakage. Adversarial attacks[35], on one hand, exploit model vulnerabilities with crafted inputs, causing it to misbehave. On the other hand, MLaaS platforms have the risk of privacy leakage and may expose users' sensitive data and confidential information[36]. These types of security breaches are particularly challenging and can cause severe financial and reputational damage to an organization or company. Therefore, ensuring the privacy and security of DL models has become crucial[37]. In this paper, we aim to unearth the underlying causes of model privacy leakage. By conducting extensive analysis and research on the influencing factors of MEAs, a major threat to model privacy, we hope to develop effective strategies to mitigate these risks.

3.2 MEAs

MEAs aim to extract information from well-trained DL models, enabling adversaries to create fake models with similar functionality[20]. This poses a significant threat to the company's competitive advantage and proprietary technology and could result in financial losses, loss of market share, and erosion of reputation and customer trust[38]. Furthermore, the MEA facilitates downstream attacks such as model inversion, membership inference, and adversarial example attacks[9], further compromising the security of DL models.

MEAs usually consist of three main steps as shown in Fig. 1. The first step for adversaries is to obtain query samples of the target model. The target model refers to the object of the adversary's attack. The adversary can collect query samples in various ways, such as crawling public datasets or purchasing datasets. Furthermore, the adversary can also optimize the quality of query samples by filtering or creating crafted samples. In the second step, the adversary uses the query sample obtained in the first step to access the target model and obtain coupled prediction information, i.e., labels or confidence scores. Finally, the adversary selects an appropriate surrogate model architecture. Then, he uses a dataset of samples querying the target model, along with its soft labels (predicted by the target model), to train the surrogate model, effectively creating a copy of the target model. The details of the attack process in our empirical evaluation are described in Section 4.3.

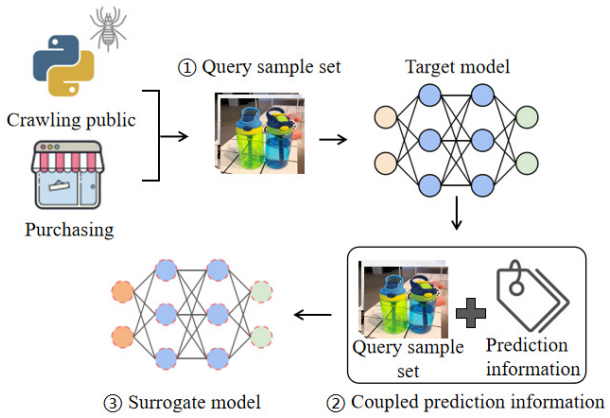


Fig. 1 Pipeline of model extraction attacks.

4 Threat Model

4.1 Adversary goal

We assume an adversary performs an MEA, using a public API to replicate the target model's decision behavior. $\mathcal{F}(X_t, \theta)$ is the target model. $\mathcal{F}(X_s, \theta')$ is the surrogate model. θ and θ' are the parameters of the target model and the surrogate model, respectively. The datasets used to train the target model and surrogate model are represented by X_t and X_s, respectively. X_e is the dataset used to evaluate the accuracy of the MEA. In this paper, we do not consider the similarity of θ and θ'. Instead, we focus on the consistency of the target model and the surrogate model in predicting labels for the evaluation set X_e. Section 5.1 provides a detailed description of the metrics for the MEA in this paper.

4.2 Adversary knowledge

Datasets. We assume that adversaries have limited unlabelled data; otherwise, they can train their own models without stealing the target model of the defenders. We determine the availability of the following two datasets to the adversary. The first query dataset shares the same distribution as the training set

of the target model but has no overlapping samples. The second query dataset has a different distribution from the training set of the target model. Our assumptions align with standard practices in MEAs as demonstrated by numerous studies[15, 39]. It is worth noting that the labels for the query dataset are obtained by accessing the target model. Section 5.2 provides a detailed description of the datasets used in this paper.

Model architectures. The adversary has a certain level of proficiency in DL and is able to select suitable model architecture for a given task. However, the adversary lacks the ability to optimize surrogate models using advanced techniques. These techniques involve skills such as model design, pruning, and parameter compression, which are beyond the adversary's capabilities. Section 5.3 provides a detailed description of the model architecture used by the surrogate model in this paper.

4.3 Adversary capabilities

Based on the workflow diagram for MEAs shown in Fig. 1, we provide a detailed description of the adversary's attack capabilities in the evaluation experiments conducted in this paper. As illustrated in Fig. 1 and described in Section 3.2, the basic steps involved in MEAs include query sample selection, query, and surrogate model training. Sample collection and surrogate model training are the most crucial components in MEAs. However, understanding, implementing, and evaluating a large number of unique extraction attacks requires significant technical effort and time, and access to this knowledge is limited. For the sake of generality, we analyze attacks that depend on abstracted attack processes from state-of-the-art MEAs in our evaluation work. For query sample selection, we employ random sampling when evaluating the impact of factors on MEAs. The specific information on the dataset owned by the adversary is described in Section 5.2. For the query stage, the target model only provides prediction labels for the query samples, but does not provide a posterior probability. For surrogate model training, both the adversary and the target model adopt the same model architecture. In the experimental setup of our study, the computations are facilitated by a high-performance graphics processing unit (GPU), specifically the NVIDIA GeForce RTX 4090. This graphics card is equipped with 24 GB of GDDR6X memory. For the MNIST[40] and Fashion MINIST (FMINIST)[41] datasets, using VGG19 as the surrogate model architecture, it takes approximately 1.5 h for adversaries to perform model extraction attacks. On the CIFAR10[42] dataset, using ResNet50 as the surrogate model, the execution time for extracting attacks is about 2 h. Similarly, on the CIFAR100[42] dataset, when using ResNet50 as the surrogate model, the time to perform model extraction attacks is approximately 3 h. All surrogate models are trained for 200 epochs. In this paper, our core objective is to comprehensively examine the impact of five distinctive factors on MEAs. Thus, we simplify and abstract the most fundamental MEA scheme in order to make it more general-purpose and widely applicable. This method allows us to derive more meaningful insights into the security of machine learning models and to develop more effective countermeasure strategies.

5 Evaluation Methodology

This study aims to reveal the impact of the model's intrinsic parameters and data quality on MEAs. In this section, we describe the specific setup used for the evaluation metrics, datasets, and evaluation experiments. It is important to note that all our experiments focus on evaluating the target model, which is the object of adversarial attacks and the entity we aim to protect.

5.1 Evaluation metrics

We leverage two evaluation metrics, i.e., fidelity (FI) and task accuracy ratio (TAR). FI is the consistency between the surrogate model and the target model for the prediction label of the evaluation set. TAR is the ratio of prediction accuracy between the surrogate model and the target model on the evaluation dataset. Formally, we assume that $D_e = \{x_1, x_2, \ldots, x_N\}$ is the evaluation dataset, and $Y_e = \{y_1^e, y_2^e, \ldots, y_N^e\}$ is the ground-truth label of the evaluation dataset. N is the number of samples in the evaluation dataset. $Y_s = \{y_1^s, y_2^s, \ldots, y_N^s\}$ is the prediction label of the surrogate model for the evaluation dataset. $Y_t = \{y_1^t, y_2^t, \ldots, y_N^t\}$ is the predicted labels of the evaluation dataset by the target model. The calculation equation of TAR is shown in Eq. (1). The calculation equation of FI is shown in Eq. (2). Accuracy$(\cdot,\cdot)$ function is a widely used standard for evaluating the performance of classification models. It is equal to the number of samples with consistent prediction labels divided by the total number of samples.

$$\mathrm{TAR} = \frac{\mathrm{Accuracy}(Y_e,\ Y_s)}{\mathrm{Accuracy}(Y_e,\ Y_t)} \tag{1}$$

$$\mathrm{FI} = \mathrm{Accuracy}(Y_t,\ Y_s) \tag{2}$$

5.2 Datasets

This paper is centered around MEAs in image classification and employs four of the most popular datasets, namely MNIST[40], Fashion MNIST (FMNIST)[41], CIFAR10[42], and CIFAR100[42]. The reason we do not select more complex datasets such as ImageNet is that the trends in the impact of data complexity on model extraction attacks are already available in all four datasets mentioned.

