IMPROVING LOW-BUDGET SEMI-SUPERVISED APPROACHES FOR MODEL EXTRACTION ATTACKS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by
Didem GENÇ

December 2024
İZMİR

We approve the thesis of **Didem GENÇ**

Examining Committee Members:

Professor Dr. Yalın BAŞTANLAR
Department of Computer Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Selma TEKİR
Department of Computer Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Özgü CAN
Department of Computer Engineering, Ege University

Professor Dr. Tolga AYAV
Department of Computer Engineering, İzmir Institute of Technology

Professor Dr. Yusuf Murat ERTEN
Department of Computer Engineering, İzmir University of Economics

9 December 2024

Professor Dr. Yalın BAŞTANLAR
Department of Computer Engineering
İzmir Institute of Technology

Dr. Emrah TOMUR
Ericsson Research

Professor Dr. Onur DEMİRÖRS
Head of the Department of
Computer Engineering

Professor Dr. Mehtap EANES
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my parents and my spouse for their unwavering support and encouragement throughout the completion of this thesis. Their understanding, patience, and belief in me gave me the strength to persevere during challenging times. This accomplishment would not have been possible without their constant encouragement and love.

I would like to extend my deepest gratitude to Assoc. Prof. Dr. Mustafa Özuysal and my co-supervisor Dr. Emrah Tomur. Dr. Özuysal has mentored me over many years, providing invaluable guidance and support that have shaped my academic journey. His mentorship and Dr. Tomur's insightful feedback and expertise have been crucial to completing this work. I am profoundly grateful to both for their support and dedication.

I am also sincerely thankful to my advisor, Prof. Dr. Yalın Baştanlar, for his close attention and guidance throughout this project, especially within a short time frame. His encouragement and expertise have greatly contributed to this research's quality, and I deeply appreciate his support.

My heartfelt thanks also go to my thesis committee members, Assoc. Prof. Dr. Selma Tekir and Assoc. Prof. Dr. Özgü Can, for their constructive feedback and valuable insights, significantly enhancing this study's rigor. I would also like to thank my jury members, Prof. Dr. Yusuf Murat Erten and Prof. Dr. Tolga Ayav, for their time, suggestions, and evaluation, all of which strengthened the final outcomes of this research.

ABSTRACT

IMPROVING LOW-BUDGET SEMI-SUPERVISED APPROACHES FOR MODEL EXTRACTION ATTACKS

Machine learning (ML) models are widely adopted across numerous fields due to their effectiveness; however, training high-accuracy models often involves substantial costs. To address this, Machine Learning as a Service (MLaaS) platforms provide cloud-based, black-box models accessible through APIs (Application Programming Interface), which raises security concerns like model extraction attacks (MEA). An MEA seeks to replicate a cloud-deployed ML model solely using black-box queries. This thesis proposes a cost-effective and accurate model extraction attack where unlabeled data is readily available, but labeled data is costly. Existing literature suggests strategies such as creating synthetic datasets, selecting data via active learning, and using semi-supervised learning. This thesis instead adopts a self-supervised learning approach for attacking a black-box model via an API. The method assumes the adversary access to a large pool of unlabeled data, which is used to train a self-supervised SimCLR model. A subset of the unlabeled data is queried through the target model to create a transfer dataset, which fine-tunes a multi-layer perceptron (MLP) added to the SimCLR encoder, forming the baseline substitute model. To enhance the substitute model accuracy, automatic labeling assigns high-confidence predictions directly as labels to the unlabeled data, while low-confidence samples are labeled based on similarity to target-labeled data. Incorporating high-entropy data during training enables the model to capture complex patterns and increase data diversity, ultimately enhancing the substitute model's accuracy. The method's effectiveness is demonstrated through experiments on CIFAR-10 and SVHN datasets.

