RADEP: A Resilient Adaptive Defense Framework Against Model Extraction Attacks

Amit Chakraborty ‡, Sayyed Farid Ahamed*, Sandip Roy*§ , Soumya Banerjee*§,
Kevin Choi†, Abdul Rahman†, Alison Hu†, Edward Bowen†, Sachin Shetty*
‡Department of CSBS, Asansol Engineering College, Asansol, West Bengal 713305, India
idmeamit@gmail.com
*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
{kevchoi, abdulrahman, aehu, edbowen}@deloitte.com

***Abstract*—Machine Learning as a Service (MLaaS) enables users to leverage powerful machine learning models through cloud-based APIs, offering scalability and ease of deployment. However, these services are vulnerable to model extraction attacks, where adversaries repeatedly query the application programming interface (API) to reconstruct a functionally similar model, compromising intellectual property and security. Despite various defense strategies being proposed, many suffer from high computational costs, limited adaptability to evolving attack techniques, and a reduction in performance for legitimate users. In this paper, we introduce a Resilient Adaptive Defense Framework for Model Extraction Attack Protection (RADEP), a multifaceted defense framework designed to counteract model extraction attacks through a multi-layered security approach. RADEP employs progressive adversarial training to enhance model resilience against extraction attempts. Malicious query detection is achieved through a combination of uncertainty quantification and behavioral pattern analysis, effectively identifying adversarial queries. Furthermore, we develop an adaptive response mechanism that dynamically modifies query outputs based on their suspicion scores, reducing the utility of stolen models. Finally, ownership verification is enforced through embedded watermarking and backdoor triggers, enabling reliable identification of unauthorized model use. Experimental evaluations demonstrate that RADEP significantly reduces extraction success rates while maintaining high detection accuracy with minimal impact on legitimate queries. Extensive experiments show that RADEP effectively defends against model extraction attacks and remains resilient even against adaptive adversaries, making it a reliable security framework for MLaaS models.**

***Index Terms*—Model extraction attack, Machine-Learning-as-a-Service (MLaaS), Deep learning, Malicious query, Security.**

I. Introduction

With the growing popularity of MLaaS, advanced models are now easily accessible via cloud-based Application Program Interface (API)s, bypassing the need for local training but also introducing new vulnerabilities. Recent studies have shown that these models are prone to extraction attacks [1], where adversaries repeatedly query the victim model using carefully crafted or surrogate inputs to reconstruct a substitute model that mimics its functionality. This unauthorized replication not only jeopardizes intellectual property but can also facilitate further adversarial exploits and compromise user privacy. Machine learning models developed and deployed for various critical infrastructure applications are increasingly targeted by adversarial threats, including ME attack [2], [3]. Adversaries may employ both black-box (without internal insight) methods (such as the JBDA-TR [1], Cloudleak [4], KnockoffNet [5], Zeroth Order Optimization [6] etc.), and many white-box (transparent or accessible model) techniques to perform these attacks. Although defenses like adversarial training [7], model pruning [8], and query detection [7] have been proposed, they incur significant computational overhead. Another common challenge with model extraction defenses is the trade-off between security and performance, where strict security measures can degrade the accuracy of the source model [9], [10]. Moreover, static defenses often fail against adaptive adversaries who modify their strategies to bypass security measures [7]. MLaaS platforms implement strong authentication mechanisms, such as API keys, OAuth, and multi-factor authentication, to restrict model access to authorized users [11]. However, an authenticated adversary can still execute model extraction by issuing an excessive number of queries. This necessitates a strong research focus on developing a enhanced defense technique to protect MLaaS models while ensuring their integrity and confidentiality.

