applied sciences

MDPI

Article

Exploring the Efficacy of Learning Techniques in Model Extraction Attacks on Image Classifiers: A Comparative Study

Dong Han [1], Reza Babaei [1], Shangqing Zhao [2] and Samuel Cheng [1,*]

[1] School of Electrical and Computer Engineering, University of Oklahoma, Norman, OK 73019, USA; dong@ou.edu (D.H.); rezababaei@ou.edu (R.B.)
[2] School of Computer Science, University of Oklahoma, Norman, OK 73019, USA; shangqing@ou.edu
* Correspondence: samuel.cheng@ou.edu; Tel.: +1-918-401-0409

Abstract: In the rapidly evolving landscape of cybersecurity, model extraction attacks pose a significant challenge, undermining the integrity of machine learning models by enabling adversaries to replicate proprietary algorithms without direct access. This paper presents a comprehensive study on model extraction attacks towards image classification models, focusing on the efficacy of various Deep Q-network (DQN) extensions for enhancing the performance of surrogate models. The goal is to identify the most efficient approaches for choosing images that optimize adversarial benefits. Additionally, we explore synthetic data generation techniques, including the Jacobian-based method, Linf-projected Gradient Descent (LinfPGD), and Fast Gradient Sign Method (FGSM) aiming to facilitate the training of adversary models with enhanced performance. Our investigation also extends to the realm of data-free model extraction attacks, examining their feasibility and performance under constrained query budgets. Our investigation extends to the comparison of these methods under constrained query budgets, where the Prioritized Experience Replay (PER) technique emerges as the most effective, outperforming other DQN extensions and synthetic data generation methods. Through rigorous experimentation, including multiple trials to ensure statistical significance, this work provides valuable insights into optimizing model extraction attacks.

Keywords: synthetic data; security; active learning; reinforcement learning; model extraction attack; image classification

Citation: Han, D.; Babaei, R.; Zhao, S.; Cheng, S. Exploring the Efficacy of Learning Techniques in Model Extraction Attacks on Image Classifiers: A Comparative Study. *Appl. Sci.* **2024**, *14*, 3785. https://doi.org/10.3390/app14093785

Academic Editor: Christos Bouras

Received: 2 April 2024
Revised: 21 April 2024
Accepted: 28 April 2024
Published: 29 April 2024

1. Introduction

The rise of adversarial attacks targeting image classification models has become a major issue, highlighting vulnerabilities in these models that can be exploited to deceive them into making incorrect predictions or classifications [1–5]. Among the various categories of adversarial attacks, the black-box adversarial attack problem stands out due to its practical relevance and the realistic constraints it imposes on attackers. In contrast to white-box attacks, which assume attackers possess complete information about the architecture and parameters of the model, black-box assaults are conducted with the understanding that the attacker has little to no understanding of the target model's internal workings [6]. This situation is more prevalent in practical settings, where attackers usually do not have direct access to proprietary or confidential models.

A key challenge in executing successful black-box attacks lies in the strategic utilization of inquiries to the target model. Each query provides feedback regarding the model's output for a given input, but excessive querying can not only raise suspicion but also become impractical due to rate limits or cost considerations [7]. Prior art in this domain has explored various strategies to generate adversarial examples under these constraints, yet there remains a significant gap in utilizing query feedback effectively [8]. This inefficiency in feedback utilization leads to suboptimal attack strategies that require a large number of queries, thereby increasing the risk of detection and reducing the feasibility of the attack [9].

A standard example of an adversarial model extraction attack appears in Figure 1. By systematically querying the model and analyzing its responses, attackers can infer valuable information, allowing them to construct a surrogate model that tightly resembles the original [10,11].

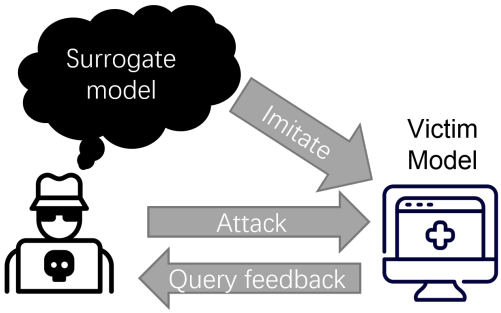

Figure 1. Black-box adversarial model extraction attack. The surrogate model is refined using feedback from queries to more closely mirror the victim model.

The implications of successful black-box adversarial model extraction attacks are multifaceted, ranging from intellectual property theft and undermining the competitive advantage to enabling more sophisticated attacks, such as crafting adversarial examples or identifying vulnerabilities within the model [12]. In this research, we investigate two key aspects of model extraction attacks: sample selection and data-free strategies. Sample selection involves determining which input samples are most advantageous for querying the target model, potentially leading to more efficient and effective extraction. Conversely, data-free attacks seek to replicate a model without requiring accessibility to the training data's initial source, presenting a significant threat to privacy.

For sample selection, we explore the efficacy of various extensions to the Deep Q Network (DQN) [13,14], leveraging the combined strengths of active learning and reinforcement learning techniques. Active learning, in our approach, serves to judiciously select queries that are most likely to yield informative feedback from the victim model, thereby enhancing the training of a surrogate model, which simulates the target model's decision boundaries. At the same time, reinforcement learning refines the process of selecting queries, establishing an adaptive approach that reduces the number of queries needed while enhancing the strength of the adversarial examples created [15]. By leveraging the power of DQN and its variants, such as Double DQN (DDQN) [16], Dueling DQN [17], Noisy network [18], Soft update [19], Prioritized Experience Replay (PER) [20], and Emphasizing Recent Experience (ERE) [21], we aim to identify the most informative samples for querying the target model. Our approach involves training DQN agents to learn sample selection strategies that maximize the information gained from each query, potentially leading to more accurate model extraction with fewer queries.

In addition to sample selection, we investigate techniques for generating synthetic samples to aid in model extraction. Specifically, we evaluate the performance of gradient-based approaches, including Jacobian-based methods [22], Linf-projected Gradient Descent (LinfPGD) [23], and Fast Gradient Sign Method (FGSM) [24]. These techniques utilize the target model's gradients to produce samples that assist an adversary train a replica model with improved performance. Furthermore, we explore data-free model extraction attacks [25], which aim to extract a model without needing access to the original training dataset. This approach raises significant privacy concerns, as it potentially enables an adversary to obtain a functional replica of a model solely through querying, without requiring any data from the model owner.

Through extensive experiments and analysis, we aim to provide valuable insights into the effectiveness of these sample selection and data-free strategies for model extraction attacks. Our findings will contribute to a better understanding of the vulnerabilities present in image classification frameworks and contribute to the creation of robust defense mech-

anisms to protect intellectual property and preserve privacy. Our key contributions are outlined as follows: (i) To the best of the author's knowledge, this work presents the first attempt to leverage DQN extensions to model extraction attacks, which has not been explored in the existing literature. (ii) The development of an open-source framework that integrates popular DQN extensions such as Prioritized Experience Replay, Double DQN, and Dueling DQN. We illustrate how these can be used to enhance attack methodologies against neural networks. (iii) The conduct of comparative experiments that benchmark the effectiveness of these methods against traditional approaches, providing a clear demonstration of their relative performance and advancing the understanding of effective attack strategies in the realm of AI security. (iv) By evaluating different techniques for synthetic data generation, such as FGSM and LinfPGD, and comparing their effectiveness in the context of model extraction, our work identifies best practices and guidelines for practitioners. This helps in refining the approach to generating synthetic data, which is crucial for training robust adversarial models.

This paper is organized as follows: Section 2 provides an overview of related research in adversarial machine learning, underscoring the progress and challenges of current attack methodologies. Section 3 outlines the process used to develop the model extraction methodology. It provides a comprehensive explanation of how active learning and reinforcement learning strategies were combined, along with techniques for selecting samples to create synthetic data. Section 4 presents the experimental framework and the outcomes of applying our approach to image classification models. Discusses the implications of our findings. Finally, Section 5 suggests avenues for future investigation, and wraps up the paper.

2. Related Work

Recent developments in adversarial attacks on machine learning have attracted considerable interest, highlighting how vulnerable machine learning models are to these kinds of attacks. Han et al. [26] provide a comprehensive survey of adversarial attacks and countermeasures in various domains such as images, graphs, and text, highlighting the evolving landscape of studies in adversarial machine learning. This underscores the urgency of developing strong defense strategies to mitigate these attacks. Pitropakis et al. [27] offers a comprehensive classification and overview of machine learning attacks, concentrating on the development of adversarial machine learning research during the past ten years. They highlight the importance of grasping the nature of adversarial attacks and the defenses against them, especially in models constructed using deep neural networks, and introduce a classification system for different attack methodologies. Gong et al. [28] and Arawinkumaar et al. [29] highlight the vulnerability of machine learning algorithms, particularly deep neural networks, to adversarial attacks. They emphasize the impact of carefully designed perturbations on victim samples, leading to incorrect predictions by the target model. Furthermore, Alshahrani et al. [30] focus on examining adversarial attack strategies and the systems for detecting these attacks within systems for detecting network intrusions using supervised machine learning. Their work illuminates the critical requirement for robust defensive systems to fend off hostile assaults.