ÖZET

MODEL ÇIKARMA SALDIRILARI İÇİN DÜŞÜK BÜTÇELİ YARI-DENETİMLİ YAKLAŞIMLARIN İYİLEŞTİRİLMESİ

Makine öğrenimi (ML) modelleri, etkinlikleri nedeniyle birçok alanda yaygın olarak kullanılmaktadır; ancak yüksek doğruluğa sahip modelleri eğitmenin maliyeti de yüksektik. Bu bağlamda, MLaaS (Machine Learning as a Service) platformları, API'ler aracılığıyla erişilebilen bulut tabanlı kara kutu modeller sunarak, model çalma saldırıları gibi güvenlik sorunlarını gündeme getirmektedir. Model çalma saldırıları, bulutta konuşlandırılmış bir makine öğrenimi modelini yalnızca kara kutu sorgulamalarıyla kopyalamayı amaçlamaktadır. Bu tez çalışmasında, etiketlenmemiş veriye erişimin kolay olduğu ancak etiketli verinin maliyetli olduğu senaryolarda, maliyet etkin ve yüksek doğruluklu bir model çalma saldırısı geliştirilmiştir. Literatürde sentetik veri setleri oluşturma, doğal veri setlerinden aktif öğrenme ile veri seçme ve yarı denetimli öğrenme gibi stratejiler önerilmektedir. Bu çalışmada ise, API üzerindeki kara kutu bir modele saldırmak için öz-denetimli öğrenen modellerden faydanılması önerilmiştir. Bu yöntemde, saldırganın geniş bir etiketlenmemiş veri havuzuna erişimi olduğu varsayılmakta ve bu veri, öz-denetimli SimCLR modelini eğitmek için kullanılmaktadır. Etiketsiz veri kümesinden belirli bir alt küme seçilir ve hedef modele sorgular gönderilerek bu veriler etiketlenir. Bu işlem sonucunda transfer veri seti oluşturulur. İlk ikame model, transfer veri setiyle SimCLR encoder'ına eklenen bir çok katmanlı algılayıcı (MLP)'nın ince ayar yapılarak eğitilmesi ile elde edilir. İkame modelin doğruluğunu artırmak için kalan etiketlenmemiş verilere otomatik etiketleme uygulanır; yüksek güvenli çıktılar doğrudan etiket olarak kullanılırken, düşük güvenli çıktılar hedef modelin etiketlediği örneklerle olan benzerliğe göre etiketlenir. Bu süreç, modelin karmaşık örüntüleri öğrenmesini ve veri çeşitliliğini artırmasını sağlayarak ikame modelin doğruluğunu hedef modele yaklaştıracak şekilde artırır. Önerilen methodun verimliliği CIFAR10 ve SVHN datasetleri üzerinde deneyler yapılarak verilmiştir.

To my lovely children; Eymen and Kerem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Machine learning (ML) models have been widely adopted in recent years in diverse industries, revolutionizing areas such as healthcare, finance, and autonomous systems. This popularity is largely due to their ability to learn complex patterns from large datasets, enabling them to make highly accurate predictions and solve problems that were previously difficult or impossible to tackle. However, despite their success, creating high-performing ML models is costly and labor-intensive. Training these models often requires vast amounts of data, significant computational resources, expert knowledge, and a long development cycle, making them valuable intellectual properties.

Therefore, companies and researchers have increasingly relied on machine learning as a service (MLaaS) platforms to reduce the financial and resource demands of training efforts. These platforms offer pre-trained models hosted on the cloud, accessible to clients through Application Programming Interfaces (APIs). This model-sharing approach has become highly popular for its convenience, allowing users to leverage powerful ML models without internal development or maintenance procedures. MLaaS platforms typically follow a pay-per-query basis, where users submit input data to a black-box system and receive corresponding predictions without gaining any insight into the model's architecture, hyperparameters, or training data. Although this service model benefits clients by reducing computational costs and speeding up deployment, it also introduces serious security and privacy challenges.

One of the primary concerns in this black-box setting is the vulnerability of ML models to model extraction attacks (MEA). A model extraction attack occurs when an adversary, through repeated queries to a target model, attempts to reconstruct a copy of the model's functionality without access to its internal details. A model extraction attack aims to replicate the target model's decision-making process or general functionality, allowing the adversary to create a substitute model that performs similarly to the original. Moreover, the substitute model can be used for various malicious purposes, ranging from the theft of intellectual property (adversary can deploy a similar model locally without incurring development costs) to enabling more dangerous attacks, such as evasion attacks (Papernot

et al. 2017), where adversarial examples are crafted to deceive the model, and membership inference attacks, which aim to determine whether specific data points were part of the model's training set.

Model extraction can have severe consequences for businesses that depend on proprietary ML models as key assets. A substitute model can reduce the business value of the original model by providing similar functionality at a lower cost, potentially weakening its market position. Additionally, a substitute model can be used as a white-box model to expose weaknesses in the target model, paving the way for further attacks, such as privacy breaches and security issues. Given the significant risks posed by model extraction attacks, it becomes crucial to thoroughly investigate their mechanisms to better understand their impact and inform the development of robust defenses.

Understanding and studying model extraction attacks (MEAs) is critical for advancing the field of machine learning security. Although defenses are essential for mitigating potential risks, their development must be informed by a deep understanding of the attack mechanisms to which they are designed to counteract. By focusing on attacks, this research aims to uncover inherent vulnerabilities in machine learning models, particularly those deployed in black-box settings such as MLaaS platforms. Studying attacks provides valuable information on how adversaries exploit these vulnerabilities, enabling the identification of weaknesses that might otherwise go unnoticed. Furthermore, the development of novel attack strategies can act as a stress test for existing defenses, highlighting their limitations and guiding the creation of more robust protective measures. Without a comprehensive understanding of attacks, defensive strategies risk being incomplete, overly reliant on assumptions, or ineffective against evolving adversarial tactics. Accordingly, this thesis prioritizes the study of model extraction attacks to contribute to a more holistic understanding of the threat landscape, ultimately aiding in the design of effective and efficient defense mechanisms in future work. This proactive approach ensures that defenses are not only reactive, but also resilient against the sophisticated and resourceful techniques employed by attackers.