To address these challenges, we propose RADEP, a multi-layered defense framework that integrates complementary techniques to protect MLaaS models from extraction and privacy attacks. RADEP combines *progressive adversarial training* for resilience against evolving threats, *malicious query detection* using uncertainty and behavioral analysis, and an *adaptive query response* mechanism that perturbs suspicious queries while preserving utility for legitimate users. It also includes *ownership verification* via embedded backdoor triggers and lightweight watermarking, enabling reliable detection of unauthorized model usage. The main contributions of this paper are as follows:

- We propose RADEP, a multifaceted defense framework against model extraction attacks, integrating progressive adversarial training, malicious query detection, adaptive query response, and ownership verification. Through detailed analysis and experiments, we demonstrate how RADEP effectively reduces the success of extraction attacks and limits the utility of stolen models.

- We introduce a query detection system that uses uncertainty metrics and behavioral analysis to potentially malicious queries, and a dynamic response mechanism that degrades adversarial outputs.
- We ensure malicious query detection time to be less than 0.01 ms and adaptive query response times between 15 ms (MNIST) and 60 ms (ImageNette), with ownership verification completed within 520.5 to 850.5 ms. This low overhead enhances scalability, making RADEP suitable for deployment in resource-constrained MLaaS environments.
- We conduct extensive experiments to evaluate RADEP's effectiveness against state-of-the-art model extraction attacks, including *JBDA-TR* [1], *Cloudleak* [4], and *KnockoffNet* [5]. The results demonstrate RADEP's superior resilience across different datasets and attack scenarios.

The remaining part of this paper is as follows: Section II outlines the threat model, detailing the attacker's objectives, knowledge, and capabilities. Section III describes the proposed RADEP framework. Section IV 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 V.

II. Threat Model

In this section, we outline the threat model, detailing the adversary's objective, knowledge and strategy of the proposed model extraction attack defense framework [7].

A. Adversary Objective

Model stealing attacks typically aim to replicate various aspects of a target model, such as its architecture, hyperparameters, or overall functionality [7]. In this paper, we focus specifically on functionality stealing, where the attacker's goal is to build a substitute model that closely matches the performance of the victim model. To assess the success of such attacks, we use two metrics: (i) test accuracy, which measures how well the substitute model performs on the victim model's test data, and (ii) fidelity, defined as the level of agreement between the outputs of the victim and the extracted model on identical inputs. An ME attacker might be primarily interested in replicating the victim model to avoid ongoing API costs, or might use a successful extraction to facilitate further attacks, such as adversarial or membership inference attacks.

B. Adversary Knowledge

We assume that the adversary has limited access to data and can only interact with the victim model via a black-box (without internal insight) interface [12]. In this context, "data-limited" implies that the attacker only has access to a small set of natural samples, while "black-box access" (without internal insight) means that the attacker can only submit inputs and observe the corresponding outputs from the victim model. Based on the type of outputs received, ME attacks can be categorized into two scenarios: (i) the hard-label scenario, where only the predicted class is returned, and (ii) the soft-label scenario, where the full probability distribution is provided. In our evaluations, we test our defense framework under both of these conditions.

C. Adversary Strategy

The adversary, limited by a small number of natural samples, overcomes data scarcity by either generating synthetic data or by employing surrogate data to query the victim model. The resulting query-output pairs are then used to train a substitute model. In our evaluation, we consider three advanced attack strategies. First, in the *JBDA-TR* attack, an enhanced version of Jacobian-based dataset augmentation is employed [1]. For each sample in the training set, synthetic samples are iteratively generated using a targeted variant of the Fast Gradient Sign Method (FGSM) [13], where a random target class is selected in each iteration. The synthetic samples produced are then labeled by the victim model and added to the training data to retrain the substitute model. Second, the *Cloudleak* [4] attack similarly relies on synthetic sample generation but differs by using a feature-based adversarial attack to create these samples; the adversary subsequently fine-tunes a pre-trained substitute model with the newly obtained labeled data. Finally, the KnockoffNet attack [5] bypasses synthetic sample generation by querying the victim model with surrogate data from the same or a related distribution and training the substitute model on the responses. These strategies collectively enable the adversary to effectively replicate the victim model's functionality despite having only limited natural data.