The MNIST and FMNIST datasets consist of grayscale images, each with dimensions of 28 × 28 pixels. The sample number of training sets and test sets is 60 000 and 10 000, respectively. The CIFAR10 and CIFAR100 datasets consist of 3 × 32 × 32 pixels, each with 50 000 samples in the training sets and 10 000 samples in the testing sets. In this paper, we require three types of datasets: datasets for training the target model D_t, datasets for training surrogate models D_s, and datasets for evaluating the MEA performance D_e. For the D_t dataset, we use training sets in the public dataset to train the target model. Specifically, for both the MNIST and FMNIST datasets, the training target model has a data sample size of 60 000. For both CIFAR10 and CIFAR100 datasets, the number of data samples of the training target model is 50 000. Notably, in the experiment of evaluating the impact of data quality on MEAs, the training dataset of the target model is modified based on the above D_t. The specific detailed settings are described in Sections 6.4 and 6.5. For the dataset D_s, we randomly select 80% of the samples in each test set of the four public datasets. For the dataset D_e, we randomly select 20% of the samples from the test set of each public dataset. It is worth noting that D_s and D_e do not overlap. The labels for the training dataset of the target model are ground-truth labels. The labels for the training dataset of the surrogate model are provided by the target model, i.e., soft labels.

Moreover, taking into account the knowledge of adversaries described in Section 4.2, dataset D_s falls in two types: same distribution and different distribution. The target model uses the same type of dataset as the surrogate model when we set it to the same distribution. For example, when the target model uses the MNIST dataset, the surrogate model also uses the MNIST dataset. The target model and the surrogate model use different types of datasets when we set it to different distributions. The MNIST and FMNIST datasets serve as controls, while the CIFAR10 and CIFAR100 datasets are cross-referenced. For example, when the target model uses the MNIST dataset, the surrogate model uses the FMNIST dataset.

5.3 Experimental protocol

Our exploration into the impact of MEAs unfolds from two points: the intrinsic parameters of the model and data quality. Diving into the model's intrinsic parameters, we focus on three influencing factors: batch size, learning rate, and optimizer. For data quality, our attention is devoted to revealing the effects of data annotation and data selection on the susceptibility to the MEA. In our experiments, we use the ResNet50 architecture for both the target and surrogate models with datasets such as CIFAR10 and CIFAR100. For the MNIST and FMNIST datasets, however, we uniformly apply the VGG19 architecture to both models. The reason for not exploring additional model architectures is that existing research[17] has clarified the impact of differences between target and surrogate model architectures on model extraction attacks. The more complex the architecture of the target model, the more difficult it is for adversaries to steal. Therefore, we limit the architecture of the target model and surrogate models to ensure the reliability of the experimental results in this paper. The goals, knowledge, and abilities of the adversary in our evaluation experiment are described in Section 4.

5.3.1 Batch size

We aim to answer the question: what is the impact of batch size on MEAs? Batch size determines how many data samples are used in each training iteration. It affects the speed and stability of model learning, as well as memory usage and training time. In this experiment, various configurations are examined to evaluate the impact of batch size on the MEA. The batch size of the target model is set at multiple levels—16, 32, 64, 128, and 256—while the batch size of the surrogate model is set to 64. Furthermore, a consistent set of training parameters is maintained in both target and surrogate models for each dataset to ensure standardized comparisons. For instance, the learning rate is set to MultiStepLR for both models, and the Adam optimizer is uniformly adopted. This

ensures that variations in experimental results can be more reliably attributed to differences in batch sizes, thus facilitating a more focused and reliable analysis of their effects. The experimental analysis of the impact of target model batch size on the MEA is described in Section 6.1.

5.3.2 Learning rate

We aim to answer the question: what is the impact of learning rate schemes on MEAs? Learning rate is the key factor that determines the updating range of model parameters. Our purpose is to study the potential variability of risks associated with the MEA across different learning rates, aiming to increase awareness and circumvent potential model vulnerabilities. We evaluate five popular learning rate strategies: ReduceLROnPlateau, StepLR, MultiStepLR, ExponentialLR, and CosineAnnealingLR. For each dataset under consideration, the target model is trained to employ each of the aforementioned five learning rate strategies. In this experiment, we consider the configuration of the target model and the surrogate model. The optimizations in both models are performed using the Adam optimizer, maintaining a uniform batch size of 64 across all experiments. Moreover, the learning rate of the surrogate model is set to MultiStepLR. This standardized setting ensures consistent and comparative evaluation, enabling a centralized analysis of the impact of learning rates on the MEA. The experimental analysis of the impact of the learning rate for the target model on the MEA is presented in Section 6.2.

5.3.3 Optimizer

We aim to answer the question: what is the impact of optimizers on MEAs? The optimizer determines how the model parameters are updated. Different optimization algorithms have different mathematical characteristics, which affect the convergence speed and performance of the model. In this experiment, the optimizer of the target model is designated as a variable, while its learning rate and batch size are maintained constant, specifically set at MultiStepLR and 64, respectively. We evaluate the impact of eight prevalent optimizers on the MEA, namely adaptive delta (Adadelta)[43], adaptive gradient (Adagrad), adaptive moment estimation (Adam)[44], Adamax[44], averaged stochastic gradient descent (ASGD)[45], root mean square propagation (RMSprop)[46], root propagation (Rprop)[46], and stochastic gradient descent (SGD)[45]. For detailed analysis, each dataset is trained using each of the eight optimizers, resulting in eight different target models per dataset. This method allows us to evaluate the unique impact of each optimizer on the MEA susceptibility. For all datasets, the surrogate model uses a learning rate and batch size matching those established in the target model. Moreover, the optimizer of the surrogate model is set to Adam. Section 6.3 provides a detailed analysis of the conclusions about the optimizer's impact on the MEA.

5.3.4 Data annotation

We aim to answer the question: what is the impact of data annotation on MEAs? Accurate data annotation ensures that the model can learn from the correct information. If the data is labeled incorrectly or inconsistently, the model may learn the wrong pattern, resulting in performance degradation. In the evaluation of data annotation on the impact of the MEA, the training parameters of the target model are maintained consistently, while the training set of the target model is subject to variation. The training parameters, such as a batch size of 64, the Adam optimizer, and a MultiStepLR learning rate, remain consistent for both the target and surrogate models across different datasets. Central to our exploration is the measurement of the MEA susceptibility due to changes in label accuracy in the target model's training set. This method allows us to focus on analyzing how annotation accuracy impacts the model's robustness against the MEA. In this experiment, we employ two distinct methods to change the accuracy of labels in the training dataset of the target model: random modification and expert simulation. For the random modification method, the labels in the target model training set are changed in a non-systematic manner according to a predetermined ratio. Specifically, this means that the modification of labels is performed randomly, allowing for a certain degree of difference from the original ground truth labels. The process is gradual, starting from the initial modification of 1000 labels and continuing in steps of 1000 labels until a total of 25 000 labels have been modified. This method allows for the gradual introduction of randomness in the label set, effectively creating a diverse dataset that can impact model learning and performance. The random modification method reflects the possibility of random inaccuracies in label assignment in practical scenarios. Conversely, the expert simulation method is orchestrated to emulate the method of knowledgeable

labeling experts. Labeling experts have domain expertise that affects the accuracy of their labeling, and our strategy is designed to simulate these expert variances. Initially, we develop specialized models for each dataset as an alternative to traditional expert annotators. In the process of training these professional models, we carefully record the task accuracy at each stage, allowing us to replicate the range of proficiency typically exhibited by human experts. This simulation reflects the reality that experts have different levels of professional knowledge, which in turn affects their label accuracy. The experimental analysis of the impact of data cleaning on the MEA is described in Section 6.4.