1.1. Problem Definition

Model extraction attacks have emerged as a critical concern in the domain of machine learning security, particularly for Machine Learning-as-a-Service (MLaaS) platforms (Oliynyk, Mayer, and Rauber 2023), (Genç, Özuysal, and Tomur 2023). In black-box settings, adversaries aim to replicate the decision-making process of a target model without access to its internal architecture, hyperparameters, or training data. The attacker collects input-output pairs by querying the target model via an API and uses these data to train a local substitute model. A major challenge in such attacks is optimizing the **query budget**, as the number of queries directly impacts the cost of the attack and the likelihood of detection. Traditional methods for creating substitute models are heavily based on supervised learning (Tramèr et al. 2016),(Papernot et al. 2017),(Pal et al. 2020), which requires a substantial amount of labeled data. This creates a trade-off: while more queries provide better training data for the substitute model, they also increase the cost and risk of detection, limiting the practicality of the attack.

To address the challenge of minimizing query costs, prior research has explored synthetic query generation, active learning, and reinforcement learning to optimize query selection (Genç, Özuysal, and Tomur 2023). Although these strategies improve efficiency, they often fail to achieve substitute model accuracy comparable to the target model. Semi-supervised approaches, such as those proposed by Jagielski et al. (Jagielski et al. 2020), have demonstrated promising results in reducing the query budget while maintaining high substitute model accuracy. However, existing semi-supervised methods, such as MixMatch (Berthelot et al. 2019), rely on complex techniques such as consistency regularization and entropy minimization, making them challenging to implement without significant computational resources and technical expertise. This creates a gap for developing more accessible and efficient methods that balance query efficiency and substitute model performance, particularly in adversarial scenarios where resources are limited.

1.2. Aim of Thesis

This thesis aims to develop an innovative and resource-efficient approach to model extraction attacks (MEAs) in black-box MLaaS settings by leveraging contrastive self-

supervised learning frameworks. Contrastive self-supervised learning is chosen as the foundation for this work because it effectively addresses the critical trade-offs in MEAs. Unlike traditional supervised approaches that require a significant amount of costly labeled data, contrastive self-supervised learning relies on abundant unlabeled data to learn high-quality, generalizable representations. This reduces the dependency on labeled datasets, thereby lowering the query budget needed for training a substitute model. In addition, self-supervised pre-trained models are both widely available and easy to fine-tune, making them an attractive option for adversaries with limited computational resources or technical expertise. By utilizing the pre-trained representations derived from these models, the proposed method seeks to enhance the efficiency and practicality of model extraction attacks significantly. This approach effectively addresses the primary limitations of existing methods in the literature, offering a strategy for model extraction that is not only cost-effective but also straightforward and resource-efficient.

The core research question driving this thesis is: How can contrastive self-supervised learning be leveraged to perform efficient and effective model extraction attacks in black-box MLaaS settings while minimizing the query budget and achieving high substitute model accuracy? By addressing this question, the thesis seeks to resolve the trade-offs between query budget, attack efficiency, and model performance that have plagued existing methods. The choice of contrastive self-supervised learning not only aligns with the need for a practical and cost-effective solution but also highlights the potential of these frameworks in advancing the state-of-the-art in model extraction attacks. The research findings aim to provide new insights into adversarial strategies while also offering valuable guidance for developing more robust defenses against such attacks.

1.3. Contribution of the Thesis

This thesis presents a novel approach to model extraction attacks by leveraging contrastive self-supervised learning methods to minimize query budget requirements while maintaining high substitute model accuracy. The proposed methodology integrates SimCLR-based feature learning with pseudo-labeling strategies, establishing a cost-effective and efficient framework for constructing accurate substitute models in black-box attack scenarios. The contributions of this work are summarized as follows:

- A new model extraction approach based on the SimCLR contrastive self-supervised learning framework has been developed to extract meaningful representations from unlabeled data. This method reduces the need for large labeled datasets, making the attack more cost-effective and practical for real-world applications.

- The proposed method has been shown to outperform existing techniques, such as MixMatch (Jagielski et al. 2020), particularly in low-query scenarios. It achieves state-of-the-art results by efficiently balancing query budget and model performance.

- A comprehensive threat model has been outlined to better understand the risks associated with model extraction attacks. This includes an analysis of adversarial capabilities, attack surfaces, and the potential consequences for machine learning systems, offering a foundation for future work on defensive strategies.

- Extensive experiments and ablation study have been conducted to evaluate the effectiveness of the proposed approach. The results highlight its scalability and adaptability, demonstrating its potential to address the limitations of current model extraction techniques.

- The proposed method has been applied to the medical domain using the Chest X-ray dataset, demonstrating its feasibility in real-world healthcare applications. The results validate the approach's ability to extract high-performing substitute models even in domains where labeled data is scarce, highlighting its potential impact on medical AI security and robustness.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1. Model Extraction Attack

A model extraction attack (MEA) occurs when an attacker attempts to build a machine learning model that closely matches, or even exceeds, the performance of a target model with less effort than it would take to train a new model from scratch. The attacker's main objective is to replicate the decision-making process or functionality of the target model by repeatedly querying it. Essentially, the attacker uses the target model as a guide. Tramer et al. (Tramèr et al. 2016) introduced one of the earliest model extraction attacks, targeting traditional models such as logistic regression to extract parameters and trying to recreate the decision boundaries of models such as support vector machines (SVM) and multilayer perceptrons (MLP).