In addition, recent research has extended the scope of adversarial attacks to various domains such as network security, IoT, and 5G communications. Ibitoye et al. [31] discuss adversarial machine learning in cyberwarfare, with a focus on attacks against malware classifiers. Sagduyu et al. [32] explore the new vulnerabilities presented by adversarial machine learning, particularly focusing on attacks targeting wireless communications within 5G networks. Meanwhile, Luo et al. [33] introduce an approach based on adversarial machine learning for executing partial-model attacks during the data fusion/aggregation stage in IoT systems, emphasizing the importance of establishing strong defense measures in IoT settings.

Significant progress has been achieved in recent studies focused on active learning and active reinforcement learning within the realm of adversarial attacks. Chen et al. [34] provide an extensive review of adversarial attacks within the realm of reinforcement learning, examining these issues through the lens of AI security, emphasizing the need to understand and mitigate adversarial threats in reinforcement learning systems. Oikarinen et al. [35] introduce RADIAL-RL, a method intended to strengthen reinforcement learning agents' resistance to adversarial attacks. This underscores the progress in creating methods aimed at bolstering the durability of reinforcement learning algorithms. Furthermore, Zhang et al. [36,37] delve into how an adversary can employ reinforcement learning to mount attacks on a victim agent, exploring this dynamic in their research, demonstrating the use of reinforcement learning in modeling adversarial behavior. Moreover, recent research has extended the scope of active reinforcement learning to various domains such as IoT, robotics, and healthcare. Haider et al. [38] propose ACADIA, an efficient and robust adversarial attack method against deep reinforcement learning, demonstrating advancements in adversarial attack techniques in reinforcement learning. Additionally, Ye et al. [39] investigate the use of reinforcement learning for conducting black-box adversarial attacks on healthcare risk prediction models. Their work emphasizes the potential of reinforcement learning to challenge complex systems through adversarial tactics.

Recent research in active learning within the domain of adversarial machine learning highlights the increasing interest in tackling adversarial challenges by employing active learning strategies. Aghdam et al. [40] explore the application of active learning in deep detection models, concentrating on segmentation, convolutional neural networks, and the broader field of deep learning. This research underscores the promise of active learning methods in enhancing the detecting neural networks' resistance to adversarial attacks. Furthermore, Zhang et al. [41] examine Graph Neural Network (GNN) projective ranking-based evasion attacks, showcasing their high effectiveness and the ability to transfer these attack strategies successfully. This study showcases the application of active learning in devising adversarial attacks and evaluating their impact on graph neural networks. Moreover, Miller et al. [42] delve into adversarial active learning, emphasizing prior work related to using active learning in an adversarial setting. This study offers valuable perspectives on incorporating active learning strategies to bolster the defense of machine learning models against adversarial attacks. Furthermore, Dineen [43] investigates the use of reinforcement learning for executing data poisoning attacks on graph neural networks (GNN), highlighting the potential of active learning in addressing adversarial attacks in graph classification settings. Additionally, Debicha [44] examines how adversarial network traffic can be transferred across various machine learning-based intrusion detection systems, assessing its impact and effectiveness, shedding light on the application of active learning in detecting and rejecting adversarial attacks in network security. In our research, we integrate image data within a reinforcement learning framework, adapting it for suitability with an image classification model. This approach can be considered a synergistic combination of active learning and reinforcement learning methodologies.

Black-box adversarial attacks, which leverage query feedback to target generative samples, are designed to optimize generated images in a way that they more effectively deceive the victim model [45,46]. Such attacks aim to meticulously select and refine these generative samples, improving their ability to bypass detection mechanisms.

Several studies have focused on enhancing the success rates of model extraction attacks through various methodologies. For instance, Dong et al. [1] utilized momentum iterative algorithms on a group of models, showcasing the susceptibility of models trained to defend against adversarial attacks to black-box attacks. Similarly, Fang et al. [47] conducted experimental evaluations with the black-box MNIST classification model, highlighting the practical viability of approaches to black-box attacks that are based on reinforcement learning. Furthermore, Chen et al. [48] introduced a method for extracting black-box Deep Reinforcement Learning (DRL) models by observing only their interactions with the environment. Moreover, Tsingenopoulos et al. [49] unveiled AutoAttacker, an innovative

reinforcement learning framework that enables agents to navigate around a black-box model by making queries, thereby successfully extracting and undermining its decision-making processes. Ren et al. [50] suggested enhancing the transferability of black-box adversarial model extraction attacks through the adoption of a lifelong learning framework, drawing inspiration from the principles of lifelong learning. Additionally, Zhang et al. [51] crafted a brute-force approach to model extraction, targeting machine learning-based systems in cybersecurity. This method addresses certain limitations of existing adversarial attack strategies, particularly those relying on generative adversarial networks.

Moreover, Shukla et al. [52] and Cheng et al. [53] explored the development of black-box model extraction attacks that generate hard labels, concentrating on the creation of adversarial examples using only restricted information, such as output labels. These studies shed light on the optimization algorithms and query-efficient approaches for crafting hard-label adversarial attacks.

3. Methods

3.1. Problem Definition

The core problem addressed in model extraction attacks is developing a surrogate model capable of accurately imitating the performance of a sophisticated, victim online image classification model using a reduced dataset. This problem arises due to limited access to the victim model or API and the constraints associated with the computational cost, data privacy, or proprietary limitations that restrict the volume of data available for training. The challenge involves selecting an optimal subset of images that retains the comprehensive diversity and complexity of the larger dataset to effectively train the surrogate model. The surrogate model aims to minimize loss measures such as mean squared error (MSE) between the surrogate and victim models' predictions while achieving comparable accuracy, precision, and recall to the victim model. Our research seeks to explore various methods for image selection, data augmentation, and model training that can leverage limited data to their maximum potential, thereby reducing the dependency on large datasets without compromising the model's performance. Our goal is to offer meaningful observations on the efficiency of employing sample selection and data-free tactics in executing model extraction attacks.

3.2. Active Learning with Reinforcement Learning for Model Extraction Attack

Our goal is to choose an optimal subset of images that encapsulates the full range of diversity and complexity found in the larger dataset, enabling the effective training of the surrogate model. This can be considered as an active learning problem. We set the active learning (AL) challenge using a Markov decision process model, drawing inspiration from related studies such as (Sener et al., 2017 [54]; Casanova et al., 2020 [55]; Gao et al., 2022 [56]). The policy network π is represented as an agent for reinforcement learning, in detail a deep Q-network [57]. This method allows the model to develop selection strategies by leveraging the outputs from both the victim and surrogate models, guided by data-driven insights. Our approach differs from existing methods in the following ways: the problem we address, our definitions of states, actions, and rewards, and the specific reinforcement learning algorithm we utilize to determine the best policy.

We aim to enhance the performance of surrogate models, S, parameterized by θ, using a smaller number of images from a large dataset. This process is iteratively conducted until a specified threshold is met. At each iteration, t, a policy network π, parameterized by ϕ, selects a single image from a randomly sampled subset U for training the surrogate model. The selected images are also processed by the victim model to acquire its outputs. To assess performance, we employ a straightforward metric: determining the percentage of images from the total dataset that are correctly classified.

In our methodology, we begin by training the victim model using the entire training dataset, establishing it as the target. We then divide the training data into three distinct segments. A specific portion, designated as D_t, is used for pre-training the surrogate

models. To mitigate overestimation issues during the policy network's training phase, we initialize ten different surrogate models, all sharing the same architecture. For training the policy network π, another subset, D_r is employed to facilitate the active learning process over multiple episodes. This process focuses on refining an acquisition function designed to enhance the surrogate model's performance using a carefully selected set of K images. We utilize a separate data partition, D_e, to evaluate the policy network, where the surrogate models are trained up to the defined budget B. The total query budget is $K * B$, but for simplicity, we refer to it as B. Figure 2 describes the main workflow of training for the Policy Network and surrogate models.

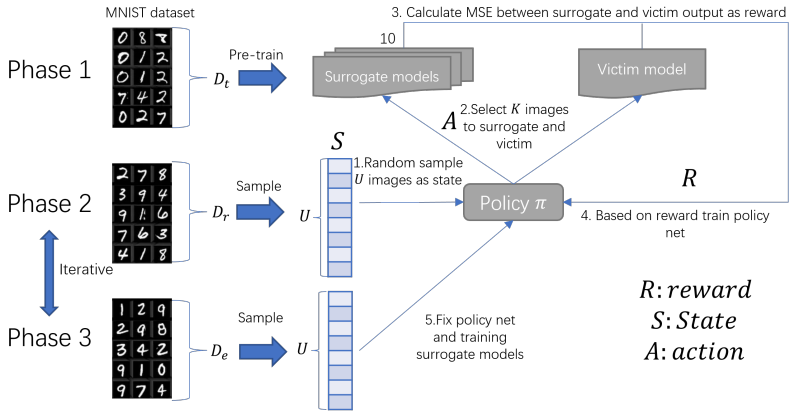

Figure 2. The main workflow of training for Policy π and surrogate models. We split the MNIST dataset into three parts. Phase 1, we use split D_t to pre-train the surrogate models. Phase 2, U images are sampled uniformly from the split D_r. Concatenated U images feature which is computed using CNN feature extractor as the state representation. Policy π selects K images and gives surrogate and victim models. Reward is obtained from the prediction MSE of surrogate and victim models. After policy π is trained, fix it and continue training surrogate model in Phase 3. Phase 2 and Phase 3 are trained iteratively until the final budget B is achieved.