5.3.5 Data selection

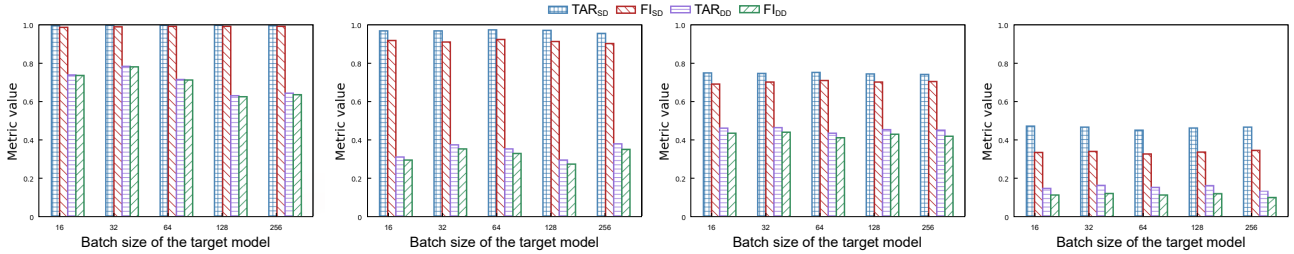
We aim to answer the question: what is the impact of data selection on MEAs? Data selection is a frequently used process in data processing, ensuring richness in the training data for machine learning models. The diversity and comprehensiveness of the datasets are crucial to building a robust and generalizable model. Rich data information can provide more perspectives and features, helping the model better understand all aspects of the problem and perform better when faced with novel or unseen data. In this experiment, both the target and surrogate models adhere to the training parameters previously established in Section 5.3.4 across diverse datasets, while introducing variations in the training set. We set up two schemes to explore the impact of data selection on the MEA, i.e., data volume (DV) and active learning (AL). Specifically, DV modulates the sample size in the target model's training set to facilitate its management. DV serves as a baseline for AL, evaluating whether datasets selected through AL significantly impact the MEA compared to randomly selected datasets of equal size. AL refers to technologies popularly used for data selection. It strategically submits challenging samples for expert labeling, fostering a gradual enhancement of the model's performance through iterative machine learning training. In this experiment, the ModAL framework[47] is utilized as the foundation for active learning. For the ModAL framework, the estimator used is set to a RandomForest classifier, the query strategy is set to entropy sampling, and the initial label sample size is set to 5000. Utilizing publicly available datasets that are already accompanied by ground-truth labels, our experiment predominantly focuses on employing AL methodologies for the selection of intricate samples to train target models. The experimental analysis of the impact of data selection on the MEA is described in Section 6.5.

6 Analysis and Result

6.1 Batch size

Figure 2 shows the impact of batch size on MEAs, where the x-coordinate denotes the batch size configured during the training of the target model and the y-coordinate represents the metric value. TAR is the proportion of the surrogate model's task accuracy relative to that of the target model, and FI stands for the fidelity of the surrogate model. Additionally, subscripts provide further clarification: SD indicates that the target and surrogate models' training sets share the same distribution, while DD signifies that their training sets come from different distributions.

Same distribution. In Fig. 2, our attention is primarily directed toward the bar chart of TAR_{SD} and FI_{SD}. Upon observation, it is evident that the surrogate model's performance, in terms of metrics TAR_{SD} and FI_{SD}, remains relatively stable despite the varying batch sizes of the target model. For the CIFAR100 dataset, the variance of TAR_{SD} is 0.006, and the variance of FI_{SD} is 0.005, respectively. Variance not only indicates how much the sample deviates from the mean but also reflects the extent of fluctuation among the samples. Therefore, a smaller variance denotes



(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 2 Impacts of the target model batch size on MEAs.

greater stability. For the MNIST, FMNIST, and CIFAR10 datasets, TAR_{SD} variances are 0.001, 0.005, and 0.003, respectively, and the variances of FI_{SD} are 0.001, 0.006, and 0.006, respectively.

Remark: Given that the choice of batch size does not significantly impact the effectiveness of MEAs when the adversary knows the distribution of the training dataset for the target model, defenders should consider strategies other than manipulating batch size to protect their model.

Different distribution. In this paragraph, we turn our attention to the trends in the TAR_{DD} and FI_{DD} annotations in Fig. 2. From the experimental statistical results, we can see that the variation trends of TAR_{DD} and FI_{DD} are inconsistent for different datasets. For the CIFAR100 dataset, the difference between the maximum and minimum values of TAR_{DD}, observed when the target model's batch size is set to 128 and 16, respectively, is 3.06%. For the CIFAR10 dataset, when the batch size of the target model is set to 16 and 64, the TAR_{DD} value reaches the minimum and maximum. For both MNIST and FMNIST datasets, the minimum value of TAR_{DD} occurs when the target model batch size is set to 32. However, for the MNIST and FMNIST datasets, the maximum values of TAR_{DD} appear when the target model batch size is set to 128 and 16, respectively. The trend of TAR_{DD} remains consistent with FI_{DD}.

Remark: When the adversary lacks knowledge about the target model's training dataset, the batch size of the target model significantly impacts MEAs, but the specific pattern of this variation has not yet been explored.

6.2 Learning rate

Figure 3 illustrates the impact of the learning rate on MEAs, with the horizontal axis denoting the learning rate mechanisms employed during the training of the target model, and the vertical axis depicting the corresponding metric values. Figure 3 includes annotations similar to those in Fig. 2. For a more streamlined presentation, we abbreviate the learning rates: ReduceLROnPlateau, StepLR, MultiStepLR, ExponentialLR, and CosineAnnealingLR are represented as RLR, SLR, MLR, ELR, and CLR, respectively.

Same distribution. In Fig. 3, there are inconsistencies in the TAP_{SD} and FI_{SD} trends of simple datasets, (MNIST and FMNIST) and complex datasets, (CIFAR10 and CIFAR100). For simple datasets, regardless of the learning rate mechanism adopted by the target model, the surrogate model stolen by the adversary shows very small fluctuations in TAP_{SD} and FI_{SD}. However, for complex datasets, the learning rate used by the target model significantly affects the MEA. The model with MRL as the learning rate is more resilient than the model with ERL as the learning rate. For example, from Fig. 3d, we can see that when the learning rate of the target model is set to ERL, its TAP_{SD} value is 52.23%, and FI_{SD} value is 31.6%. Moreover, when the learning rate of the target model is set to MRL, its TAP_{SD} and FI_{SD} values are 48.96% and 34.2%, respectively. For the CIFAR100 dataset, the differences between TAP_{SD} and FI_{SD} are 3.27% and 2.6%, respectively, when comparing models trained with ERL and MRL learning rates.

Remark: Using the MLR strategy enhances the confidentiality of the target model against MEAs for complex datasets, while the ELR increases susceptibility. To enhance model security, defenders should integrate MLR into their training protocols, particularly for models dealing with complex or high-dimensional data.

Different distribution. Unlike the scenario where the same distributions are used, when the adversary is unaware of the target model's training dataset

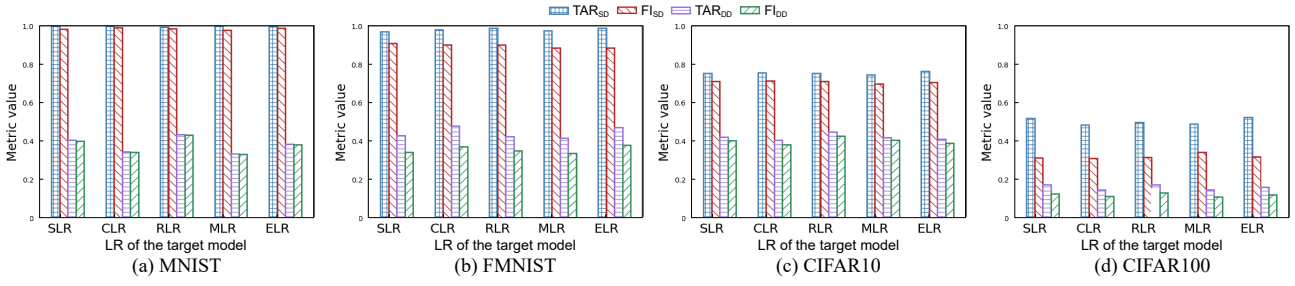


Fig. 3 Impact of the target model learning rate on MEAs. ReduceLROnPlateau, StepLR, MultiStepLR, ExponentialLR, and CosineAnnealingLR are represented as RLR, SLR, MLR, ELR, and CLR, respectively.

distribution, setting the model's learning rate to MLR minimizes the extent of model extraction by the adversary, regardless of the dataset type utilized. For the MNIST, FMNIST, and CIFAR100 datasets, when the learning rate of the target model is set to MLR, their TAR_{DD} values are 33.35%, 41.61%, and 14.62%, respectively. Similarly, the FI_{DD} values are 32.95%, 33.55%, and 10.9%, respectively. While this experiment allows for the identification of the safest learning rate, it does not yield a consistent conclusion regarding the most vulnerable learning rate. Because the susceptibility to attacks varies among different learning rates, depending on the dataset utilized. For example, when RLR is the learning rate of the target model for the CIFAR10 and MNIST datasets, the indices of TAR_{DD} and FI_{DD} stolen by the adversary reach their highest values. For other datasets, such as FMNIST and CIFAR100, the TAR_{DD} and FI_{DD} values peaked when the learning rate of the target model was set to CLR and SLR, respectively.