In this attack, the attacker starts by sending a set of inputs to the target model, which is often hosted on an API, and collects the resulting predictions as output. These input-output pairs then serve as data for training a new model, often called the substitute model, which is intended to mirror the behavior of the original model. This extraction process takes place in a black box setting, which means that the attacker does not have direct information about the model's internal parameters, architecture, hyperparameters, or training data distribution. The only interaction with the target model is through querying inputs and receiving outputs. Figure 2.1 illustrates the process of a model extraction attack, and frequently used terminology related to model extraction is provided in Table 2.1 below.

Definition 1. *Target Model*

The target model is the machine learning model that the adversary aims to replicate. Typically deployed as a cloud-based MLaaS (Machine Learning as a Service) solution, it provides users with black-box access, meaning its internal structure remains hidden. This target model can be any of the widely used machine learning models discussed in the literature. The target model may employ linear or non-linear supervised algorithms for discriminative tasks, such as logistic regression, support vector machines (SVM), or neural

networks (NN). In such cases, a target model f_T receives an input $x \in \mathbb{R}^n$ and produces an output $f_T(x) = y \in \mathbb{R}^K$. Generative models, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), are also used as target models; in these instances, the model learns the training data distribution and generates images resembling the learned samples, with the adversary aiming to replicate this learned distribution. The target model is sometimes referred to as the "victim" or "secret" model in the literature, but we will consistently refer to it as the *target model* f_T.

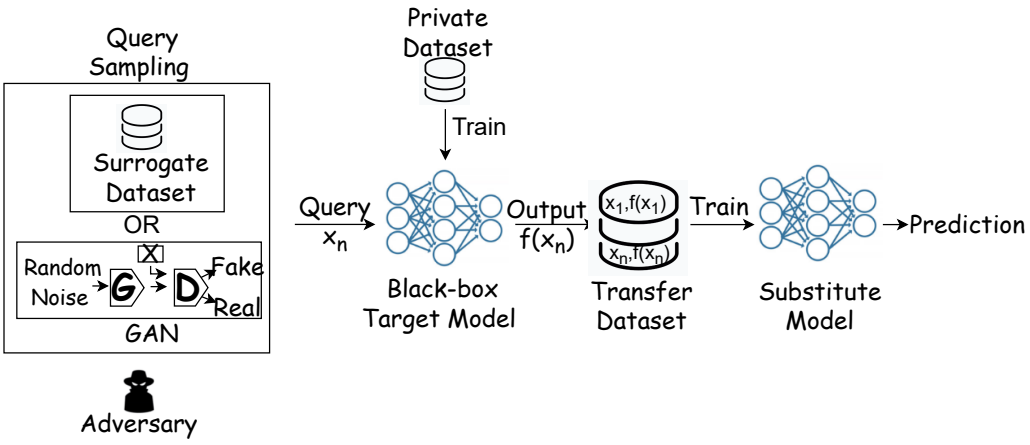


Figure 2.1. Model Extraction Attack Overview

Definition 2. *Substitute Model*

The substitute model is a clone or copy of the target model. The adversary aims to construct a local model that behaves as functionally similar as possible to the target model. The substitute model, denoted as f_S, is trained on a transfer dataset. In the literature, this model is also referred to as a knock-off (Orekondy, Schiele, and Fritz 2019), surrogate, clone, or extracted model. We will refer to this model as the *substitute model* f_S (Tramèr et al. 2016; Jagielski et al. 2020).

Definition 3. *Transfer Dataset*

In a model extraction attack, the transfer dataset serves as a link between the target and

substitute models. The transfer dataset is denoted by $D_A = \{x_i, f_T(x_i)\}_{i=0}^{N}$, where N represents the query budget. It consists of input-output pairs obtained by querying the target model. The attacker uses this dataset to train a local model that imitates the target model's behavior, effectively treating the target model as a source of labels. This approach reduces the need for large labeled datasets, allowing the attacker to build a substitute model that closely approximates the functionality of the original model at a reduced cost.

Definition 4. *Private Dataset*
The private dataset $D_P = \{x_i, y_i\}_{i=0}^{M}$ refers to the training dataset of the target model. This dataset is proprietary and confidential, and it is generally assumed that the adversary has no access to it nor even to its underlying distribution. In the literature, the private dataset is also referred to as the *secret dataset*.

Definition 5. *Surrogate Dataset*
In a model extraction attack, the choice of query samples is crucial, as the goal is to extract the maximum information from the model with a minimal number of queries. The dataset from which these queries are sampled is called the surrogate dataset. Ideally, the surrogate dataset should have a distribution that closely resembles the private dataset. This dataset may consist of publicly available datasets, such as CIFAR10, MNIST, or ImageNet, or it can be crafted manually using methods like GANs. This dataset is also known as the thief, adversary, or surrogate dataset in the literature. We will refer to it as the *surrogate dataset*, D_S.