III. Proposed RADEP

This section provides a detailed description of each phase of the proposed RADEP framework. Figure 1 illustrates the architecture, highlighting the interconnection of each phase.

A. Progressive Adversarial Training

Adversarial training is a key defense mechanism that improves model resilience by exposing the model to diverse adversarial perturbations during training. Our approach integrates multiple adversarial techniques, including Fast Gradient Sign Method (FGSM) [13], Projected Gradient Descent (PGD) [14], and DeepFool [15], to generate adversarial examples that challenge the model's decision boundaries. For example, FGSM perturbs an input x in the direction of the gradient of the loss function $J(\theta, x, y)$ as follows:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)) \tag{1}$$

where ϵ is a small perturbation factor, θ represents the model parameters, and $J(\theta, x, y)$ is the loss function. PGD extends this concept by applying iterative updates to refine the adversarial example, while DeepFool minimizes the l_2-norm to produce more precise perturbations [1]. In addition to these methods, our approach incorporates adaptive training updates that periodically generate and integrate new adversarial examples into the training process. This continuous update mechanism allows the model to adapt to emerging threats without requiring complete retraining, thereby enhancing its resilience against increasingly sophisticated extraction attacks.

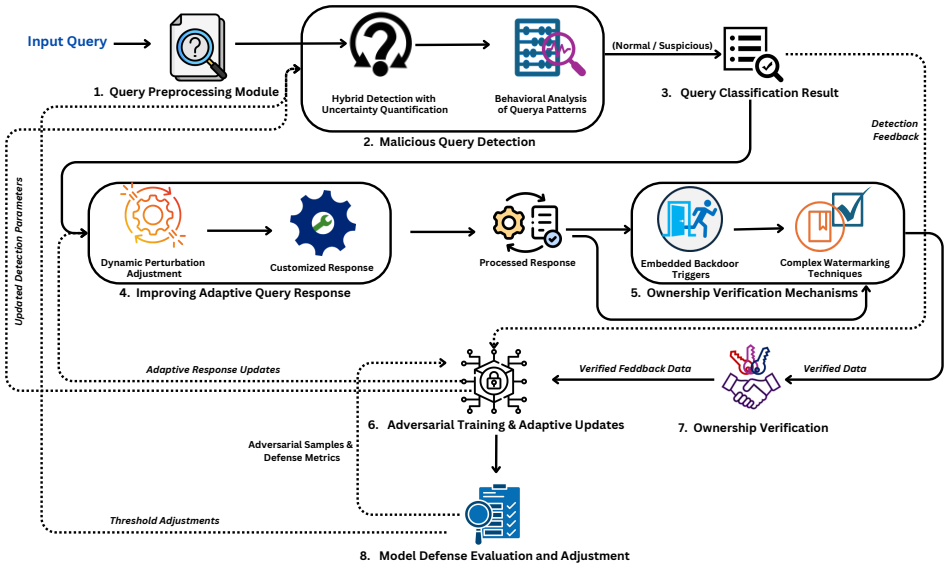

Fig. 1: Architecture of the proposed framework RADEP.

B. Malicious Query Detection Mechanisms

The proposed malicious query detection mechanism combines uncertainty quantification and behavioral analysis to flag malicious queries while preserving normal user experience. In the uncertainty component, a composite score is computed as:

$$U(x) = \alpha_1(1-P_{\max}(x))+\alpha_2 H(x)+\alpha_3(1-M(x))+\alpha_4\sigma(x) \quad (2)$$

where $P_{\max}(x)$ refers to the maximum softmax probability [16]), and the entropy $H(x)$ is computed as:

$$H(x) = -\sum_{i=1}^{K} P(y=i \mid x,\theta)\log P(y=i \mid x,\theta) \quad (3)$$

where $M(x)$ is the margin between the top two predictions, and $\sigma(x)$ is the Bayesian-based uncertainty [17] (e.g., via Monte Carlo Dropout). The weights α_i are calibrated using validation data, and queries with $U(x) > \tau$ are flagged as suspicious.