Remark: Regardless of the dataset used, the target model with MLR as its learning rate demonstrates the lowest susceptibility to adversaries in terms of model functionality stolen. This indicates that implementing MLR as a learning rate is beneficial for improving the confidentiality of the model.

6.3 Optimizer

Figure 4 shows the optimizer's impact on MEAs, where the horizontal axis represents the kind of optimizer set when training the target model and the vertical axis represents the metric value. The symbols in Fig. 4 have the same meaning as those in Fig. 2, as described in Section 6.1. For clarity, we abbreviate the optimizers Adadelta, Adagrad, Adam, Adamax, ASGD, RMSprop, Rprop, and stochastic gradient descent (SGD) as AA, AD, AM, AX, AS, RM, RP, and SG, respectively.

Same distribution. From TAP_{SD} and FI_{SD} in Fig. 4, we can see that regardless of the dataset, the target model with SG and AA as the optimizer is more susceptible to MEAs. For simple datasets (MNIST and FMNIST), the optimizers most vulnerable to the MEA are AA, RP, and SG. For complex datasets (CIFAR10 and CIFAR100), the optimizers most susceptible to the MEA are AA, AD, and SG. When the target model with SG as the optimizer is extracted by the MEA, TAP_{SD} values in different datasets (MNIST, FMNIST, CIFAR10, and CIFAR100) can reach 100%, 101%, 104%, 53.43%, respectively, while FI_{SD} values can reach 96.7%, 94.8%, 71.7%, and 35.05%. When extracting the target model of AA as the optimizer via the MEA, TAP_{SD} in different datasets (MNIST, FMNIST, CIFAR10, and CIFAR100) values can reach 100%, 98.94%, 94.84%, 55.17%, and the FI_{SD} value can reach 93.6%, 93%, 66.3%, and 33.2%. The TAP_{SD} value can exceed 100% when the task accuracy of the surrogate model surpasses that of the target model.

Remark: When the optimizer of the target model is set to SG or AA, the model becomes more vulnerable to replication through MEAs. This indicates that defenders should avoid using SG and AA as optimizers to protect the confidentiality of the model.

Different distribution. Compared with the experimental results of the same distribution, the different distribution also exhibits some analogous experimental phenomena. For the MNIST dataset, the optimizers showing the highest vulnerabilities are SG, AA, and RP, with TAR_{DD} values of 100%, 82.15%, and 73.69%, respectively. For the FMNIST dataset, models optimized using SG, AA, and AM are most prone to the MEA, with corresponding TAR_{DD} values of 46.75%, 42.82%, and 35.44%, respectively. For the CIFAR10 dataset, the optimizers AA, AD, and AX lead to the highest susceptibility, marked by TAR_{DD} values of 54.79%, 47.96%, and 42.54%, respectively.

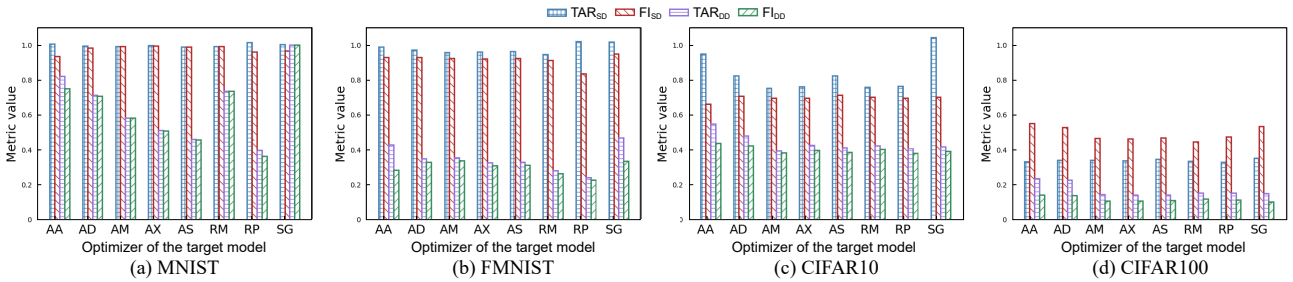


Fig. 4 Impact of the target model optimizer on MEAs. Adadelta, Adagrad, Adam, Adamax, ASGD, RMSprop, Rprop, and SGD are represented as AA, AD, AM, AX, AS, RM, RP, and SG, respectively.

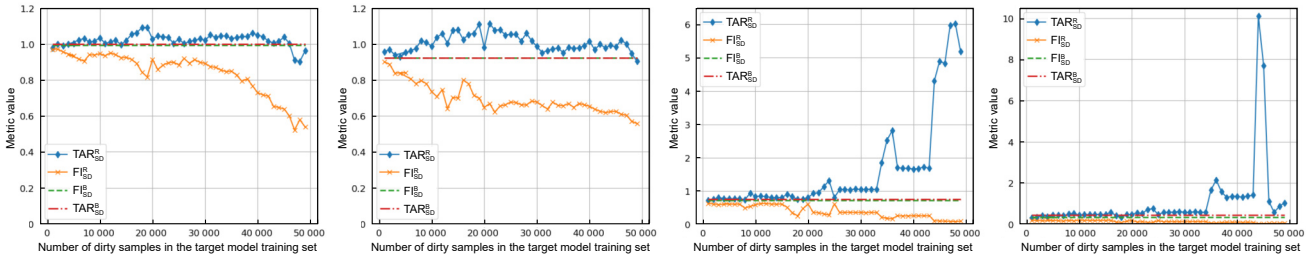
Meanwhile, for the CIFAR100 dataset, the top-3 optimizers marked by vulnerability are AA, AD, and RP, with respective $\mathrm{TAR}_{\mathrm{DD}}$ values of 23.54%, 22.71%, and 17.56%, respectively.

Remark: The AA optimizer consistently ranks as the most susceptible to MEAs across all evaluated datasets. This suggests that defenders should avoid using AA as an optimizer to protect the confidentiality of the model.

6.4 Data annotation

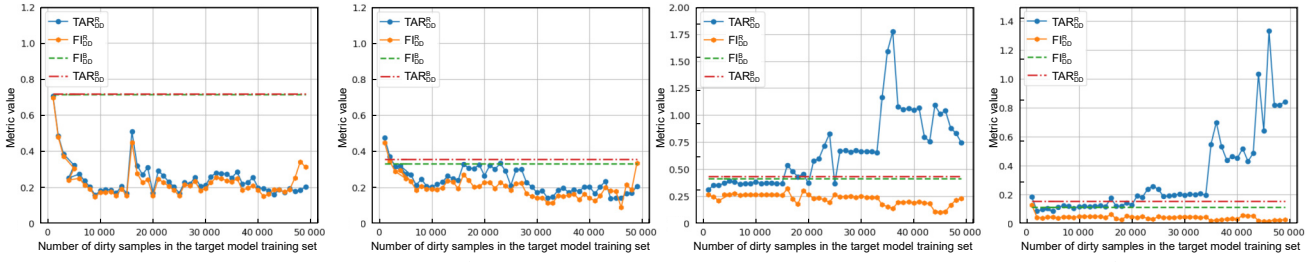
The experimental analysis of data annotation encompasses two pivotal facets: random modification and expert simulation, detailed in Section 5.3.4. In the visual representation of our experimental statistics, we employ superscripts R and E to distinguish between random modification and expert simulation methods, respectively. In Figs. 5–10, $\mathrm{TAP}^{\mathrm{B}}_{\mathrm{SD}}$ and $\mathrm{FI}^{\mathrm{B}}_{\mathrm{SD}}$ represent the baselines under the same distribution. The superscript B means baseline. The baselines $\mathrm{TAP}^{\mathrm{B}}_{\mathrm{SD}}$ and $\mathrm{FI}^{\mathrm{B}}_{\mathrm{SD}}$ correspond to scenarios where the target model training set is devoid of dirty samples. $\mathrm{TAR}^{\mathrm{B}}_{\mathrm{DD}}$ and $\mathrm{FI}^{\mathrm{B}}_{\mathrm{DD}}$ are the baselines for $\mathrm{TAR}_{\mathrm{DD}}$ and $\mathrm{FI}_{\mathrm{DD}}$, respectively, and they are interpreted similarly to $\mathrm{TAP}^{\mathrm{B}}_{\mathrm{SD}}$ and $\mathrm{FI}^{\mathrm{B}}_{\mathrm{SD}}$ but in the context of a different distribution.