Definition 6. *Accuracy Extraction*
The goal of accuracy extraction is to maximize the task accuracy of the substitute model, defined as $\max\,[\text{argmax}(f_s(x_i)) = y_i]$.

Definition 7. *Fidelity Extraction*
The goal of fidelity extraction is for the substitute model to closely match the target model, represented by $\max\,[\text{argmax}(f_t(x_i)) = \text{argmax}(f_s(x_i))]$. This ensures that any errors made by the substitute model are consistent with those of the target model.
For clarity, Table 2.1 provides a summary of the terminology discussed above.

Table 2.1. Summary of Model Extraction Attack Terminology

Notation	Terminology	Berief Definition
f_T	Target Model	Model under attack
f_S	Substitute Model	Locally constructed model
D_P	Private Dataset	Target model's training dataset
D_S	Surrogate Dataset	Dataset where the queries are selected
D_A	Transfer Dataset	Substitute model's training dataset

2.2. Literature Review

The study of model extraction attacks has evolved considerably since the concept was first introduced. Early works were primarily concerned with adversarial attacks on simple models, particularly linear classifiers, but recent research has expanded the scope to include deep learning architectures, generative models, and advanced query techniques. This literature review examines the evolution of model extraction techniques from the query source perspective.

The roots of model extraction can be traced back to the work of Lowd and Meek (Lowd and Meek 2005), who introduced the concept of adversarial learning for linear binary models. Their attack, which targets reproducing the model's weights, operates by identifying sign witness pairs, which are two samples that differ in only one feature and belong to different classes. The attack works by adjusting the feature values of a sample and performing a line search to extract the model's weight values accurately. This allows the adversary to build a replication of the target model that performs the same as the original. To carry out this attack, the adversary must know the target model's architecture and have two data points; one positive and one negative. Although the attack can perfectly extract the model's weights, it has a major drawback: inefficiency. It requires at least 11 queries per parameter, which makes it unsuitable for larger models. This limitation reduces the attack's ability to scale, especially for more complex models.

Later, (Tramèr et al. 2016) proposed an attack that efficiently extracts the learned parameters from models such as Multi-class Logistic Regression and Multi-Layer Perceptron, requiring fewer queries. They sent data samples to the target model and collected

the corresponding outputs. Using these outputs, they created a system of equations that described the relationship between the inputs and the model parameters. By solving these equations, they could recover the values of the model's parameters. Although this attack is more efficient than Lowd and Meek's method, it still requires knowledge of the target model's architecture and access to data samples for querying. Later, (Reith, Schneider, and Tkachenko 2019) extended this approach to extract parameters from Support Vector Regression (SVR) models with linear or quadratic kernels.

The attacks discussed so far can be classified as equation-solving attacks. Another attack approach, a retraining attack, is first proposed by (Tramèr et al. 2016). In this type of attack, input-output pairs $(x, f_T(x))$ are obtained by repeatedly querying the target model, allowing them to train a local substitute model that approximates the target model's decision boundary. Retraining attack is the predominant strategy for model extraction. Unlike equation-solving, retraining does not rely on confidence scores, making it applicable to a broader range of models, including those that only return class labels. Tramèr's method remains a foundational technique for subsequent model extraction attack (MEA) research, particularly in black-box settings where the target model's internal parameters are inaccessible.

The source of queries plays a key role in model extraction attacks. Samples close to the target model's decision boundary provide more valuable information, making them critical for extracting the model effectively. To achieve this, adversaries aim to identify or generate informative samples while minimizing the number of queries needed. Adversaries have two main strategies to generate these queries. One option is to use publicly available surrogate datasets, which are natural datasets containing data types similar to those of the target model's training set. The other approach is to create synthetic queries specifically designed to probe the target model. Both methods come with their own advantages and challenges. For example, surrogate datasets can save time, but may not be perfectly aligned with the target model's distribution. On the other hand, synthetic queries can be tailored to the task, but may require significant computational effort and an increased query budget. In the following subsections, we examine the studies proposed in the literature regarding natural dataset sampling and synthetic query generation in detail.

2.2.1. Synthetic Query Generation

Many existing model extraction attacks assume that the adversary has access to a surrogate dataset, which mirrors the private data used to train the target model. However, this assumption limits the application of these techniques, especially for valuable models trained on rare or difficult-to-access datasets. To overcome this, data-free MEAs assume that attackers lack access to private datasets and instead train generative models to produce synthetic samples for querying. (Kariyappa, Prakash, and Qureshi 2021) were the first to propose this approach, using a Generative Adversarial Network (GAN) to create samples. Since black-box attacks do not allow direct access to gradients, they employed zeroth-order gradient approximation methods to estimate directional derivatives, which, although effective, significantly increased the query budget. Similarly, (Truong et al. 2021) enhanced this approach by replacing KL-divergence loss with L_1 -norm loss, which helped avoid the vanishing gradient problem and ensured the substitute model closely mimicked the target model's outputs, although this still required a high number of queries. To mitigate this issue, (Miura, Shibahara, and Yanai 2024) introduced a gradient-based explanation technique called Vanilla Gradient. By analyzing the effect of each pixel on the target model's confidence scores, they reduced the reliance on gradient approximations. Although this streamlined approach showed similar results to earlier methods, it did not significantly decrease the required queries.