The MDP framework is defined by a sequence of transitions $(s_t, a_t, r_{t+1}, s_{t+1})$. Within this framework, for any given state $s_t \in \mathcal{S}$, the agent can choose an action $a_t \in \mathcal{A}$, which involves selecting specific images from the unlabeled set U. Each action is associated with the selection of a particular image. The agent then receives a reward r_{t+1}, calculated based on the feedback from both the surrogate and victim models. It is important to note that the definitions of states and actions are not contingent on the specific architecture of the surrogate models. Our goal is to develop a policy that enhances sample selection to improve the accuracy of the surrogate model. To achieve this, we utilize a DQN architecture, employing transitions from an experience replay memory ε to train the policy π.

We initiate the process by assigning a set of initial weights θ_0 to the ten surrogate models that have been trained using the dataset D_t. The following steps are performed during each iteration t:

1. Randomly sampled subset U from D_r.
2. The state s_t is computed from the feature extractor by given U.
3. Using an ϵ-greedy policy, the policy agent chooses action a_t. Each action corresponds to selecting one image to be used for training.
4. The surrogate models and the victim model generate output based on the selected image.
5. The agent receives the reward r_{t+1}, which is calculated as the performance differential between the surrogate model and the victim models.
6. The surrogate models are trained one iteration on the recently selected images.

3.3. DQN

The desired agent is designed to adhere to an optimal policy, linking each state to an action that maximizes total future rewards. We conduct our DQN training using a divided

dataset D_r. The agent observes the current state U of the environment. Using a neural network, the agent selects actions (image) based on the current Q-values, which estimate the quality of a state-action combination. After taking an action, the agent receives a reward and transitions to a new state. The Q-values are updated based on the reward received and the maximum predicted Q-values for the new state, using the Bellman equation. To stabilize learning, DQN uses a replay buffer where past experience tuples (state, action, reward, next state) are stored. The agent then samples from this buffer to break the correlation between sequential observations. The development of Deep Q-Networks (DQN) has led to numerous important advancements, each aimed at overcoming certain shortcomings and improving the overall efficacy of the algorithm. These advancements have greatly enhanced the functionality of the target DQN framework, facilitating more effective and resilient learning across intricate settings. In this segment, we introduce the enhancements to the DQN model that were evaluated. In the actual implementation, we replace the DQN model with its extensions solely to evaluate and compare their performance. We also test different combinations of these extensions to see which synergies are most effective, providing insights into how these extensions interact.

3.3.1. Double DQN

Building on the foundational work [16], we establish the Double DQN as our baseline model. DDQN uses two neural networks, one for selecting the best action and another for evaluating the action's value. When updating the Q-values, the action selection is performed by the online network, but the Q-value for updating is estimated using the target network. A recognized limitation of the DQN algorithm is its propensity to overestimate actual rewards, resulting in exaggerated Q-values. The Double DQN technique [58] offers a solution to this problem through an alteration of the Bellman equation utilized in the Deep Q-network. This modification involves separating the processes of action selection and action evaluation, as follows:

$$Q(s,a;\theta) = r + \gamma Q\left(s', \text{argmax}_{a'} Q\left(s', a'; \theta\right); \theta'\right) \quad (1)$$

In this approach, the primary neural network, denoted by θ, is responsible for identifying the optimal next action, a'. Meanwhile, the target network assesses the selected action to calculate the Q-value. The straightforward adjustment has been demonstrated to mitigate overestimation issues, leading to improved policy outcomes.

3.3.2. Dueling Deep Q-Learning

Wang et al.'s [17] The Dueling DQN approach divides the Q-values into two separate parts, which improves upon the classic DQN: the advantage function, $A(s,a)$, and the value function, $V(s)$. The value function evaluates the expected reward from a certain circumstance s, but the advantage function illustrates the relative benefit of choosing a particular action, a, over other options. Completing Q-value calculations for each state-action combination is made possible by combining these two functions.

Dueling DQN introduces a novel neural network architecture. The network has two streams to separately estimate (a) the state value function and (b) the advantage for each action which represents the additional value of taking that action compared to others. These two values are then combined to calculate the Q-value for each action at a given state, allowing the network to learn which states are (or are not) valuable, independent of the actions taken. Such an adjustment enhances the network's learning efficiency, especially in scenarios where the precise value of each action is less critical, by allowing it to concentrate on understanding the state's value function.

3.3.3. Target Network Soft Update

The soft update technique for target networks represents an important innovation for deep reinforcement learning, as highlighted by Lillicrap et al. [19]. The Soft Update technique is commonly used in reinforcement learning algorithms that utilize two neural

networks: the main network and the target network. The main network is actively trained during each iteration, while the target network is used to estimate future rewards more stably. The structure of both networks is typically identical and consists of several layers (e.g., convolutional layers for image inputs, followed by dense layers), but the target network's weights are updated less frequently to provide stable learning targets. This method is designed to enhance the stability and effectiveness of training neural networks for RL applications. Unlike hard updates, which copy the parameters from the main network to the target network abruptly and at specified intervals, soft updates progressively move the target network's parameters closer to those of the main network. By doing so, it avoids the potential for significant disruptions in the target network that could undermine the learning process. In the soft update method, the weights of the target network are slowly updated to the weights of the main network using an interpolation parameter τ. The parameter τ, typically set to a value like 0.001 as proposed in the DPG algorithm, regulates the rate of this blending process. The update mechanism for the target network is described accordingly.

$$\theta^{\text{target}} = \theta \times \tau + \theta^{\text{target}} \times (1 - \tau) \tag{2}$$

The modest magnitude of the parameter τ ensures that the target network transitions towards the Q-network's parameters in a gradual manner. To make this gradual adjustment impactful, it necessitates regular updates. By using a soft update strategy, the target network may more easily follow changes made to the main network, thereby fostering a learning environment that is both more stable and efficient. This method has shown its worth, especially in intricate reinforcement learning endeavors, where preserving stability throughout training is vital for attaining peak performance.

3.3.4. Prioritized Experience Replay

Schaul et al. [20] in 2015, established the idea of prioritized experience replay (PER), and proposes a solution that incorporates an additional data structure to rank the importance of each transition. This system then selects experiences for replay based on these prioritized rankings. The network used with PER is the same as the previous DQN network. Unlike regular experience replay where transitions are sampled uniformly, PER modifies the sampling mechanism by prioritizing transitions from which there is more to learn (i.e., transitions with higher prediction error). To compensate for the altered distribution caused by prioritized sampling, importance-sampling weights alpha are used to adjust the learning updates, ensuring that the learning remains unbiased. After each learning update, the priorities in the replay buffer need to be updated based on the new errors.

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \tag{3}$$

The amount of bias in the sampling process is controlled by the hyperparameter alpha, and priorities are determined by the temporal difference error that the agent recorded during its most recent training session on a certain event. This approach allows the agent to improve its efficiency in learning from the samples by concentrating on regions where its predictions are the most erroneous. Replay buffer additions are given top priority, guaranteeing a bias toward more recent events. However, it is crucial to acknowledge that this method may also favor stochastic transitions, despite the potentially minimal knowledge gain they offer.

3.3.5. Emphasizing Recent Experience

Originally created to accelerate the Soft Actor-Critic (SAC) algorithm's rate of convergence, as discussed by Wang et al. [21] in 2019, this approach has the potential to be adapted across various algorithms and tasks that benefit from faster incorporation of recent experiences, especially in complex scenarios involving multiple elements. The fundamental idea is to sample a mini-batch from the whole dataset in the replay buffer before moving on to the parameter updating phase. For each following mini-batch, The sampling interval

is gradually limited for each subsequent mini-batch to increasingly focus on newer data points. This strategy is underpinned by two key principles: (i) increasing the frequency of sampling recent data and (ii) organizing updates in a manner that prevents the erasure of newer information by older data. The implementation of Experience Replay Emphasis (ERE) introduces a simple but efficient sampling method, allowing the agent to give precedence to newer transitions while still acknowledging the value of past experiences.

3.3.6. Noisy Network

Often used in place of the epsilon-greedy approach, noisy networks allow for a more dynamic and effective exploration phase during training, as highlighted by Fortunato et al. [18]. In contrast to the epsilon-greedy method, which modulates exploration through a fixed probability, noisy networks inject randomness straight into the parameters of the network. A Noisy Network modifies a standard neural network architecture by adding noise to the weights or biases of the network to encourage exploration. This typically involves replacing some or all of the network's layers with noisy layers where the noise is parameterized and learned during training. This approach allows for a more refined and ongoing exploration. The Noisy Network concept pioneers the introduction of a noisy linear layer that combines deterministic elements with stochastic components, offering a fresh perspective on network exploration strategies. Noise parameters are added to the network's weights and biases, which are learned through gradient descent along with the standard parameters of the network. For example, a noisy linear layer can be represented as:

$$y = (b + Wx) + \left(b_{\text{noisy}} \odot \epsilon^b + \left(W_{\text{noisy}} \odot \epsilon^w\right)x\right) \tag{4}$$

The random variables in this case are ϵ^b and ϵ^w, and the operation of element-wise multiplication is denoted by $\odot$. The network can gradually reduce the noise's impact over time; however, the rate of this decline may vary across various parts of the state space. This mechanism permits exploration that is tailored to specific states, incorporating a natural form of self-reduction in reliance on noise. Such a dynamic approach to exploration achieves a more nuanced equilibrium between exploring new possibilities and leveraging existing knowledge, thereby enhancing the network's learning efficiency and flexibility in dealing with complex scenarios.