Same distribution. For random modification method, it is evident that the presence of random dirty samples assists the adversary in extracting the task accuracy of the target model (see Fig. 5). Specifically, it is observed that in most cases $\mathrm{TAP}^{\mathrm{R}}_{\mathrm{SD}}$ is higher than $\mathrm{TAP}^{\mathrm{B}}_{\mathrm{SD}}$. However, the presence of random dirty samples in the target model does not necessarily benefit the adversary in extracting the fidelity of the target model. Based on the observations from Fig. 5, it is evident that across no matter what dataset it is, a consistent experimental phenomenon emerges, showing that $\mathrm{TAP}^{\mathrm{R}}_{\mathrm{SD}}$ exceeds $\mathrm{TAP}^{\mathrm{B}}_{\mathrm{SD}}$, while $\mathrm{FI}^{\mathrm{R}}_{\mathrm{SD}}$ remains less than $\mathrm{FI}^{\mathrm{B}}_{\mathrm{SD}}$.



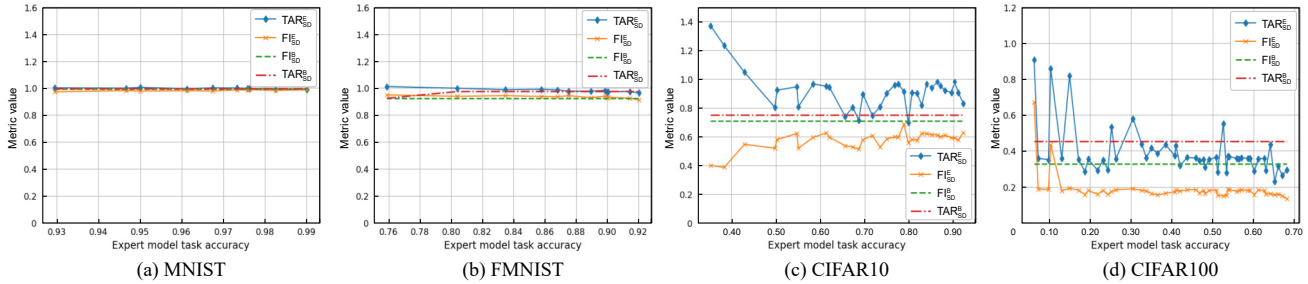
(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 5 Impact of target model data annotation on MEAs in the random modification method (same distribution).



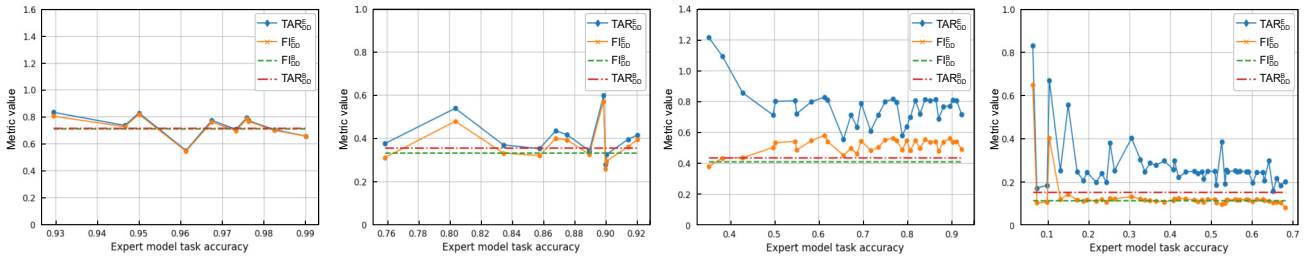
(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 6 Impact of target model data annotation on MEAs in the random modification method (different distribution).



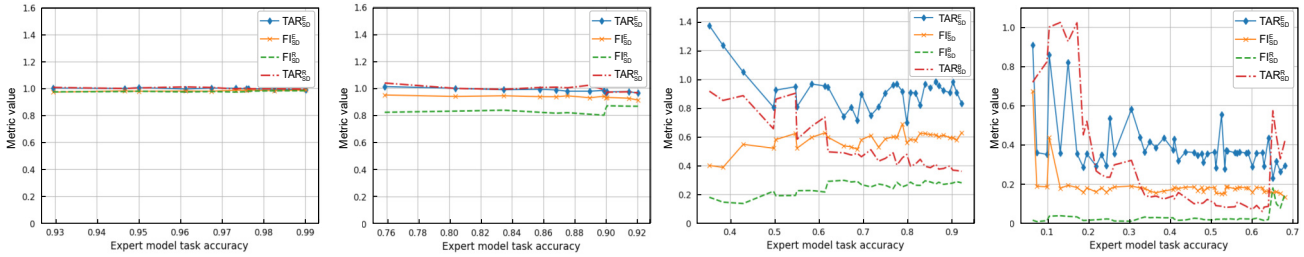
(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 7 Impact of target model data annotation on MEAs in the expert simulation method (same distribution).



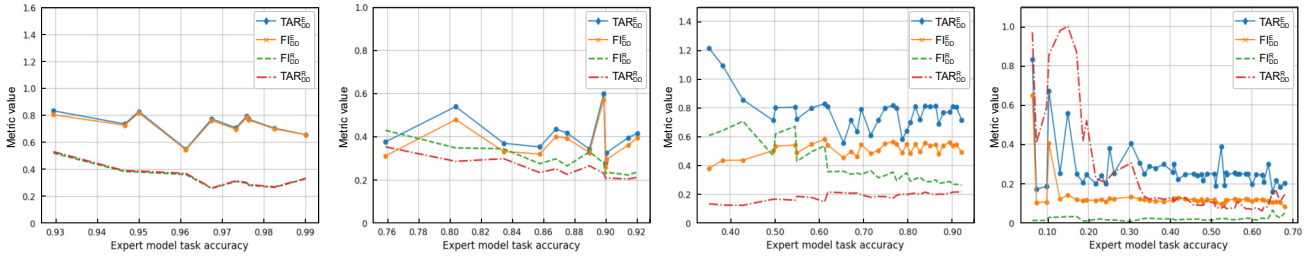
(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 8 Impact of target model data annotation on MEAs in the expert simulation method (different distribution).



(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 9 Comparison of the impact of target model data annotation on MEAs in random modification and the expert simulation methods (same distribution).



(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 10 Comparison of the impact of target model data annotation on MEAs in random modification and the expert simulation methods (different distribution).

For expert simulation method, we can clearly observe from Fig. 7 that different datasets yield diverse performances. For the MNIST and FMNIST datasets, there is no significant difference between TAP^{E}_{SD} and FI^{E}_{SD} from the baseline (TAP^{B}_{SD} and FI^{B}_{SD}). For CIFAR10 and CIFAR100 datasets, FI^{E}_{SD} has a lower value than FI^{B}_{SD}. This shows that mixing dirty samples into the target model training set is not conducive for the adversary to copy the decision behavior of the target model.

Figure 9 displays comparative statistics illustrating the influence of target model data annotation on MEAs in scenarios of random modification and expert simulation methods. To maintain a level playing field between random modification and expert simulation, an equal number of dirty samples are maintained in both setups. For the MNIST dataset, whether it is TAR or FI, the values of the random modification and expert simulation are basically consistent. For the FMNIST dataset, regardless of random modification and expert model, the values of TAR mostly overlap and remain consistent. However, from a fidelity perspective, FI^{E}_{SD} is higher than FI^{R}_{SD}. For the CIFAR10 and CIFAR100 datasets, we conclude that the expert simulation is more easily extracted by adversaries than the random modification, i.e., $TAP^{E}_{SD} > TAP^{R}_{SD}$ and $FI^{E}_{SD} > FI^{R}_{SD}$. By synthesizing the experimental data presented in Figs. 5, 7, and 9, we perform a comprehensive analysis focusing on fidelity. This led us to a consistent conclusion across Figs. 5, 7, and 9: FI^{R}_{SD} is less than FI^{B}_{SD}, FI^{E}_{SD} is less than FI^{B}_{SD}, and FI^{E}_{SD} is greater than FI^{R}_{SD}. In short, FI^{B}_{SD}, FI^{R}_{SD}, and FI^{E}_{SD} form a clear

relationship, i.e., $FI^{R}_{SD} < FI^{E}_{SD} < FI^{B}_{SD}$.

Remark: Regardless of which mislabeling strategy is present in the training set of the target model, the model tends to be more resistant to MEAs than using purely clean datasets. This indicates that introducing a certain degree of label noise or intentionally mislabeling during the training process may enhance the model's defense against MEAs.