(Gong et al. 2021) explored a different strategy by applying model inversion techniques to MEAs. They viewed the process as an encoder-decoder problem, where the target model's predictions were treated as encoded vectors that could be inverted to generate representative samples. By selecting high-confidence samples using a core-set algorithm, they effectively reduced the query budget and achieved high accuracy and fidelity in the substitute model. Moreover, this approach generated queries that mimicked normal data distributions, making it difficult for defense mechanisms like PRADA (Juuti et al. 2019) to detect the attack. Another notable contribution came from (Papernot et al. 2017), who introduced Jacobian-based dataset enhancement (JBDA). They began with a small set of initial samples labeled by the target model and iteratively generated synthetic samples by modifying inputs based on the Jacobian matrix's directional variations. This approach ensured efficient augmentation without exponential query growth and allowed the

substitute model to craft adversarial samples transferable to the target model, amplifying the attack's impact.

As a conclusion, these efforts highlight the diverse methods used to improve MEAs. While synthetic data generation has made it possible to mount effective attacks even in constrained scenarios, challenges such as query efficiency, computational costs, and model fidelity remain drawbacks of this approach.

2.2.2. Natural Query Sampling

Three methods are commonly used to select the best query samples: active learning, reinforcement learning, and random selection.

The simplest query sampling approach among the natural datasets is to simply select samples uniformly at random. Shi *et al.* (Shi, Sagduyu, and Grushin 2017) mounted an MEA on naive Bayes, Support Vector Machine (SVM), and Deep Neural Network (DNN) models in the text classification domain by randomly selecting the queries among private datasets. Likewise, (Tramèr et al. 2016) and (Orekondy, Schiele, and Fritz 2019) also used random uniform sampling to evaluate their studies.

Among the sampling techniques, active learning is the most extensively studied. Chandrasekaran *et al.* (Chandrasekaran et al. 2020) demonstrated the application of two active learning methods, namely probably approximately correct (PAC) and query synthesis (QS), for stealing models such as Decision Trees (DTs), Random Forests (RFs), Linear Binary Models (LBMs), and Support Vector Machines (SVMs). They also used the extended adaptive training (EAT) approach to sharply decrease the query budget (5x-224x) compared to (Tramèr et al. 2016) for kernel support vector machines. One way to sample queries is to first select an initial subset randomly and label this set by querying f_T. Later, train an initial substitute model, f_S, on these labeled data, which can be exploited to sample queries using an active learning approach. Tramèr *et al.* (Tramèr et al. 2016) proposed an adaptive retraining approach driven by active learning that selects samples that f_S is least certain about. Similarly, Pal *et al.* (Pal et al. 2020) investigated active learning techniques, including uncertainty sampling, K-center, and DeepFool-based Active Learning (DFAL), to pinpoint the most informative samples, which were then utilized in their "Activethief" attack.

(Orekondy, Schiele, and Fritz 2019) were the first to apply reinforcement learning to select optimal samples in their study, called the Knockoff attack. In this method, a learning policy is updated at each sampling time according to a predefined reward function.

Alternatively, the adversary may only query a subset of the surrogate dataset and use semi-supervised learning to exploit the remaining unlabeled part. Jagielski *et al.* (Jagielski et al. 2020) exploited two semi-supervised learning methods for accuracy extraction; rotation loss and MixMatch. They showed that only querying 4000 samples and using the semi-supervised approach on the remaining unlabeled data improve upon labeling the whole dataset when the dataset size is small enough. Consequently, Table 2.2 summarizes various studies focusing on sampling queries from natural surrogate datasets, specifically within the image domain.

Table 2.2. Model Extraction Attack Literature Which Uses Natural Dataset While Conducting Attack. Model abbreviations; LR: Logistic Regression, MLR: Multilayer Perceptron, CNN: Convolutional Neural Network, DNN: Deep Neural Network, k-SVM: Kernel-Support Vector Machine, DT: Decision Tree, NB: Naive Bayes.