3.4. Synthetic Data Generation Techniques

In the training of surrogate models using synthetic generation techniques, our workflow is structured into two main phases to leverage the image dataset effectively. In Phase 1, we utilize a portion of the dataset, denoted as D_t, to pre-train the surrogate models. This initial training establishes a baseline capability of the surrogate models to approximate the behavior of the target or victim model. Moving into Phase 2, we sample images uniformly from another dataset split, D_r, to begin the synthetic data generation process. Here, the pre-trained surrogate model is employed to generate synthetic image data, which involves perturbing original images to create modified versions. These altered images are then fed into the victim model, which processes them and provides predictions based on its learned parameters. The combination of the perturbed images and the victim model's predictions is then used to further train the surrogate models. This approach ensures that the surrogate models are not only trained on real data but are also finely tuned to mimic the victim model's behavior as closely as possible by learning from synthetically generated adversarial examples. In Figure 3 shows the overall workflow.

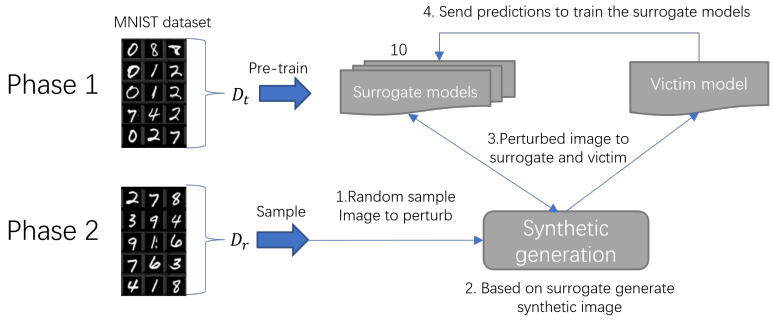

Figure 3. The main workflow of training surrogate models using synthetic generation techniques. We split the MNIST dataset into two parts. In phase 1, we use split D_t to pre-train the surrogate models. In phase 2, the image is sampled uniformly from the split D_r. Generate synthetic image data using the surrogate model. Submit the altered image to the victim model to obtain predictions. Use the perturbed image and its associated predictions to train the surrogate models.

3.4.1. Jacobian-Based

The Jacobian-based technique is a gradient-based strategy that creates synthetic samples for model extraction by utilizing the target model's Jacobian matrix [22]. The sensitivity of the model's outputs to its inputs is captured by the Jacobian matrix, which offers useful information for creating instructive examples. A target model f and an input sample x are used to calculate the Jacobian matrix $J(x)$.

$$J(x) = \frac{\partial f(x)}{\partial x}$$

The partial derivatives of the model's outputs in relation to each input feature are contained in the Jacobian matrix $J(x)$, which essentially captures the local behavior of the algorithm around the input value x.

We can apply a perturbation onto the data x in the way of the Jacobian matrix to produce synthetic samples. In particular, a synthetic sample x' can be acquired as follows:

$$x' = x + \epsilon \cdot \text{sign}(J(x))$$

where the step size that regulates the perturbation's magnitude is ϵ, and $\text{sign}(\cdot)$ is the sign function applied element-wise to the Jacobian matrix.

The intuition behind this approach is that perturbing the input sample in the direction of the Jacobian matrix is likely to produce synthetic samples that exhibit high sensitivity to the target model's outputs. These synthetic samples can then be used to train a replica model, potentially improving its performance and accuracy in approximating the target model. Various strategies can be employed to optimize the synthetic sample generation process, such as exploring different step sizes, combining multiple Jacobian-based samples, or incorporating additional constraints or regularization techniques. It is crucial to remember that the Jacobian-based approach relies on being able to reach the gradients of the target model, which could not be possible in some circumstances. In such cases, alternative gradient estimation techniques or other synthetic data generation approaches may be necessary.

3.4.2. The Fast Gradient Sign Method

A gradient-based method for producing adversarial instances is the Fast Gradient Sign Method (FGSM) [24], which can be adapted for the purpose of synthetic data generation in model extraction attacks. FGSM is an efficient and computationally inexpensive method that creates artificial samples by utilizing gradients that are part of the target model.

Given a target model f, an input sample x, and a loss function $\mathcal{L}(\cdot,\cdot)$, By perturbing x in the orientation of the gradient of the loss function in relation to the input, the FGSM creates a synthetic sample x':

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(x), y))$$

where y is the ground truth label or target output for x. The sign function applied element-wise to the gradient is sign$(\cdot)$, and a tiny variable ϵ regulates the size of the perturbation.

The fundamental concept underlying FGSM is to traverse the trajectory that aligns with the gradient of the loss function about the input, to produce a synthetic sample x' that maximizes its loss function $\mathcal{L}(f(x'), y)$. This perturbation is intended to generate samples that are informative and challenging for the target model, potentially leading to improved model extraction performance when used for training a replica model. FGSM is an efficient method as it requires only a single gradient computation per input sample. However, it may be susceptible to gradient masking or obfuscation techniques employed by the victim model for defense against model extraction attacks. To enhance the effectiveness of FGSM for model extraction, various extensions and modifications can be explored, such as iterative approaches (e.g., Projected Gradient Descent), incorporating additional constraints or regularization techniques, or combining FGSM with other synthetic data generation methods.

3.4.3. Linf-Projected Gradient Descent

An iterative gradient-based technique for creating adversarial examples is the Linf-projected Gradient Descent (LinfPGD) [23]. This technique may be modified to create synthetic data for model extraction assaults. An expansion of the Fast Gradient Sign Method (FGSM) is LinfPGD which aims to create more robust and effective synthetic samples. Given a victim model f, an input sample x, a loss function $\mathcal{L}(\cdot,\cdot)$, and a maximum perturbation budget ϵ, Under the L_∞ norm constraint, LinfPGD iteratively updates x in the path of the direction of the gradient of the loss function to produce a synthetic sample x':

$$x't + 1 = \Pi\epsilon\left(x'_t + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(f(x'_t), y))\right)$$

where y is the ground truth label or target output for x, α is the step size, sign$(\cdot)$ is the sign function applied element-wise to the gradient, and $\Pi_\epsilon(\cdot)$ is the projection operator that projects the disturbance to the L_∞ sphere with radius ϵ.

The iterative nature of LinfPGD allows for more sophisticated and effective synthetic sample generation compared to FGSM. The synthetic sample is updated often in the path of the loss function gradient, LinfPGD can potentially explore a larger region of the input space and generate samples that are more informative and challenging for the target model. In order to preserve the semantic and structural characteristics of the input data, the L_∞ norm constraint makes sure that the synthetic samples stay boundedly close to the original input sample.

LinfPGD offers several advantages over FGSM, including improved robustness to gradient masking or obfuscation techniques, and the ability to generate more diverse and informative synthetic samples. However, it is computationally more expensive than FGSM due to the iterative nature of the algorithm. Various modifications and extensions to LinfPGD can be explored, such as adjusting the step size, incorporating momentum or restart strategies, or combining it with other synthetic data generation techniques to further enhance its effectiveness for model extraction.

3.4.4. Data-Free Model Extraction

A technique called "Data-Free Model Extraction" [25] seeks to extract a working duplicate of a target model without having access to the initial training set of data. This technique raises significant privacy concerns, as it potentially enables an adversary to obtain

a model's functionality solely through querying, without the need for any proprietary data from the model owner.

The following actions are usually involved in the data-free model extraction process: Query Generation: In the absence of training data, the adversary generates a set of synthetic input samples to query the victim model. There are several ways to create these artificial samples, including gradient-based approaches, Generative Adversarial Networks (GANs), or other data synthesis approaches. Model Querying: The victim model is queried using the created synthetic input samples, and the matching output predictions or labels are gathered. Replica Model Training: The original training data are replaced with the input-output pairs that are gathered from the target model queries. The functionality of the target model is essentially extracted by using these query responses to train a replica model.

The key challenge in data-free model extraction lies in generating synthetic input samples that can effectively capture the target model's behavior with a variety of inputs. Techniques like gradient-based methods or GANs can be employed to create synthetic samples that provide useful information and accurately depict the decision limits of the target model.

4. Experiments

This section commences by outlining the objectives guiding our experiments, followed by a synopsis of the datasets utilized to assess our methodology, the configuration of our experiments, and the comparative analysis. Our evaluation of the algorithm employs the MNIST dataset with the outcomes presented herein. A single 24 GB VRAM NVIDIA RTX A5000 GPU was used to train each model.

4.1. Experiment Objectives

To Evaluate the Effectiveness of DQN Extensions for Model Extraction Attacks: The objective is to determine how different extensions to the Deep Q-network (DQN) algorithm—such as Double DQN, Dueling DQN, Noisy Networks, Prioritized Experience Replay (PER), etc.—impact the efficiency of surrogate models in model extraction attacks. This involves assessing these extensions' ability to select informative samples that optimize adversarial benefits under limited query budgets.

To Assess Synthetic Data Generation Techniques: The study aims to evaluate the performance of various synthetic data generation methods, including the Jacobian-based method, Linf-projected Gradient Descent (LinfPGD), and Fast Gradient Sign Method (FGSM), in training surrogate models for model extraction. The focus is on the methods' capability to facilitate the training of adversary models with enhanced performance. Another objective is to explore the feasibility and effectiveness of data-free model extraction attacks, which attempt to replicate the functionality of a target model without access to the original training data. This involves assessing how well such attacks perform in constrained query environments.