Different distribution. For random modification method (see Fig. 6), simple datasets and complex datasets exhibit different conclusions in terms of metrics TAR and FI. For the MNIST and FMNIST datasets, TAR^{R}_{DD} and FI^{R}_{DD} are lower than baseline TAR^{B}_{DD} and FI^{B}_{DD}, respectively. For the CIFAR10 and CIFAR100 datasets, FI^{R}_{DD} is lower than baseline FI^{B}_{DD}. However, in terms of the performance of TAR^{R}_{DD}, a significant trend is observed: as the number of dirty samples in the dataset increases, the performance of TAR^{R}_{DD} changes significantly, initially slightly lower than TAR^{B}_{DD}, gradually exceeding TAR^{B}_{DD}, and finally becoming significantly more high. Overall, the presence of random dirty samples does not favor the adversary in replicating the decision behavior of the target model, i.e., it is not conducive to the adversary stealing a higher-fidelity surrogate model compared with using a pure clean training set.

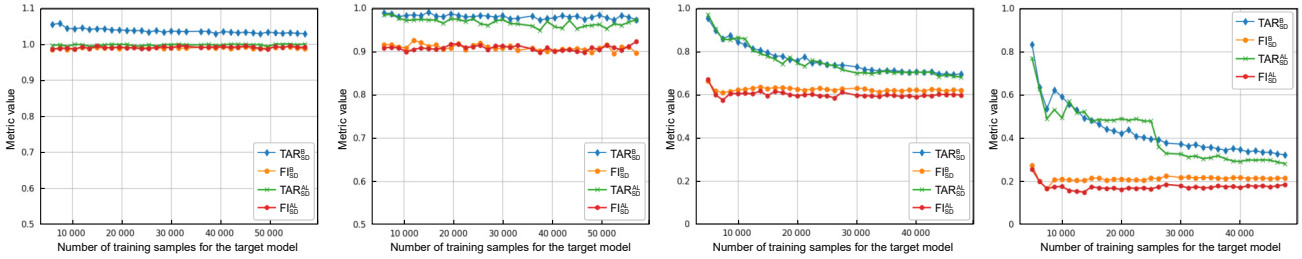
For expert simulation method (see Fig. 8), for the MNIST and FMNIST datasets, the indices TAR^{E}_{DD} and FI^{E}_{DD} demonstrate fluctuations without displaying any consistent pattern or trend. In the case of the CIFAR10 dataset, a detailed observation reveals a declining trend in both TAR^{E}_{DD} and FI^{E}_{DD} as the task accuracy of the expert model increases. However, even in this downward trend, these measures remain above their respective baseline values, indicating that these TAR^{E}_{DD} and FI^{E}_{DD} have a sustained advantage compared to the baseline. For the CIFAR100 dataset, as the task accuracy of the expert model enhances, both TAR^{E}_{DD} and FI^{E}_{DD} show a downward trend. Notably, FI^{E}_{DD} approaches the baseline FI^{B}_{DD} value, while TAR^{E}_{DD} remains higher than its baseline TAR^{B}_{DD} Under different distributions, it is evident that the presence of expert-induced dirty labels render the target model more vulnerable to the MEA compared to models trained exclusively with pure clean samples.

The comparative statistics of random modification and expert simulation methods are shown in Fig. 10. The experimental results under different distributions show trends analogous to those in the same distribution. In the analysis of the metric FI, it is evident that the expert simulation exhibits greater susceptibility to the MEA when compared to the random modification. Specifically, this vulnerability is reflected in the metric, where FI^{E}_{DD}, representing the expert simulation, is consistently higher than the random modification FI^{B}_{DD}. This trend suggests that the expert simulation may provide the MEA with more exploitable patterns than the random modification.

Remark: Models trained using dirty sample datasets obtained through expert simulation are more vulnerable to MEAs than models trained using purely clean datasets or randomly modified datasets. Defenders should be cautious about the source and nature of the data used for training.

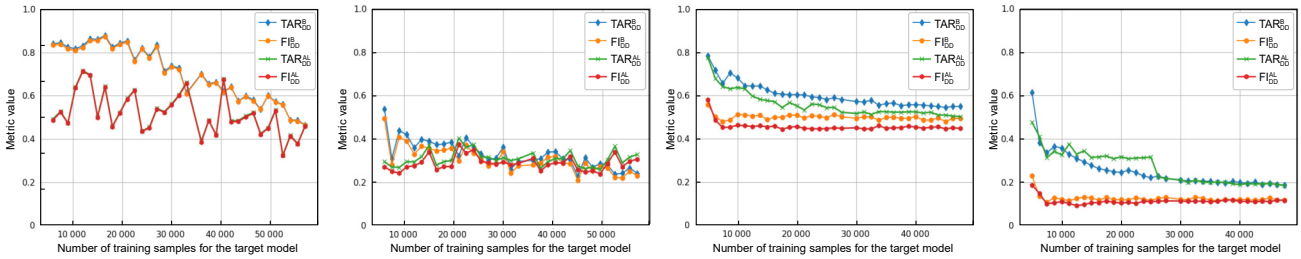
6.5 Data selection

Figures 11 and 12 show the impact of data selection on MEAs, with a comprehensive overview of the experimental design detailed in Section 5.3.5. In Figs. 11 and 12, the superscript B means baseline. The baseline denotes scenarios where the target model's training dataset has not undergone any selection by the data selection algorithm. Conversely, the superscript AL symbolizes instances where the target model's training dataset has been selected through the implementation of active learning algorithms. The subscripts SD and DD represent MEAs under the same



(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 11 Comparison of the impact of target model data selection on MEAs (same distribution).



(a) MNIST (b) FMNIST (c) CIFAR10 (d) CIFAR100

Fig. 12 Comparison of the impact of target model data selection on MEAs (different distribution).

distribution and different distribution settings, respectively.

Same distribution. From Fig. 11, it is evident that in both TAR and FI, the target model without data selection optimization is more susceptible to the MEA compared to the target model with dataset selection optimization. Especially in more complex datasets, this phenomenon is more pronounced. For example, for the CIFAR10 dataset, when the number of training samples for the target model is 21 250, the difference between $\mathrm{TAP}_{\mathrm{SD}}^{\mathrm{B}}$ and $\mathrm{TAP}_{\mathrm{SD}}^{\mathrm{AL}}$ reaches its maximum of 4.41%; meanwhile, the difference between $\mathrm{FI}_{\mathrm{SD}}^{\mathrm{B}}$ and $\mathrm{FI}_{\mathrm{SD}}^{\mathrm{AL}}$ also reaches 2%. For the CIFAR100 dataset, when the quantity of training samples utilized for the target model equals 10 000, the difference between $\mathrm{TAP}_{\mathrm{SD}}^{\mathrm{B}}$ and $\mathrm{TAP}_{\mathrm{SD}}^{\mathrm{AL}}$ reaches its peak at 9.69%, the difference between $\mathrm{FI}_{\mathrm{SD}}^{\mathrm{B}}$ and $\mathrm{FI}_{\mathrm{SD}}^{\mathrm{AL}}$ also reaches 3.32%.

Remark: Data selection enhances a model's resilience against MEAs, exhibiting particularly enhanced robustness in the context of complex datasets. This indicates that careful selection of training data plays a crucial role in enhancing the model's resistance to MEAs.

Different distribution. Similar to results under the same distribution, target models without data selection are more vulnerable to the MEA when compared to those with data selection, even under different distributions (see Fig. 12). For simple datasets, the gap between $\mathrm{TAR}_{\mathrm{DD}}^{\mathrm{B}}$ and $\mathrm{TAR}_{\mathrm{DD}}^{\mathrm{AL}}$, as well as between $\mathrm{FI}_{\mathrm{DD}}^{\mathrm{B}}$ and $\mathrm{FI}_{\mathrm{DD}}^{\mathrm{AL}}$, are relatively large. Especially for the MNIST dataset, the maximum difference between $\mathrm{TAR}_{\mathrm{DD}}^{\mathrm{B}}$ and $\mathrm{TAR}_{\mathrm{DD}}^{\mathrm{AL}}$ is 38.42% when the number of samples in the training set of the target model is 24 000. For complex datasets, the gap between $\mathrm{TAR}_{\mathrm{DD}}^{\mathrm{B}}$ and $\mathrm{TAR}_{\mathrm{DD}}^{\mathrm{AL}}$ is significantly smaller compared to the simple datasets. For example, for the CIFAR100 dataset, when the number of training samples for the target model is 5000, the difference between $\mathrm{TAR}_{\mathrm{DD}}^{\mathrm{B}}$ and $\mathrm{TAR}_{\mathrm{DD}}^{\mathrm{AL}}$ reaches its maximum of 13.77%.