Reference	**Target Model**	**Surrogate Dataset**	**Private Dataset**	**Query Budget**
Tramer et al.	LR/MLR/NN k-SVM/DT	Digits German Credit Adult Steak Survey	Digits German Credit Adult Steak Survey	650 1150 1485 4013
Chandrasekaran et al.	k-SVM/DT	Breast Cancer Adult Mushroom Diabetes	Breast Cancer Adult Mushroom Diabetes	119 48 1001 166
Orekondy et al.	CNN	ILSVRC OpenImages	Caltech CUBS200 Indoor Diabetic	60k
Pal et al.	CNN	ILSVRC	MNIST Cifar10 GTSRB	30k
Jagielski et al.	CNN	ImageNet Cifar10 SVHN	Social Media Cifar10 SVHN	4k

CHAPTER 3

SEMI-SUPERVISED MODEL EXTRACTION ATTACK

Self-supervised learning (SSL) has widespread adoption for its ability to reduce the need for labeled data by utilizing pseudo-labels to learn representations applicable to downstream tasks. Contrastive learning (CL) is a discriminative method that focuses on grouping similar samples together and pushing different samples apart by using a similarity metric to determine how close two embeddings are. By combining contrastive and self-supervised learning, a strong framework is produced that allows models to acquire meaningful and transferable representations from unlabeled data, which makes them extremely useful for a range of applications. Several contrastive self-supervised learning models, including SimCLR (Chen et al. 2020), SwAV (Caron et al. 2020), BYOL (Grill et al. 2020), MoCo (He et al. 2020), have been proposed over the years. Each of these models offers unique methods for learning representations from unlabeled data (Bastanlar and Orhan 2022). Among them, the SimCLR model is widely adopted in the literature for its reliability, high accuracy, and robust performance. Additionally, it provides access to numerous pre-trained models on various datasets. Therefore, the SimCLR model was selected for the application in this thesis.

In the SimCLR framework, the model learns to recognize two different views of the same image as similar while distinguishing them from views of other images. This framework heavily depends on random augmentations, such as cropping, flipping, and color changes, to each image while generating the views of the images. Once the image views are obtained, they are fed into a neural network, such as a ResNet encoder, which produces representations. These representations are projected into a latent space, and the model is trained to make similar images close to each other and different images far apart in this space. The model learns meaningful representations of the unlabeled data through this training process, which can later be fine-tuned for specific tasks like classification, object detection, or segmentation.

Semi-supervised training is one of the learning techniques that successfully benefits the utilization of a large amount of unlabeled data and a small amount of labeled data in the training process. This method aims to exploit supervised and unsupervised learning,

enabling the labeled data to provide the model with explicit direction while the unlabeled data helps to reveal more general patterns and representations in the domain. Therefore, semi-supervised learning is an efficient and practical approach for scenarios where labeled data is costly to acquire or insufficient within the domain, while unlabeled data is readily available and easily accessible.

In this thesis, we processed the unlabeled data using the general representations that the self-supervised model had learned and then mapped these representations to labels using the transfer dataset. In other words, the labeled transfer dataset produced from the outputs of the target model is used to fine-tune the self-supervised model, which is first trained on unlabeled data to learn general representations. The fine-tuning procedure of a SimCLR model is given in Figure 3.1. Consequently, this approach is classified as semi-supervised learning, as it combines a large volume of unlabeled data with a small amount of labeled data. However, since the labeled and unlabeled data originated from the same dataset, it cannot be categorized as transfer learning.

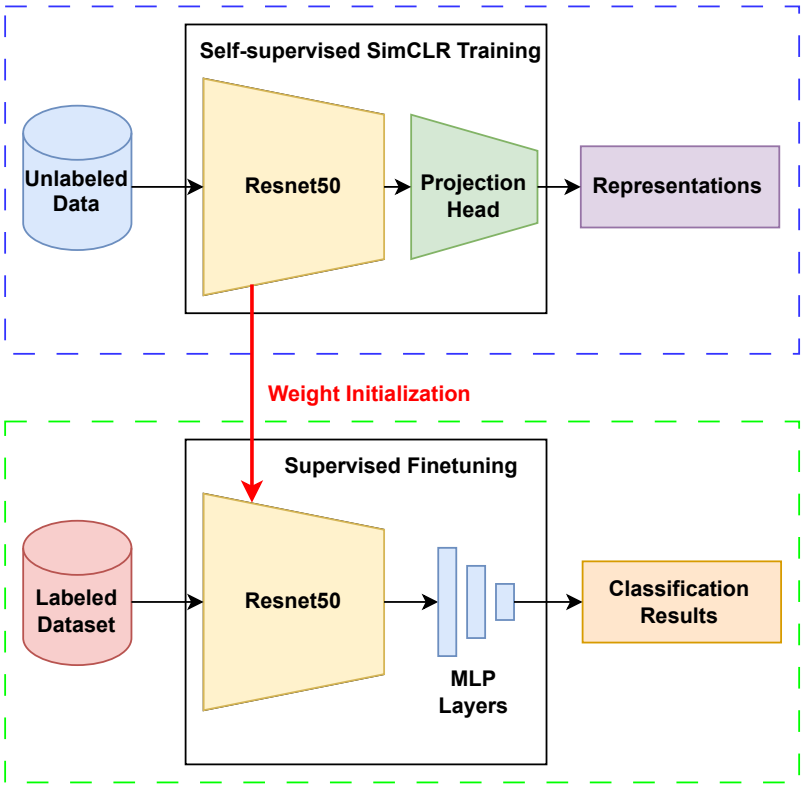


Figure 3.1. Fine-tuning Procedure of Self-supervised SimCLR Model

3.1. Methodology

This study introduces a novel technique for extracting a black-box machine learning model using self-supervised contrastive learning with a low query budget. The approach is divided into six primary stages, each contributing to maximizing the success of model extraction within given query limitations.