On the other side, the behavioral analysis segments queries into fixed intervals that measure query frequency and variance. Abnormal query rates or unusually low variance, identified using statistical methods such as Kullback-Leibler (KL) divergence [18], indicate potential extraction attempts. Combined, these two approaches create a resilient detection system with high accuracy and minimal false positives.

C. Adaptive Query Response

Adaptive query response adjusts output perturbations according to the evaluated suspicion level of each query, as outlined in Algorithm 1. In our approach, each query is assigned a suspicion score $S(q_i)$ (as shown in line 2 of Algorithm 1), computed through different uncertainty metrics such as softmax probability $P_{\max}(q)$, predicted entropy $H(q)$, margin of top two predictions $M(q)$, and Bayesian-based uncertainty $\sigma(q)$.

Based on this score, the system dynamically adjusts the perturbation level applied to the query response. Higher suspicion results in stronger perturbations through techniques such as label flipping and additive noise [19], while highly suspicious queries undergo adaptive label scaling that redistributes output probabilities to obscure the model's behavior. The perturbed response is calculated (Algorithm 1 line 9), ensuring that responses are customized for different levels of query suspicion. RADEP customizes responses by comparing incoming queries to previously flagged ones, degrading outputs for suspicious patterns. Periodic threshold recalibration ensures adaptability to evolving attacks with minimal impact on legitimate users.

Algorithm 1 Adaptive Query Response

Require: Set of queries $q_1, \ldots, q_N$
Ensure: Perturbed responses for each query
1: **for** each query q_i **do**
2: *Compute:* $S(q_i) = \alpha_1(1 - P_{\max}) + \alpha_2 H(q_i) + \alpha_3(1 - M(q_i)) + \alpha_4\sigma(q_i)$ ▷ *suspicion score*
3: $\epsilon = \begin{cases} \epsilon_{\text{low}} & \text{if } S(q_i) \le \tau_1 \\ \epsilon_{\text{medium}} & \text{if } \tau_1 < S(q_i) \le \tau_2 \\ \epsilon_{\text{high}} & \text{if } S(q_i) > \tau_2 \end{cases}$
4: $P_{\text{pert}}(y \mid q_i) = P(y \mid q_i) + \delta(S(q_i)), \quad \delta(S(q_i)) \sim \mathcal{N}(0, \epsilon)$ ▷ *Perturbed responses*
5: **end for**
6: **return** Perturbed responses $P_{\text{pert}}(y \mid q_i)$.

D. Ownership Verification Mechanisms

To confirm model ownership, we embed distinctive signatures into the model through both backdoor triggers [20] and complex watermarking techniques [8]. In the backdoor trigger approach, specific trigger queries are crafted to produce distinct, predetermined responses that serve as a digital signature. The model is trained so that for each trigger query q^i_{trigger}, it produces a designated output y^i_{trigger}, confirming that if a suspected model returns these outputs, its origin can be verified. These triggers are carefully isolated from the normal input distribution to remain resilient against modifications like pruning or fine-tuning [8].

Complementing this, complex watermarking techniques are used to subtly modify the model's standard outputs, em-

bedding a distributed signature that persists across typical queries. For example, the output probability distribution for a watermark query q_w is adjusted as

$$P_{\text{watermark}}(y \mid q_w) = P(y \mid q_w) + \epsilon(y, q_w) \tag{4}$$

where $\epsilon(y, q_w)$ is a small, query-dependent perturbation that forms the watermark signature. This combined strategy of backdoor triggers and watermarking creates a dual-layered verification mechanism, providing resilient evidence of ownership even after the model undergoes adversarial modifications.