4.2. Experimental Setup

4.2.1. MNIST

In the fields of computer vision and machine learning, the MNIST dataset (Modified National Institute of Standards and Technology dataset) [59] is a famous dataset, widely used for benchmarking image processing systems and algorithms. It consists of a large collection of handwritten digits, specifically 70,000 28×28 pixel grayscale images of the digits 0 through 9; 10,000 images are kept aside for testing, while the remaining 60,000 are utilized for training purposes in this dataset. We divided the training set into 1000 photos using uniform sampling to build D_t (where we obtain the pre-trained surrogate model), 21,000 images to build D_r, and the remaining images in the training set to build D_e, where we fixed trained policy and continue training surrogate model to evaluate the policy. The performance of the surrogate model is evaluated through the use of the test set.

4.2.2. Implementation Details

From the training set D_r, we sample a subset of $U = 20$ images to construct the state representation. To extract data from these images, a Convolutional Neural Network (CNN) feature extractor is utilized, which is then concatenated into a state. The policy network's action corresponds to the index number of the chosen image within U. The split D_r is used to optimize the hyper-parameters, which are chosen based on the best configuration for our technique as well as the baseline methods, and to create the rewards for the DQN. The MSE difference between the victim models' and the surrogate model's predictions is what determines the reward, and it is always negative. In order to solve the overestimation problem inside the policy network, we employ 10 surrogate models and calculate the reward by computing the mean of 10 MSE. The victim model selected for our study is ExquisiteNetV2 (2021) [60], which achieved an accuracy of 99.71% on the MNIST dataset.

In synthetic data generation attacks, we continue to utilize images from D_r. An image is randomly selected for perturbation using the previously mentioned techniques. This perturbed image is then fed into the victim model to produce a prediction. Subsequently, the surrogate model is trained using pairs of the perturbed image and its corresponding prediction. Regarding the data-free model extraction method, we employed randomly generated noise images as inputs to train the surrogate model, while the remaining implementation followed a similar approach to the other techniques. Additionally, we employed the entropy method, a purely active learning approach devoid of reinforcement learning, to serve as a benchmark against the synthetic data generation techniques.

4.2.3. Evaluation

The evaluation process comprises two stages: assessing the performance of the policy network and using our approach to assess the surrogate model's efficacy. With a set budget B, the policy network π is trained using D_r to promote selecting an image that will improve performance in an environment of sparse data. We test the learned acquisition function and the synthetic data generation techniques on D_e, where we train the surrogate models for various budgets until the budget is satisfied.

Once the budget is reached, we fix the policy network and proceed to evaluate the surrogate model with the test data. The objective of our work would be to show that through the application of the aforementioned method, the surrogate model is capable of producing diverse outcomes. Our goal is to evaluate this method alongside others to identify the most efficient approach, one that requires fewer images to reach equivalent levels of performance. Essentially, this involves training the surrogate model with an identical quantity of images while achieving superior results in comparison to alternative methods.

4.3. Results

In Figure 4, we contrast different approaches with growing budgets for executing black-box attacks. The "Budget" labeled X-axis shows the number of queries we train the surrogate model. We set specific policies in place and trained surrogate models accordingly. Following this, we conducted evaluations of the surrogate models using a test dataset. Given the significant variability inherent to the reinforcement learning (RL) algorithm, we conducted 10 trials to average the accuracy across various trained policy models. The policy models were trained with $K = 4$ and a budget of 3000. As the DQN model was initially trained using a budget of 3000, the total budget range represented in this figure spans from 3000 to 13,000. We also integrated all DQN extensions to identify the configuration that achieves optimal performance. The findings reveal that this combination of methods outperforms each method when used individually. Specifically, among the standalone applications, PER demonstrated the highest performance.

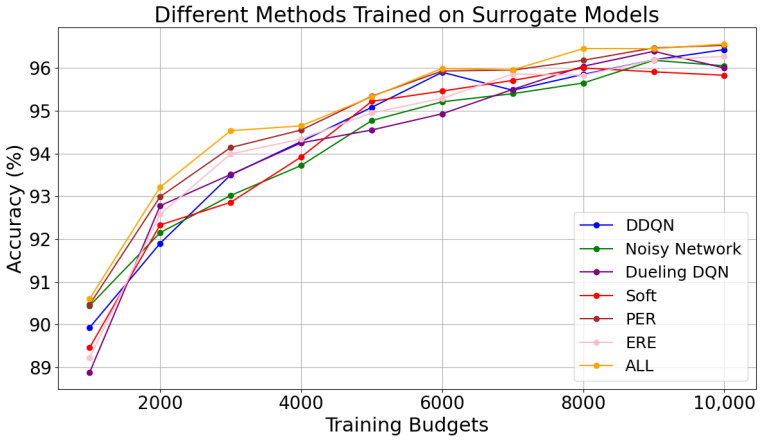

Figure 4. Performance of different DQN policies on surrogate models. Here's the figure showing the performance of the DoubleDQN (DDQN), Noisy Network, Dueling DQN, Soft update, Emphasizing Recent Experiences (ERE), and Prioritized Experience Replay (PER). Each line represents a different DQN model, with the X-axis indicating training surrogate model budgets from 1000 to 10,000 in 1000 intervals, and the Y-axis displaying the model's accuracy.

We also investigate how an increase in budget to train policy models affects the performance of our method. Figure 5 shows the results of the increase in the budget for different trained policy performances. Due to limited accessibility regarding the victim model, we are interested in the limited queries for the evaluation of the surrogate model. We are constrained by an evaluation budget of 1000 queries. It is important to note that we also utilize budgets during the training of the policy models, so the total query budget will be 1000 plus additional budget allocations for the different policy model variants. Across all models, the Prioritized Experience Replay (PER) generally yields higher accuracy compared to the other methods and Double DQN also performs well. We evaluated the performance with various sample sizes of selected images, specifically for $K = 1$ and $K = 3$. However, these configurations did not outperform $K = 4$. Therefore, we present our best performance results achieved with a sample size of $K = 4$.

For synthetic data generation methods, the experiments aimed to assess the effectiveness of various techniques in producing samples that enable to training of a model with improved performance for model extraction attacks. Specifically, we looked into the Jacobian-based, Linf-projected Gradient Descent (LinfPGD), Fast Gradient Sign Method (FGSM), Entropy, and a data-free model extraction approach. To ensure a robust and fair analysis, the experiment was repeated 10 times, with the results' mean calculated to account for variability. We also added one DQN method to compare with synthetic data generation methods. Figure 6 illustrating these findings was created to visually compare the performances of the different methods. Among these techniques, PER emerged as the superior method, consistently outperforming the others in terms of achieving higher accuracy in the model extraction process. In synthetic data generation methods, LinfPGD outperforms other methods. Conversely, the data-free method demonstrated significantly lower performance. This outcome suggests that the allocated query budget may have been insufficient for generating realistic and useful images for training purposes. The effectiveness of synthetic data generation in model extraction attacks appears to be highly dependent on the ability to produce quality training data, a criterion where the data-free method fell short under the constraints of a limited query budget.