Remark: The implementation of data selection algorithms notably diminishes the effectiveness of MEAs, thereby reducing their likelihood of successful exploitation. It is recommended that defenders use data selection algorithms as a fundamental component of model training to resist MEAs.

7 Conclusion

In this paper, we propose to evaluate the impact of MEAs from two key dimensions. Firstly, we explore the intrinsic impact of the target model, delving into how core components such as optimizers and learning rates influence MEAs. Secondly, from the perspective of data quality, we study how the accuracy and richness of data affect MEAs. Specifically, in evaluating the intrinsic parameters impact of the model, we conclude that the target model achieves the highest confidentiality with a MultiStepLR learning rate. Conversely, target models using Adadelta as the optimizer face a greater risk of being attacked by adversaries. Therefore, we recommend prioritizing the use of the MultiStepLR learning rate when defending against MEAs to enhance the confidentiality of the target model. In addition, the optimizer should be carefully selected to avoid using Adadelta, as it may increase the risk of the model being attacked. In terms of data quality, our results indicate that models trained with datasets containing dirty samples are less likely to be stolen by MEAs, compared to those trained with purely clean sample data, especially when the adversary is aware of the target model's training set distribution. Conversely, the richness of the training data significantly enhances the model's resilience against MEAs. Therefore, we recommend using a dataset containing dirty samples for training, which can reduce the risk of MEAs stealing the model. Secondly, ensuring the richness of training data will significantly

enhance the model's ability to resist MEAs. Through these two aspects of evaluation, we can gain a deeper understanding of the potential threats of MEAs and provide more targeted strategies and methods for strengthening the security of the model. In our future work, we plan to delve into the impact of multi-parameter interactions on MEAs. We will not only analyze how individual parameters affect the security of the model but also evaluate how different parameter combinations work together. We will also expand our research scope to include more diverse and more complex datasets to evaluate how these datasets affect the vulnerability of the model. Moreover, we are committed to developing a more detailed theoretical framework that will not only explain our current findings but also guide further research in this area. This work helps to promote the development of more effective defense strategies, ensuring the robustness and security of the model in the face of various threats.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62402131, 62132018, 62372130, 62102108, and 62372120), the Guangzhou Basic and Applied Basic Research Project (Nos. 2023A04J1725 and 2024A03J0397), Guangzhou Science and Technology Projects (No. 2025A03J3182), the Natural Science Foundation of Guangdong Province (No. 2022A1515010061), and the Postdoctoral Fellowship Program of CPSF (No. GZC20230595).

References

[1] A. Alzu'bi, A. Alomar, S. Alkhaza'leh, A. Abuarqoub, and M. Hammoudeh, A review of privacy and security of edge computing in smart healthcare systems: Issues, challenges, and research directions, *Tsinghua Science and Technology*, no. 4, pp. 1152–1180, 2024.

[2] M. A. Alghamdi, A. S. AL–Malaise AL–Ghamdi, and M. Ragab, Predicting energy consumption using stacked LSTM snapshot ensemble, *Big Data Mining and Analytics*, vol. 7, no. 2, pp. 247–270, 2024.

[3] P. Vaishnavi, K. Eykholt, and A. Rahmati, Transferring adversarial robustness through robust representation matching, in *Proc. 31st USENIX Conf. Security Symp.*, Boston, MA, USA, 2022, pp. 2083–2098.

[4] Y. Shen, X. He, Y. Han, and Y. Zhang, Model stealing attacks against inductive graph neural networks, in *Proc. 2022 IEEE Symp. Security and Privacy*, San Francisco, CA, USA, 2022, pp. 1175–1192.

[5] Y. Chen, C. Shen, C. Wang, and Y. Zhang, Teacher model fingerprinting attacks against transfer learning, in *Proc. 31st USENIX Conf. Security Symp.*, Boston, MA, USA, 2022, pp. 3593–3610.

[6] S. Zhou, T. Zhu, D. Ye, W. Zhou, and W. Zhao, Inversion-guided defense: Detecting model stealing attacks by output inverting, *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 4130–4145, 2024.

[7] J. Chen, J. Wang, T. Peng, Y. Sun, P. Cheng, S. Ji, X. Ma, B. Li, and D. Song, Copy, right? A testing framework for copyright protection of deep learning models, in *Proc. 2022 IEEE Symp. Security and Privacy*, San Francisco, CA, USA, 2022, pp. 824–841.

[8] M. Tang, A. Dai, L. DiValentin, A. Ding, A. Hass, N. Z. Gong, Y. Chen, and H. Li, ModelGuard: Information-theoretic defense against model extraction attacks, in *Proc. 33rd USENIX Conf. Security Symp.*, Philadelphia, PA, USA, 2024, p. 297.

[9] S. Kariyappa, A. Prakash, and M. K. Qureshi, MAZE: Data-free model stealing attack using zeroth-order gradient estimation, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021, pp. 13809–13818.

[10] A. Mutemi and F. Bacao, E-commerce fraud detection based on machine learning techniques: Systematic literature review, *Big Data Mining and Analytics*, vol. 7, no. 2, pp. 419–444, 2024.

[11] S. M. Tonni, D. Vatsalan, F. Farokhi, D. Kaafar, Z. Lu, and G. Tangari, Data and model dependencies of membership inference attack, arXiv preprint arXiv: 2002.06856, 2020.

[12] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang, M. Backes, E. De Cristofaro, M. Fritz, and Y. Zhang, ML-Doctor: Holistic risk assessment of inference attacks against machine learning models, in *Proc. 31st USENIX Conf. Security Symp.*, Boston, MA, USA, 2022, pp. 4525–4542.

[13] T. Nayan, Q. Guo, M. Al Duniawi, M. Botacin, S. Uluagac, and R. Sun, SoK: All you need to know about on-device ML model extraction - the gap between research and practice, in *Proc. 33rd USENIX Conf. Security Symp.*, Hiladelphia, PA, USA, 2024, p. 293.

[14] J. Rosenthal, E. Enouen, H. V. Pham, and L. Tan, DisGUIDE: Disagreement-guided data-free model extraction, in *Proc. 37th AAAI Conf. Artificial Intelligence*, Washington, DC, USA, 2023, pp. 9614–9622.

[15] Y. He, G. Meng, K. Chen, X. Hu, and J He, DRMI: A dataset reduction technology based on mutual information for black-box attacks, in *Proc. 30th USENIX Conf. Security Symp.*, 2021, pp. 1901–1918.

[16] Z. Lin, K. Xu, C. Fang, H. Zheng, A. A. Jaheezuddin, and J. Shi, QUDA: Query-limited data-free model extraction, in *Proc. 2023 ACM Asia Conf. Computer and Communications Security*, Melbourne, Australia, 2023, pp. 913–924.

[17] A. Yan, H. Yan, L. Hu, X. Liu, and T. Huang, Holistic implicit factor evaluation of model extraction attacks, *IEEE Trans. Depend. Secur. Comput.*, vol. 20, no. 6, pp. 4678–4689, 2023.

[18] S. Kaisar, M. M. Rashid, A. Chowdhury, S. S. Shafin, J. Kamruzzaman, and A. Diro, Enhancing telemarketing success using ensemble-based online machine learning,

Big Data Mining and Analytics, vol. 7, no. 2, pp. 294–314, 2024.

[19] B. Hanin and D. Rolnick, Deep ReLU networks have surprisingly few activation patterns, in *Proc. 33rd Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2019, p. 33.

[20] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, High accuracy and high fidelity extraction of neural networks, in *Proc. 29th USENIX Conf. Security Symp*., Virtual Event, 2020, p. 76.

[21] M. Yan, C. W. Fletcher, and J. Torrellas, Cache telepathy: Leveraging shared resource attacks to learn DNN architectures, in *Proc. 29th USENIX Conf. Security Symp*., Virtual Event, 2020, pp. 2003–2020.

[22] D. Rolnick and K. P. Kording, Reverse-engineering deep ReLU networks, in *Proc. 37th Int. Conf. Machine Learning*, Virtual Event, 2020, p. 757.

[23] N. Carlini, M. Jagielski, and I. Mironov, Cryptanalytic extraction of neural network models, in *Proc. 40th Annu. Int. Cryptology Conf. Advances in Cryptology*, Santa Barbara, CA, USA, 2020, pp. 189–218.