E. Regular Evaluation and Adjustment

We implement an automated attack simulation that periodically evaluates the defense system. Various adversarial attacks, such as FGSM [13], PGD [14], and Zeroth Order Optimization [6], are simulated to generate adversarial examples. The system logs the attack success rate by comparing misclassified examples with clean data, and monitors the false positive rate to minimize impact on legitimate queries. Successful adversarial examples are added to the training set, and detection thresholds are adjusted if the attack success rate exceeds a set tolerance. This feedback loop continuously refines adversarial training, query detection, and response strategies, maintaining high performance while ensuring resilience against extraction attacks.

IV. Experimental Evaluation

In this section, we first describe our experimental setup, then analyze the impact of adversarial training on model extraction attacks. Next, we compare our detection method with Out of Distribution (OOD) detection [16] and evaluate the overall effectiveness of RADEP, using Dynamic Adversarial Watermarking of Neural networks (DAWN) [20], deceptive perturbation [21], adaptive misinformation [16], and AMAO [7] as baselines.

A. Experimental Setup

1) Datasets and Victim Models: We consider LeNet-5 for MNIST, AlexNet for FMNIST, ResNet18 for CIFAR-10, and ResNet34 for ImageNette. These datasets collectively offer varying levels of image complexity and dataset sizes, enabling a comprehensive evaluation of RADEP's effectiveness under diverse experimental settings.

2) Attack Configuration: To strengthen the attack setting and evaluate our defense against a more capable adversary, we configure the substitute model to mirror the victim model's architecture. This design ensures that if RADEP proves effective under such stringent conditions, its resilience would be even greater when the adversary lacks this knowledge.

In *JBDA-TR* [1] and *Cloudleak* [4], the substitute model begins training with 100, 100, 1,000, and 1,000 samples for MNIST, F-MNIST, CIFAR-10, and ImageNette, respectively. The corresponding query budgets for these datasets are 10,000, 10,000, 100,000, and 100,000. JBDA-TR involves six iterative augmentation rounds, with the substitute model being trained for 20 epochs after each round. Alternatively, *Cloudleak* employs a pre-trained substitute model that is fine-tuned for 20 epochs to enhance attack performance.

For *KnockoffNet* [5], surrogate datasets are leveraged for training the substitute models: FashionMNIST for MNIST, MNIST for F-MNIST, CIFAR-100 for CIFAR-10, and ImageNet for ImageNette. The assigned query budgets are 60,000, 60,000, 50,000, and 13,000 for the respective datasets. Each substitute model is trained for a total of 50 epochs to improve extraction accuracy.

Our primary goal is to assess the robustness of RADEP against a range of model extraction attacks rather than performing a comparative analysis among the attacks themselves. Although the query budgets vary across different attack strategies due to their unique characteristics, this variation does not affect the fairness of our defense evaluation.

3) Metrics: To evaluate RADEP's performance, we consider test accuracy, fidelity, and watermark accuracy as primary metrics, focusing on both defense effectiveness and model robustness. Additionally, we analyze computational overhead to ensure practical applicability. Since fidelity closely aligns with test accuracy trends, we exclude its detailed results for brevity.

4) Comparison with Existing Defenses: We evaluate RADEP against five state-of-the-art defense strategies to comprehensively measure its effectiveness in mitigating model extraction attacks. PRADA [1] identifies malicious queries by analyzing Gaussian distance distribution deviations, based on the observation that synthetic queries often cluster around seed samples. OOD Detection [16] classifies adversarial inputs as out-of-distribution using tailored detection methods. DAWN [20] introduces a hashing mechanism that perturbs predicted labels, limiting extraction performance while embedding backdoors for ownership verification. Deceptive Perturbation [21] alters probability vectors to mislead attackers without affecting the original classification. AMAO [7] presents an end-to-end defense framework that protects the model throughout its lifecycle, from training to deployment. Adaptive Misinformation [16] degrades stolen model accuracy by generating incorrect outputs through a reverse model trained with reverse cross-entropy loss.

While these defenses mainly concentrate on query detection or response alteration, RADEP advances beyond them by integrating adversarial training with a hybrid query detection framework. Furthermore, it enhances ownership verification using dual-layer techniques, combining watermarking and backdoor triggers to ensure both robust defense against extraction and reliable tracing of unauthorized model usage.