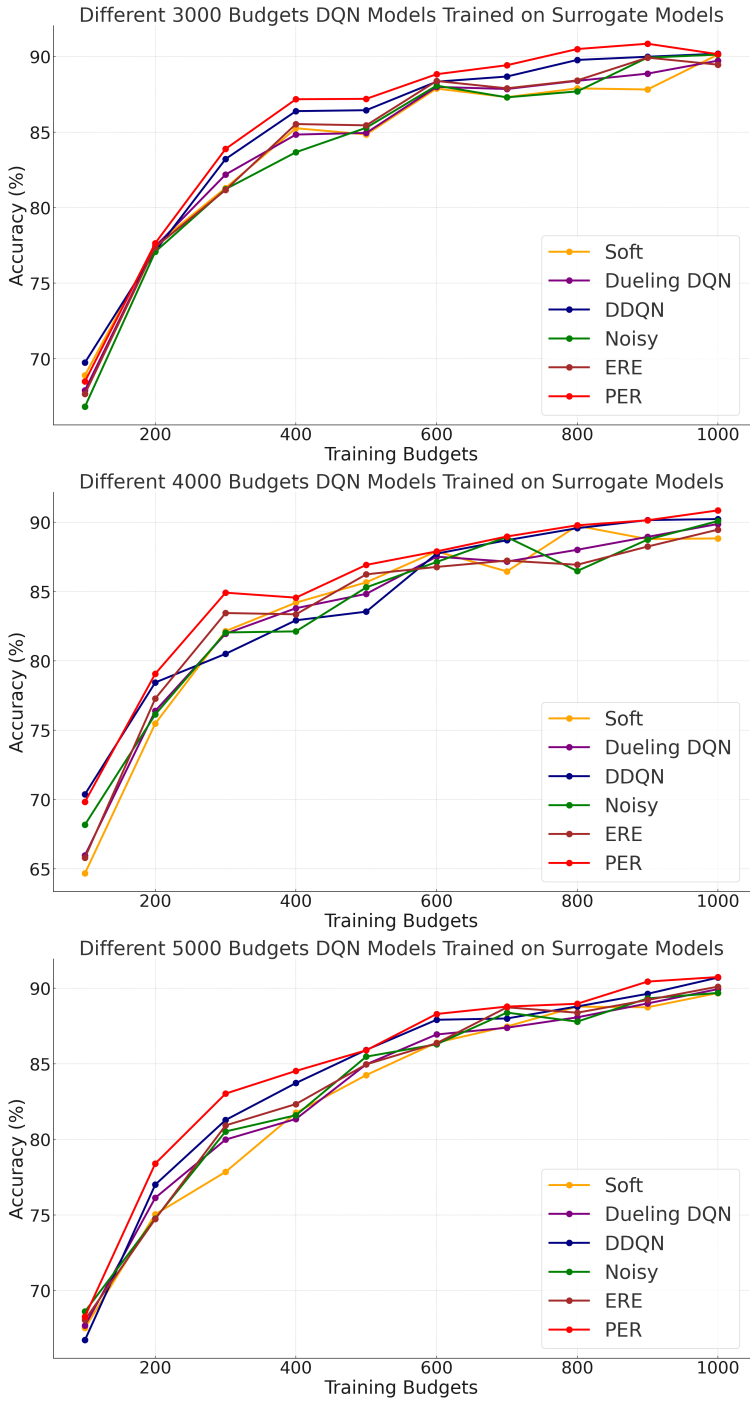

Figure 5. Performance of various DQN policy variants on surrogate models as the training budget increases. The X-axis represents the training budget for the surrogate model, ranging from 100 to 1000 in increments of 100, while the Y-axis shows the model's accuracy. The policy variants evaluated include DoubleDQN (DDQN), Noisy Network, Dueling DQN, Soft Update, Emphasizing Recent Experiences (ERE), and Prioritized Experience Replay (PER). It is important to note that direct comparison across the three subfigures is not appropriate since the query budgets used for training the DQN models differ in each case. The total budgets are 4000 for the first subplot, 5000 for the second, and 6000 for the third.

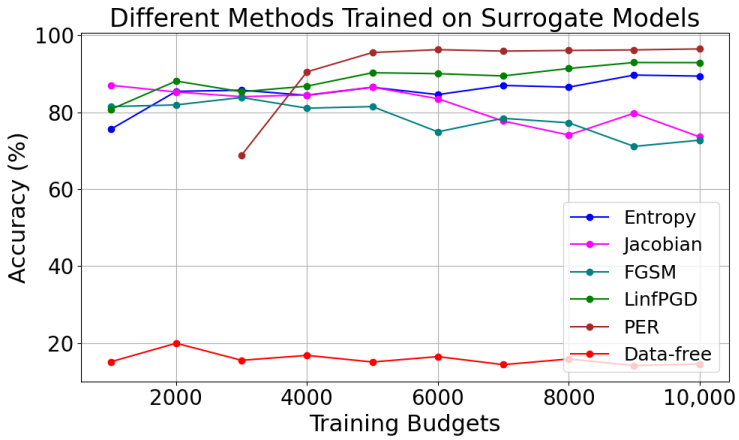

Figure 6. Performance of different synthetic data generation techniques with PER method on surrogate models. The figure above illustrates the Jacobian-based methods, Linf-projected Gradient Descent (LinfPGD), Fast Gradient Sign Method (FGSM), Entropy, and a data-free model extraction method. The X-axis indicates training budgets from 1000 to 10,000 in 1000 intervals, and the Y-axis displays the model's accuracy. For comparison, we also included the PER method trained using the DQN model. Since a budget of 3000 was utilized to train the DQN model, its performance is plotted to start from a budget of 3000 up to 10,000.

4.4. Ablation Studies

We have demonstrated that several improvements to DQN can successfully achieve better performance. Then we conducted additional experiments to test combinations of these extensions and assess how the removal of each DQN extension affects the performance of model extraction. To better comprehend the influence of each component on the DQN agent, we conducted ablation studies. In each study, we systematically removed one component from the complete set of these extensions. Note, that ERE is implemented based on PER, so when PER is removed in our ablation study, ERE is also omitted. In this context, we focus on the scenario without PER. Figure 7 shows a comparison of the accuracy score of the full combinations to five ablated variants.

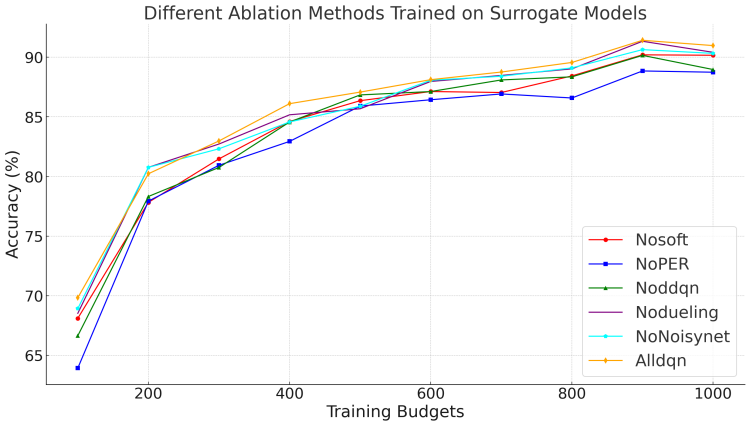

Figure 7. Performance of different ablation methods trained on surrogate models. The figure above compares our combination method with five different ablations. The X-axis indicates training budgets from 100 to 1000 in 100 intervals, and the Y-axis displays the model's accuracy. Since a budget of 3000 was utilized to train the DQN model, the total budget is 4000.

The findings indicate that Prioritized Experience Replay (PER) was the most critical element among DQN enhancements, as its removal significantly reduced performance. The soft update and double DQN followed in importance, according to the results of other ablations. It is noteworthy that in the initial stages of learning, specifically during the first

200 queries, there was a noticeable difference between the ablations and the full agent. Overall, removing the dueling network and Noisy net from the combined setup did not result in a significant performance change. Both the dueling DQN and Noisy net tend to introduce instability, complicating the maintenance of a stable learning trajectory and convergence to an optimal policy.

4.5. Generalizability

In addition, we conducted preliminary experiments on the CIFAR-100 dataset to demonstrate the robustness of our model. The CIFAR-100 dataset is a popular benchmark in the field of machine learning, particularly for image recognition tasks. Developed by the Canadian Institute for Advanced Research (CIFAR) [61], this dataset contains 60,000 32 $\times$ 32 color images categorized into 100 classes, each class comprising 600 images. We divided the training set into 10,000 photos using uniform sampling to build D_t (where we obtain the pre-trained surrogate model), 21,000 images to build D_r, and the remaining images in the training set to build D_e. This setup makes the CIFAR-100 dataset particularly challenging and useful for developing and testing advanced image classification algorithms that require fine-grained recognition capabilities.

All the training and evaluation process is followed by previous methods. To simplify, we conduct five trials to average the accuracy across the policy models. The victim model selected for our study is EffNet-L2 (2020) [62], which achieved an accuracy of 96.08% on the dataset. In Figure 8, we contrast different approaches with increasing budgets for model extraction attacks. The policy models were trained with $K = 4$ and a budget of 3000. We double the queries for better results. The total budget range represented in this figure spans from 3000 to 23,000. From experiments with the CIFAR100 dataset, we observe that larger victim and surrogate models increase the required runtime. The results show that the performance on the CIFAR-100 dataset is relatively low. This dataset is inherently more complex than simpler ones such as MNIST because it comprises 100 classes. This level of granularity necessitates a model with greater capacity to adequately distinguish between the numerous classes. Notably, as the model sizes increase and datasets become more complex, performance tends to decrease, requiring additional queries to effectively extract the model.

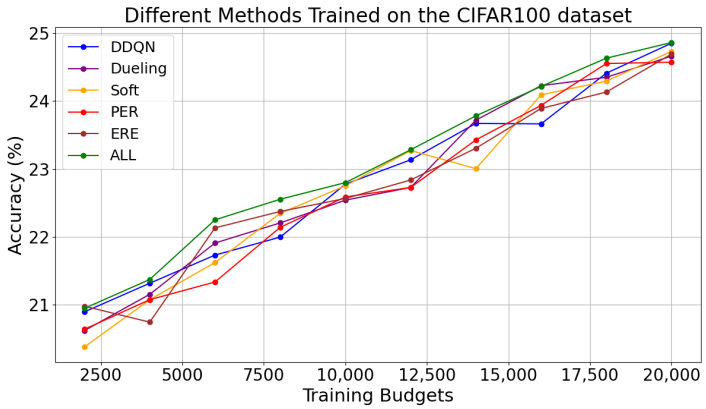

Figure 8. Performance of different methods trained on CIFAR100 dataset. The figure above compares our combination method with five different DQN extensions. The X-axis indicates training budgets from 2000 to 20,000 in 2000 intervals, and the Y-axis displays the model's accuracy. Since a budget of 3000 was utilized to train the DQN model, the total budget is from 3000 to 23,000.

4.6. Discussion

In this part, we analyze the primary results of the tests and discuss the findings of the experiment. Across all the experiments, utilizing DQN models outperformed synthetic data generation methods. This superiority stems from the fact that DQN models leverage the original data to train the surrogate model, thereby imbuing it with the strongest

knowledge derived from the dataset. The experimental findings indicate that employing the Prioritized Experience Replay (PER) technique leads to improved performance across all the experiments. This suggests that the PER approach, which likely involves more sophisticated decision-making or optimization processes, is more effective at utilizing the training budget to improve model performance. Experiences are ranked by PER according to their temporal difference (TD) error, or the discrepancy between the expected and actual rewards. Experiences with higher TD errors are considered more valuable for learning because they indicate situations where the model's predictions were significantly off. By focusing on these experiences, PER allows the model to learn and adapt more quickly to critical aspects of the environment. By concentrating on experiences that the model currently finds most challenging, PER can speed up the learning process, leading to faster convergence towards optimal policies. This targeted learning approach can be particularly effective in complex environments where certain states and actions are more critical to mastering the task. Focusing on high-error experiences accelerates learning, Additionally, PER includes safeguards against the model concentrating too much on a small number of experiences, thereby maintaining a balance between exploring new knowledge and exploiting known information. This balance helps in achieving not only faster but also more stable learning outcomes.