[24] I. A. Canales-Martínez, J. Chávez-Saab, A. Hambitzer, F. Rodríguez-Henríquez, N. Satpute and A. Shamir, Polynomial time cryptanalytic extraction of neural network models, in *Proc. 43rd Annu. Int. Conf. Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, Zurich, Switzerland, 2024, pp. 3–33.

[25] Y. Gao, H. Qiu, Z. Zhang, B. Wang, H. Ma, A. Abuadbba, M. Xue, A. Fu, and S. Nepal, DeepTheft: Stealing DNN model architectures through power side channel, in *Proc. 2024 IEEE Symp. Security and Privacy*, San Francisco, CA, USA, 2024, pp. 3311–3326.

[26] B. Wu, X. Yang, S. Pan, and X. Yuan, Model extraction attacks on graph neural networks: Taxonomy and realisation, in *Proc. 2022 ACM on Asia Conf. Computer and Communications Security*, Nagasaki, Japan, 2022, pp. 337–350.

[27] J. B. Truong, P. Maini, R. J. Walls, and N. Papernot, Data-free model extraction, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Virtual Event, 2021, pp. 4771–4780.

[28] Y. Zhuang, C. Shi, M. Zhang, J. Chen, L. Lyu, P. Zhou, and L. Sun, Unveiling the secrets without data: Can graph neural networks be exploited through data-free model extraction attacks? in *Proc. 33rd USENIX Conf. Security Symp*., Philadelphia, PA, USA, 2024, p. 294.

[29] T. Wu, W. Dou, F. Wu, S. Tang, C. Hu, and J. Chen, A deployment optimization scheme over multimedia big data for large-scale media streaming application, *ACM Trans. Multimedia Comput., Commun., Appl.* (*TOMM*), vol. 12, no. 5s, p. 73, 2016.

[30] L. Qi, C. Hu, X. Zhang, M. R. Khosravi, S. Sharma, S. Pang, and T. Wang, Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment, *IEEE Trans. Ind. Inf.*, vol. 17, no. 6, pp. 4159–4167, 2021.

[31] L. Qi, R. Wang, C. Hu, S. Li, Q. He, and X. Xu, Time-aware distributed service recommendation with privacy-preservation, *Inf. Sci.*, vol. 480, pp. 354–364, 2019.

[32] S. Xie, Y. Xue, Y. Zhu, and Z. Wang, Cost effective MLaaS federation: A combinatorial reinforcement learning approach, in *Proc. IEEE Conf. Computer Communications*, London, UK, 2022, pp. 2078–2087.

[33] L. Pajola and M. Conti, Fall of giants: How popular text-based MLaaS fall against a simple evasion attack, in *Proc. 2021 IEEE European Symp. Security and Privacy*, Vienna, Austria, 2021, pp. 198–211.

[34] S. Shukla, M. Alam, S. Bhattacharya, P. Mitra, and D. Mukhopadhyay, Whispering MLaaS: Exploiting timing channels to compromise user privacy in deep neural networks, *IACR Trans. Cryptogr. Hardware Embedded Syst.*, vol. 2023, no. 2, pp. 587–613, 2023.

[35] L. Xu and J. Zhai, DCVAE-adv: A universal adversarial example generation method for white and black box attacks, *Tsinghua Science and Technology*, vol. 29, no. 2, pp. 430–446, 2024.

[36] H. Liu, N. Li, H. Kou, S. Meng, and Q. Li, FSRPCL: Privacy-preserve federated social relationship prediction with contrastive learning, *Tsinghua Science and Technology*, vol. 30, no. 4, pp. 1762–1781, 2025.

[37] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, Enhanced ids with deep learning for iot-based smart cities security, *Tsinghua Science and Technology*, vol. 29, no. 4, pp. 929–947, 2024.

[38] S. Milli, L. Schmidt, A. D. Dragan, and M. Hardt, Model reconstruction from model explanations, in *Proc. Conf. Fairness, Accountability, and Transparency*, Atlanta, GA, USA, 2019, pp. 1–9.

[39] A. Bărbălău, A. Cosma, R. T. Ionescu, and M. Popescu, Black-box ripper: Copying black-box models using generative evolutionary algorithms, in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, p. 1689.

[40] Y. LeCun, Index of /exdb/mnist, http://yann.lecun.com/exdb/mnist/, 1998.

[41] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv: 1708.07747, 2017.

[42] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*. Toronto, Canada: University of Toronto, 2009.

[43] M. D. Zeiler, ADADELTA: An adaptive learning rate method, arXiv preprint arXiv: 1212.5701, 2012.

[44] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *Proc. 3rd Int. Conf. Learning Representations*, San Diego, CA, USA, 2015, pp. 1–15.

[45] R. Johnson and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in *Proc. 27th Int. Conf. Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2013, pp. 315–323.

[46] A. Graves, Generating sequences with recurrent neural networks, arXiv preprint arXiv: 1308.0850, 2013.

[47] T. Danka and P. Horvath, modAL: A modular active learning framework for Python, arXiv preprint arXiv: 1805.00979, 2018.

Anli Yan received the BS and MS degrees from University of Jinan, Jinan, China, in 2017 and 2020, respectively, and the PhD degree in cyberspace security from Hainan University, HaiKou, China, in 2023. She is a postdoctoral researcher at School of Mathematics and Information Science and School of Artificial Intelligence, Guangzhou University, Guangzhou, China. She is currently an associate professor at Hainan University, Haikou, China. Her research interests include AI security, explainable AI, data mining, and machine learning.

Lang Li received the PhD degree at Hainan University, Haikou, China, in 2025. He is currently working as a postdoctoral researcher at School of Artificial Intelligence, Guangzhou University, Guangzhou, China. His work aims to address the challenges associated with ensuring the security and privacy of AI systems, exploring innovative methods to efficiently remove data from machine learning models without compromising their integrity or performance.

Kanghua Mo received the BS degree from South China Agricultural University, Guangzhou, China, in 2017, and the MS degree in cyberspace security from Guangzhou University, Guangzhou, China, in 2022. He is currently pursuing the PhD degree in cyberspace security at Guangzhou University, Guangzhou, China. His research interests include AI security and deep reinforcement learning.

Peigen Ye received the BS degree in computer science from Central South University, Changsha, China, in 2019, and the MS degree in cyberspace security from Guangzhou University, Guangzhou, China, in 2022. He is currently pursuing the PhD degree in cyberspace security at Beijing Institute of Technology, Beijing, China. His primary research interests include AI security, model security, and deep reinforcement learning.

Li Hu received the master degree from Guangzhou University, Guangzhou, China, and the PhD degree with Guangzhou University, Guangzhou, China, in 2025. She is currently working as a postdoctoral researcher at the Hong Kong Polytechnic University, Hong Kong, China. Her research interests include cryptography and privacy in AI.

Shaowei Wang received the PhD degree from University of Science and Technology of China, Hefei, China, in 2019. He is currently an associate professor at School of Artificial Intelligence, Guangzhou University, Guangzhou, China. He has published over 30 papers in top-tier conferences and journals, such as VLDB, INFOCOM, IJCAI, ICDE, *IEEE TPDS*, and *IEEE TKDE*. His research interests are data privacy and federated learning.

Hongyang Yan received the PhD degree in computer science and technology from Nankai University, Tianjin, China, at 2019. She is currently an associate professor at Guangzhou University, Guangzhou, China. She has published more than 30 papers in international conferences and journals, including *Information Sciences*, *TOIT*, *FGCS*, *Applied Soft Computing*, etc. She has received best paper awards of international conferences such as CSS 2019 and ProvSec 2021. She is an academic editor of *International Journal of Intelligent Systems*. She also serves as program committee and publication chairs of many international conferences such as ML4CS 2022, DMBD 2021, and ML4CS 2020. Her research interests include information privacy in machine learning, federated learning, and cloud storage.

Jin Li received the BS degree in mathematics from Southwest University, Chongqing, China, in 2002, and the MS degree in mathematics and the PhD degree in information security from Sun Yat-sen University, Guangzhou, China, in 2004 and 2007, respectively. He is currently a professor at School of Artificial Intelligence, Guangzhou University, Guangzhou, China. He is a senior member of the IEEE. He has published more than 100 papers in international conferences and journals, including IEEE INFOCOM, *IEEE Transactions on Information Forensics and Security*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, and *ESORICS*. He also served as program chair and committee for many international conferences. His research interests include the design of secure protocols in cloud computing and cryptographic protocols.