B. Evaluations on Adversarial Training

We perform experimental evaluations against *JBDA-TR*, *KnockoffNet*, and *Cloudleak* attacks, with the corresponding results summarized in Table I. The extraction models trained adversarially using RADEP consistently achieve lower accuracy compared to standard models, demonstrating the effectiveness of RADEP's defense mechanisms in both hard-label and soft-label attack settings. In the hard-label scenario, where adversary information is limited, RADEP significantly restricts model extraction. Even in the soft-label setting, RADEP effectively hinders the adversary's ability to capture decision boundary information. These findings confirm

TABLE I: The performance of the substitute model under the defense of adversarial training, where the most effecgive results are highlighted.

Attack	Scenario	Dataset	Std. train	Adv. train
JBDA-TR	Hard	MNIST	91.23	88.93
		F-MNIST	79.33	77.87
		CIFAR-10	42.80	42.05
		ImageNette	51.27	47.22
	Soft	MNIST	95.58	91.89
		F-MNIST	81.44	82.51
		CIFAR-10	43.97	43.14
		ImageNette	55.76	54.86

TABLE II: Accuracy and F1-Score comparisons demonstrate that RADEP, highlighted, outperforms OOD detection [16] and AMAO [7].

Dataset	Attack	Accuracy (%) [16] / [7] / RADEP	F1-Score [16] / [7] / RADEP
MNIST	*JBDA-TR*	76.15 / 87.85 / 90.57	0.68 / 0.87 / 0.91
	KnockoffNet	86.20 / 93.75 / 95.86	0.82 / 0.93 / 0.96
	Cloudleak	72.23 / 84.99 / 87.31	0.61 / 0.84 / 0.89
F-MNIST	*JBDA-TR*	73.05 / 81.20 / 83.73	0.66 / 0.81 / 0.85
	KnockoffNet	65.91 / 69.89 / 72.54	0.53 / 0.67 / 0.73
	Cloudleak	66.33 / 74.37 / 76.82	0.54 / 0.73 / 0.78
CIFAR-10	*JBDA-TR*	73.41 / 78.99 / 81.90	0.67 / 0.76 / 0.82
	KnockoffNet	84.55 / 86.23 / 88.85	0.83 / 0.87 / 0.91
	Cloudleak	77.04 / 77.33 / 80.28	0.73 / 0.76 / 0.80
ImageNette	*JBDA-TR*	77.11 / 80.29 / 83.43	0.80 / 0.82 / 0.86
	KnockoffNet	72.58 / 81.13 / 84.57	0.76 / 0.81 / 0.85
	Cloudleak	77.16 / 83.04 / 86.09	0.80 / 0.84 / 0.88

that RADEP's progressive adversarial training and adaptive strategies increase the adversary's query requirements, elevate computational costs, and enhance overall model robustness against extraction attacks.

C. Evaluation on Malicious Query Detection

RADEP demonstrates superior performance in detecting adversarial queries compared to *AMAO* [7] and *OOD detection* [16] approaches. It utilizes a hybrid detection framework that integrates uncertainty quantification with behavioral analysis, enhancing the robustness of its defense. As shown in Table II, RADEP substantially improves both accuracy and F1-score in detecting malicious queries. For instance, against *KnockoffNet* attacks on MNIST, RADEP attains 95.86% accuracy and an F1-score of 0.96, outperforming *AMAO* and *OOD detection*. Moreover, RADEP captures temporal patterns in query behavior, including frequency and variance, which helps detect advanced attacks like *Cloudleak*. On MNIST, RADEP achieves 87.31% detection accuracy, exceeding the performance of both *AMAO* and *OOD detection*. These results highlight RADEP's capability to efficiently handle adaptive attacks. Additionally, RADEP incorporates an adaptive response mechanism that dynamically alters detection thresholds and response strategies based on the suspicion level of queries, significantly outperforming static defenses and consistently enhancing performance across diverse attacks and datasets.