DDQN comes in second because it addresses the overestimation of action values that can occur with the standard DQN. By decoupling the selection of actions from their evaluation, DDQN reduces the bias in the learning updates, leading to more stable and reliable learning outcomes. This refinement helps it perform well, though it lacks the targeted efficiency improvements of PER. ERE ranks third and enhances learning by focusing on a mix of recent experiences and the entire history, gradually focusing more on recent interactions. This approach helps in adapting to changes and optimizing the model's performance over time by balancing between exploiting learned knowledge and exploring new information. However, it may not be as targeted or efficient in prioritizing critical learning opportunities as PER. The Noisy Network ranks the lowest among the tested methods. While it introduces noise into the network parameters to encourage exploration by diversifying the policy's behavior, this can also lead to less stable training and potentially poorer performance when the task requires precise adjustments based on subtle cues from the environment, as is the case in model extraction attacks. The randomness introduced might not always align with the most informative learning paths, unlike the more structured approaches of PER, DDQN, and ERE. In summary, the key to PER's potential superiority lies in its ability to prioritize learning from the most informative experiences, thus optimizing the learning process, accelerating convergence, and eventually resulting in enhanced DQN model performance in model extraction tasks.

One critical aspect of dealing with CIFAR-100 is the need for a model with greater capacity. This could mean exploring deeper or more complex neural network architectures that are capable of capturing a wide range of features across diverse image types. Models such as convolutional neural networks (CNNs) that are deeper and equipped with layers like residual connections might prove effective. These models can learn rich and hierarchical feature representations, making them more adept at handling the intricacies of such a complex dataset. Furthermore, the low performance on CIFAR-100 as observed might also be a result of overfitting to specific features that do not generalize well across the entire dataset. This issue can be mitigated by employing techniques such as data augmentation, regularization, and dropout to enhance the model's ability to generalize rather than memorize the training data. Another approach to improving performance, as suggested, is increasing the number of queries during training. This method, often associated with active learning, involves iteratively selecting the most informative samples from the dataset to train the model. By strategically increasing the number of queries, it is possible to refine the training process to focus on examples that are most beneficial for learning, thereby potentially improving model accuracy and robustness.

In situations where obtaining the original dataset is challenging, the use of synthetically generated data becomes necessary. Among the synthetic data generation methods evaluated, the LinfPGD approach demonstrated superior performance compared to the other techniques. This method iteratively fine-tunes adversarial examples, carefully exploring the space around the original input within the constraints of the L_∞ norm. This methodical exploration allows it to identify perturbations that are both subtle and highly effective at deceiving the model, leading to more potent adversarial examples. Its systematic approach to exploring adversarial space, coupled with controlled perturbations and iterative optimization, allows LinfPGD to achieve superior performance in creating synthetic data that extract models. The Jacobian-based method follows LinfPGD in effectiveness. It leverages the model's Jacobian matrix to determine the directions in which the input features should be perturbed to maximize output changes, focusing on the areas where the model is most sensitive. This method effectively uses gradient information to create informative adversarial examples, which can significantly aid in model extraction by highlighting the model's decision boundaries. However, because it is generally a single-step approach, it might not capture as nuanced a view of the model's vulnerabilities as the iterative LinfPGD method, resulting in slightly less effective synthetic data for training. FGSM is a more straightforward and faster approach than both LinfPGD and the Jacobian-based method, as it involves just a single step of perturbation along the gradient of the loss. This makes FGSM computationally cheaper and quicker to apply, but it also tends to be less refined. FGSM's single-step nature means it might not find the optimal perturbation for the most informative adversarial examples, especially against well-regularized or complex models. It often leads to examples that are either too easy for the model to classify correctly or too aggressive, pushing the examples out of the data distribution and, thus less effective for training robust surrogate models. While data-free methods rely on randomly generated noise as input images to train the surrogate model, the results are poor because the input-output pairs consist of random noise images paired with the victim model's outputs, which lacks meaningful correspondence.

5. Conclusions

In this research, we delved into model extraction attacks, focusing on the efficiency of various DQN extensions for sample selection and synthetic data generation techniques like Jacobian-based methods, FGSM, LinfPGD, and data-free attacks for enhancing surrogate model training. Our contributions include evaluating the effectiveness of DQN extensions (DDQN, Dueling DQN, Noisy Network, Soft update, PER, ERE) for selective sample extraction, investigating both gradient-based and data-free synthetic data generation techniques for model extraction, and conducting comprehensive experiments on MNIST datasets. Our study contrasted these approaches across different query budgets to identify the most effective strategies for executing black-box attacks under limited query scenarios.

Our findings reveal that the Prioritized Experience Replay (PER) technique consistently outperforms other methods in achieving higher accuracy during the model extraction process. Specifically, PER's sophisticated decision-making optimizes the use of training budgets more effectively. In contrast, data-free methods demonstrated significantly lower performance, likely due to an insufficient query budget for generating useful training images. Among synthetic data generation techniques, LinfPGD emerged as the most effective, systematically identifying adversarial examples that improve surrogate model performance. This highlights the importance of quality training data in model extraction attacks, where LinfPGD's methodical approach to generating adversarial examples proves crucial.

Through comprehensive testing and analysis, our study confirms that utilizing DQN models, particularly those employing the PER approach, significantly outperforms synthetic data generation methods. This is attributed to DQN models' ability to leverage original data effectively, thereby incorporating the most informative insights into the surrogate model. The research underscores the pivotal role of synthetic data generation in scenarios where access to the original dataset is constrained, with LinfPGD emerging as the

most effective method in such contexts. In conclusion, this paper sheds light on the multi-faceted strategies of model extraction attacks, demonstrating the paramount importance of sophisticated sample selection and synthetic data generation techniques in enhancing model performance. The elucidation of PER's and LinfPGD's effectiveness provides valuable insights for advancing model extraction methodologies, setting the stage for future explorations in optimizing attack strategies within constrained environments.

Author Contributions: Conceptualization, D.H. and S.C.; methodology, D.H. and S.C.; investigation, D.H.; validation, D.H. and R.B.; data curation, D.H. and R.B.; writing, D.H. and R.B.; review and editing, S.Z. and S.C.; supervision, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the NSF through SaTC 231271 and the Vice President for Research and Partnerships and the Data Institute for Societ Challenges at the University of Oklahoma through a DISC Seed Grant Award.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
2. Pham, H.; Cheng, S. Non-Iterative Cluster Routing: Analysis and Implementation Strategies. *Appl. Sci.* **2024**, *14*, 1706. [CrossRef]
3. Zhang, H.; Cheng, S.; El Amm, C.; Kim, J. Efficient pooling operator for 3D morphable models. *IEEE Trans. Vis. Comput. Graph.* 2023, *early access*.
4. Zhang, H.; Tang, W.; Na, W.; Lee, P.Y.; Kim, J. Implementation of generative adversarial network-CLS combined with bidirectional long short-term memory for lithium-ion battery state prediction. *J. Energy Storage* **2020**, *31*, 101489. [CrossRef]
5. Soltani, M.; Bonakdar, A.; Shakourifar, N.; Babaei, R.; Raahemifar, K. Efficacy of location-based features for survival prediction of patients with glioblastoma depending on resection status. *Front. Oncol.* **2021**, *11*, 661123. [CrossRef] [PubMed]
6. Dong, Y.; Cheng, S.; Pang, T.; Su, H. Query-efficient black-box adversarial attacks guided by a transfer-based prior. *arXiv* **2022**, arXiv:2203.06560. [CrossRef]
7. Wei, Z.; Chen, J.; Zhang, H.; Jiang, L.; Jiang, Y. Adaptive temporal grouping for black-box adversarial attacks on videos. In Proceedings of the 2022 International Conference on Multimedia Retrieval, Newark, NJ, USA, 27–30 June 2022. [CrossRef]
8. Yang, J.; Jiang, Y.; Huang, X.; Ni, B.; Zhao, C. Learning black-box attackers with transferable priors and query feedback. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12288–12299.
9. Ilie, C.; Popescu, M.; Stefanescu, A. Evoba: An evolution strategy as a strong baseline for black-box adversarial attacks. In *Neural Information Processing*; ICONIP 2021; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 13110. [CrossRef]
10. Tram'er, F.; Zhang, F.; Juels, A.; Reiter, M.K.; Ristenpart, T. Stealing machine learning models via prediction APIs. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 601–618.
11. Zhang, X.; Fang, C.; Shi, J. Thief, Beware of What Get You There: Towards Understanding Model Extraction Attack. *arXiv* **2021**, arXiv:2104.05921.
12. Khaled, K.; Nicolescu, G.; Magalhães, F. Careful What You Wish For: On the Extraction of Adversarially Trained Models. In Proceedings of the 2022 19th Annual International Conference on Privacy, Security & Trust (PST), Fredericton, NB, Canada, 22–24 August 2022; pp. 1–10. [CrossRef]
13. Bai, Y.; Zeng, Y.; Jiang, Y.; Wang, Y.; Xia, S.; Guo, W. Improving query efficiency of black-box adversarial attack. *arXiv* **2020**, arXiv:2009.11508. [CrossRef]
14. Han, D.; Mulyana, B.; Stankovic, V.; Cheng, S. A survey on deep reinforcement learning algorithms for robotic manipulation. *Sensors* **2023**, *23*, 3762. [CrossRef]
15. Zhu, L.; Wang, T.; Li, J.; Zhang, Z.; Shen, J.; Wang, X. Efficient query-based black-box attack against cross-modal hashing retrieval. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–25. [CrossRef]
16. Han, D.; Huong, P.; Cheng, S. Enhancing Semantic Segmentation through Reinforced Active Learning: Combating Dataset Imbalances and Bolstering Annotation Efficiency. *J. Electron. Inf. Syst.* **2023**, *5*, 45–60. [CrossRef]
17. Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.; Lanctot, M.; Freitas, N. Dueling network architectures for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 1995–2003.