TABLE III: The test accuracy of the substitute model under the defense of RADEP and the baseline defenses [16], [21], [20], where RADEP outperforms the baselines is highlighted.

Attack	Scenario	Dataset	No Defense	Baseline Defense	RADEP
JBDA-TR	Hard Label	MNIST	91.23	87.30	65.17
		F-MNIST	79.33	74.21	63.52
		CIFAR-10	42.80	35.35	30.89
		ImageNette	51.27	47.88	43.90
	Soft Label	MNIST	95.58	91.80	78.94
		F-MNIST	81.44	75.48	59.78
		CIFAR-10	43.97	40.97	34.53
		ImageNette	55.76	50.70	46.35
KnockoffNet	Hard Label	MNIST	89.57	70.44	65.51
		F-MNIST	40.38	34.95	30.76
		CIFAR-10	69.37	63.48	47.07
		ImageNette	55.90	50.88	42.54
	Soft Label	MNIST	91.72	80.56	77.92
		F-MNIST	42.10	37.81	32.18
		CIFAR-10	73.02	71.35	68.59
		ImageNette	68.18	61.45	55.33
Cloudleak	Hard Label	MNIST	83.72	73.14	65.51
		F-MNIST	76.07	67.82	61.45
		CIFAR-10	78.15	67.59	58.86
		ImageNette	86.64	73.60	66.12
	Soft Label	MNIST	86.36	75.93	71.10
		F-MNIST	78.26	71.33	62.88
		CIFAR-10	80.04	71.09	65.23
		ImageNette	88.19	78.10	69.97

D. Overall Evaluations on RADEP from End to End

In this section, we comprehensively evaluate RADEP against *JBDA-TR*, *KnockoffNet*, and *Cloudleak* attacks across multiple datasets.

1) Effectiveness of RADEP: Experimental results indicate that substitute models extracted from adversarially trained victims under RADEP consistently attain lower test accuracy than those from standard trained models. For instance, under the JBDA-TR (Hard Label) attack on MNIST, the accuracy of the substitute model reduces to 65.17%, while under Cloudleak on CIFAR-10, it declines to 58.86%. These findings validate the effectiveness of RADEP's progressive adversarial training with periodic adaptive updates in defending against iterative attacks. As illustrated in Table **??**, RADEP outperforms existing defenses, such as DAWN [16] for hard-label scenarios and Deceptive Perturbation [21] and Adaptive Misinformation [20] for soft-label scenarios, demonstrating significantly reduced attack success rates across the considered model extraction attacks.

Moreover, RADEP's hybrid query detection combined with dynamic response strategies further minimizes the impact of adversarial queries. For example, under the *KnockoffNet* (Soft Label) attack on ImageNette, the substitute model's accuracy drops to 55.33%, while under *Cloudleak* on F-MNIST, it falls

TABLE IV: Computational Overhead of RADEP.

Dataset	Phase	Overhead
MNIST	Adversarial training	8.50 (min)
	Malicious query detection	< 0.01 (ms)
	Adaptive query response	15.00 (ms)
	Ownership verification	520.50 (ms)
F-MNIST	Adversarial training	14.20 (min)
	Malicious query detection	< 0.01 (ms)
	Adaptive query response	18.90 (ms)
	Ownership verification	550.80 (ms)
CIFAR-10	Adversarial training	85.00 (min)
	Malicious query detection	< 0.01 (ms)
	Adaptive query response	32.80 (ms)
	Ownership verification	710.40 (ms)
ImageNette	Adversarial training	200.00 (min)
	Malicious query detection	< 0.01 (ms)
	Adaptive query response	60.00 (ms)
	Ownership verification	850.50 (ms)

to 61.45%. This adaptive perturbation mechanism disrupts the attack process by making query-based extraction substantially more difficult for adversaries.