18. Fortunato, M.; Azar, M.G.; Piot, B.; Menick, J.; Osband, I.; Graves, A.; Mnih, V.; Munos, R.; Hassabis, D.; Pietquin, O.; et al. Noisy Networks for Exploration. *arXiv* **2019**, arXiv:1706.10295. [CrossRef]
19. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
20. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
21. Wang, C.; Ross, K. Boosting soft actor-critic: Emphasizing recent experience without forgetting the past. *arXiv* **2019**, arXiv:1906.04009.
22. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 506–519.
23. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
24. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
25. Truong, J.B.; Maini, P.; Walls, R.J.; Papernot, N. Data-free model extraction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4771–4780.
26. Xu, H.; Ma, Y.; Liu, H.C.; Deb, D.; Liu, H.; Tang, J.L.; Jain, A.K. Adversarial attacks and defenses in images, graphs and text: A review. *Int. J. Autom. Comput.* **2020**, *17*, 151–178. [CrossRef]
27. Pitropakis, N.; Panaousis, E.; Giannetsos, T.; Anastasiadis, E.; Loukas, G. A taxonomy and survey of attacks against machine learning. *Comput. Sci. Rev.* **2019**, *34*, 100199. [CrossRef]
28. Gong, Y.; Li, B.; Poellabauer, C.; Shi, Y. Real-time adversarial attacks. *arXiv* **2019**, arXiv:1905.13399.
29. Selvakkumar, A.; Pal, S.; Jadidi, Z. Addressing adversarial machine learning attacks in smart healthcare perspectives. In *Sensing Technology: Proceedings of ICST 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 269–282.
30. Alshahrani, E.; Alghazzawi, D.; Alotaibi, R.; Rabie, O. Adversarial attacks against supervised machine learning based network intrusion detection systems. *PLoS ONE* **2022**, *17*, e0275971. [CrossRef]
31. Ibitoye, O.; Abou-Khamis, R.; Matrawy, A.; Shafiq, M.O. The Threat of Adversarial Attacks on Machine Learning in Network Security–A Survey. *arXiv* **2019**, arXiv:1911.02621.
32. Sagduyu, Y.E.; Erpek, T.; Shi, Y. Adversarial machine learning for 5G communications security. In *Game Theory and Machine Learning for Cyber Security*; Wiley Press: Hoboken, NJ, USA, 2021; pp. 270–288.
33. Luo, Z.; Zhao, S.; Lu, Z.; Sagduyu, Y.E.; Xu, J. Adversarial machine learning based partial-model attack in IoT. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, Linz, Austria, 13 July 2020; pp. 13–18.
34. Chen, T.; Liu, J.; Xiang, Y.; Niu, W.; Tong, E.; Han, Z. Adversarial attack and defense in reinforcement learning-from AI security view. *Cybersecurity* **2019**, *2*, 1–22.
35. Oikarinen, T.; Zhang, W.; Megretski, A.; Daniel, L.; Weng, T.W. Robust deep reinforcement learning through adversarial loss. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 26156–26167.
36. Zhang, H.; Chen, H.; Boning, D.; Hsieh, C.J. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv* **2021**, arXiv:2101.08452.
37. Gleave, A.; Dennis, M.; Wild, C.; Kant, N.; Levine, S.; Russell, S. Adversarial policies: Attacking deep reinforcement learning. *arXiv* **2019**, arXiv:1905.10615.
38. Ali, H.; Al Ameedi, M.; Swami, A.; Ning, R.; Li, J.; Wu, H.; Cho, J.H. ACADIA: Efficient and Robust Adversarial Attacks Against Deep Reinforcement Learning. In Proceedings of the 2022 IEEE Conference on Communications and Network Security (CNS), Austin, TX, USA, 3–5 October 2022; pp. 1–9.
39. Ye, M.; Luo, J.; Zheng, G.; Xiao, C.; Xiao, H.; Wang, T.; Ma, F. MedAttacker: Exploring black-box adversarial attacks on risk prediction models in healthcare. In Proceedings of the 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Las Vegas, NV, USA, 6–8 December 2022; pp. 1777–1780.
40. Aghdam, H.H.; Gonzalez-Garcia, A.; Weijer, J.v.d.; López, A.M. Active learning for deep detection neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3672–3680.
41. Zhang, H.; Yuan, X.; Zhou, C.; Pan, S. Projective ranking-based gnn evasion attacks. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 8402–8416. [CrossRef]
42. Miller, B.; Kantchelian, A.; Afroz, S.; Bachwani, R.; Dauber, E.; Huang, L.; Tschantz, M.C.; Joseph, A.D.; Tygar, J.D. Adversarial active learning. In Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop, Scottsdale, AZ, USA, 7 November 2014; pp. 3–14.
43. Dineen, J.; Haque, A.A.U.; Bielskas, M. Reinforcement Learning for Data Poisoning on Graph Neural Networks. In Proceedings of the Social, Cultural, and Behavioral Modeling: 14th International Conference, SBP-BRiMS 2021, Virtual Event, 6–9 July 2021; pp. 141–150.
44. Debicha, I.; Debatty, T.; Dricot, J.M.; Mees, W.; Kenaza, T. Detect & reject for transferability of black-box adversarial attacks against network intrusion detection systems. In Proceedings of the International Conference on Advances in Cyber Security, Penang, Malaysia, 24–25 August 2021; pp. 329–339.
45. Yan, Z.; Guo, Y.; Liang, J.; Zhang, C. Policy-driven attack: Learning to query for hard-label black-box adversarial examples. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.

46. Yin, F.; Zhang, Y.; Wu, B.; Feng, Y.; Zhang, J.; Fan, Y.; Yang, Y. Generalizable black-box adversarial attack with meta learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**. [CrossRef]
47. Fang, Y.; Zeng, Y.; Li, B.; Liu, L.; Zhang, L. DeepDetectNet vs RLAttackNet: An adversarial method to improve deep learning-based static malware detection model. *PLoS ONE* **2020**, *15*, e0231626. [CrossRef]
48. Chen, K.; Guo, S.; Zhang, T.; Xie, X.; Liu, Y. Stealing deep reinforcement learning models for fun and profit. In Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, Hong Kong, China, 7–11 June 2021; pp. 307–319.
49. Tsingenopoulos, I.; Preuveneers, D.; Joosen, W. AutoAttacker: A reinforcement learning approach for black-box adversarial attacks. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Stockholm, Sweden, 17–19 June 2019; pp. 229–237.
50. Ren, Z.; Han, J.; Cummins, N.; Schuller, B. Enhancing transferability of black-box adversarial attacks via lifelong learning for speech emotion recognition models. In Proceedings of the 21st Annual Conference of the International Speech Communication Association, Shanghai, China, 25–29 October 2020.
51. Zhang, S.; Xie, X.; Xu, Y. A brute-force black-box method to attack machine learning-based systems in cybersecurity. *IEEE Access* **2020**, *8*, 128250–128263. [CrossRef]
52. Shukla, S.N.; Sahu, A.K.; Willmott, D.; Kolter, Z. Simple and efficient hard label black-box adversarial attacks in low query budget regimes. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual, 14–18 August 2021; pp. 1461–1469.
53. Cheng, M.; Singh, S.; Chen, P.; Chen, P.Y.; Liu, S.; Hsieh, C.J. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv* **2019**, arXiv:1909.10773.
54. Sener, O.; Savarese, S. Active Learning for Convolutional Neural Networks: A Core-Set Approach. *arXiv* **2017**, arXiv:1708.00489. [CrossRef]
55. Casanova, A.; Pinheiro, P.O.; Rostamzadeh, N.; Pal, C.J. Reinforced active learning for image segmentation. *arXiv* **2020**, arXiv:2002.06583.
56. Gao, W.; Li, X.; Wang, Y.; Cai, Y. Medical image segmentation algorithm for three-dimensional multimodal using deep reinforcement learning and big data analytics. *Front. Public Health* **2022**, *10*, 879639. [CrossRef] [PubMed]
57. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
58. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
59. Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
60. Zhou, S.Y.; Su, C.Y. A Novel lightweight Convolutional Neural Network, ExquisiteNetV2. *arXiv* **2021**, arXiv:2105.09008.
61. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report by the University of Toronto; University of Toronto: Toronto, ON, Canada, 2009.
62. Foret, P.; Kleiner, A.; Mobahi, H.; Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv* **2020**, arXiv:2010.01412.