Furthermore, RADEP's ownership verification mechanisms offer an additional layer of protection by degrading the accuracy of extracted models. Specifically, under the *JBDA-TR* (Soft Label) attack on CIFAR-10, the substitute model's accuracy further declines to 34.53%, while for the *KnockoffNet* (Hard Label) attack on F-MNIST, it drops to 30.76%. These evaluations collectively demonstrate RADEP's capability to significantly weaken model extraction attempts while preserving data privacy.

2) Computational Overhead of RADEP: Table IV summarizes the experimental results obtained using a high-performance system with an Intel Xeon processor and NVIDIA A100 GPUs, running Ubuntu 20.04 LTS with TensorFlow and PyTorch utilizing CUDA acceleration. Performance metrics were recorded using Python's time module and framework profilers, verifying RADEP's stability and efficiency in real-time scenarios.

RADEP achieves malicious query detection within less than 0.01ms and adaptive query response times ranging between 15ms and 60ms across various datasets. Ownership verification is completed within 520.5ms to 850.5ms. Since adversarial training is performed offline during model development and ownership verification is invoked only when required, the runtime overhead for each query remains limited to detection and response, ensuring practicality for deployment.

V. Conclusion

In this paper, we presented RADEP, a multifaceted defense framework for MLaaS that combines progressive adversarial training, malicious query detection, adaptive response mechanisms, and ownership verification to counter model extraction and privacy attacks. By leveraging a multi-layered approach, RADEP reduces attack success rates and degrades adversarial responses while minimally impacting legitimate queries. Future work will focus on reducing latency, exploring advanced uncertainty metrics, and enhancing resilience in distributed settings.

References

[1] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "Prada: protecting against dnn model stealing attacks," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 512–527.

[2] 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.

[3] 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.

[4] H. Yu, K. Yang, T. Zhang, Y.-Y. Tsai, T.-Y. Ho, and Y. Jin, "Cloudleak: Large-scale deep learning models stealing through adversarial examples." in *NDSS*, vol. 38, 2020, p. 102.

[5] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4954–4963.

[6] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[7] W. Jiang, H. Li, G. Xu, T. Zhang, and R. Lu, "A comprehensive defense framework against model extraction attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 2, pp. 685–700, 2023.

[8] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, 2017, pp. 269–277.

[9] S. F. Ahamed, S. Banerjee, S. Roy, D. Quinn, M. Vucovich, K. Choi, A. Rahman, A. Hu, E. Bowen, and S. Shetty, "Accuracy-privacy trade-off in the mitigation of membership inference attack in federated learning," *arXiv preprint arXiv:2407.19119*, 2024.

[10] J. Truong, P. Maini, R. Walls *et al.*, "Data-free model extraction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, available at https://openaccess.thecvf.com.

[11] 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.

[12] S. Banerjee, S. Roy, S. F. Ahamed, D. Quinn, M. Vucovich, D. Nandakumar, K. Choi, A. Rahman, E. Bowen, and S. Shetty, "Miabad: 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.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[14] A. Madry, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[15] M. Ducoffe and F. Precioso, "Adversarial active learning for deep networks: a margin based approach," *arXiv preprint arXiv:1802.09841*, 2018.

[16] S. Kariyappa and M. K. Qureshi, "Defending against model stealing attacks with adaptive misinformation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 770–778.

[17] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for bayesian uncertainty in deep learning," *Advances in neural information processing systems*, vol. 32, 2019.

[18] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[19] H. Zhang, N. Cheng, Y. Zhang, and Z. Li, "Label flipping attacks against naive bayes on spam filtering systems," *Applied Intelligence*, vol. 51, no. 7, pp. 4503–4514, 2021.

[20] S. Szyller, B. G. Atli, S. Marchal, and N. Asokan, "Dawn: Dynamic adversarial watermarking of neural networks," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4417–4425.

[21] T. Lee, B. Edwards, I. Molloy, and D. Su, "Defending against neural network model stealing attacks using deceptive perturbations," in *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2019, pp. 43–49.