To perform the proposed model extraction attack, the adversary first acquires a large number of unlabeled samples from the same problem domain as the target model. This dataset is utilized to train a self-supervised model, as illustrated in Step A of Figure 3.2. In this step, the SimCLR framework has been utilized as the self-supervised model for the reasons explained in detail above. The self-supervised model trained and obtained after this step will later be used in the subsequent stages to analyze the image similarities and to support the development of the substitute model.

In the second stage, the adversary uses active learning techniques to wisely select a small subset from the larger unlabeled dataset. The graph cut function, a submodularity-based method (Iyer et al. 2021), is employed in active learning for its effectiveness in selecting informative samples from the unlabeled dataset. Next, the obtained subset is labeled by submitting repeated queries to the target model, resulting in the creation of what we call the "transfer dataset." This process refers to step B in Figure 3.2. Once the transfer dataset is created, a linear evaluation is conducted to approximate the target model: a 3-layer multi-layer perception (MLP) model is placed on top of the SimCLR encoder, followed by fine-tuning the complete network using the transfer dataset in a supervised manner. This process, depicted in Step C of Figure 3.2, produces the "semi-supervised baseline model."

The accuracy of the obtained substitute baseline model is found to be lower than that of the target model. This is due to the limited amount of data in the transfer dataset, which arises from the query budget constraints. To address this issue, we aim to enlarge the labeled dataset utilized during the fine-tuning process in Step C. Therefore, we introduce two pseudo-labeling processes, which are shown in Steps D and E in Figure 3.2, confidence-based and similarity-based pseudo-labeling, respectively. Confidence-based sampling relied on labeling the sample according to the output generated by the semi-supervised baseline model. The class labels corresponding to output confidence scores that exceed a

specified threshold are considered the pseudo-labels for the respective samples. In other words, since the labeling model has already been trained on the same unlabeled data in a self-supervised way, the low-entropy samples on which the model is confident are considered appropriate for direct labeling. These low-entropy samples are trustworthy candidates for pseudo-label assignment without the need for extra processing since they reflect data points for which the model has high certainty. Eventually, the confidence-based pseudo-labeling approach, shown in Step D of Figure 3.2, ensures that only reliable predictions are included in the expanded dataset, maintaining the quality of the labeled data. The dataset generated after this step is referred to as the ”pseudo-labeled transfer dataset.” Using only the examples that the semi-supervised substitute model is confident about can reduce the diversity in the pseudo-labeled dataset. To tackle this, we used the similarity-based pseudo-labeling approach to include samples with confidence scores below the threshold in the pseudo-labeled transfer dataset. Accordingly, using the self-supervised model's encoder, representations of low-confidence samples are extracted and compared to those in the pseudo-labeled transfer dataset using a distance metric. Cosine-similarity is used to determine the distance between the representations of the images. The concept of cosine similarity is based on the idea that in a given feature space, similar vectors will be positioned closer to each other, whereas dissimilar vectors will be located farther apart. Based on this, each low-confidence sample is assigned the label of its most similar counterpart in this labeled dataset. This technique enriches the dataset by including diverse and challenging examples, improving the final substitute model's generalization capabilities. An illustration of similarity-based pseudo-labeling can be found in Step E. By the end of this step, all samples in the unlabeled dataset are assigned pseudo-labels, resulting in a final transfer dataset.

Finally, the adversary uses the expanded labeled dataset, which includes high-confidence pseudo-labeled samples and similarity-based pseudo-labeled samples, to train the final substitute model. As shown in Step F of Figure 3.2, the self-supervised encoder is further fine-tuned, and an additional multi-layer perceptron (MLP) is attached to complete the model architecture. This final training phase leverages the enriched dataset to produce a substitute model that closely replicates the performance and behavior of the target model. By combining self-supervised learning, active learning, and pseudo-labeling techniques, this approach effectively reduces the query budget while achieving high accuracy in

the extracted substitute model, addressing key limitations of existing model extraction methods.

An important feature of this method is the intentional labeling of low-confidence samples (those below the predefined threshold) to increase dataset diversity. Although high-confidence predictions typically come from images the model has 'memorized', low-confidence samples represent unfamiliar data, introducing greater diversity. Integrating confidence-based and similarity-based labeling produces a more varied labeled dataset, which enables the substitute model to generalize beyond familiar patterns.

3.2. Experimental Results

This section presents the experimental results of the proposed methodology. However, it is important to note the following details in advance. The experiments in this section were conducted using the CIFAR-10 dataset (Krizhevsky, Hinton, et al. 2009) and the SVHN dataset (Netzer et al. 2011). The CIFAR-10 dataset consists of color images with a resolution of 32x32 pixels, divided into 10 classes, such as airplanes, cars, and animals, with 50,000 training samples and 10,000 test samples. Similarly, the SVHN dataset contains images of house numbers captured from real-world street views, also with a resolution of 32x32 pixels. It is divided into 10 classes representing the digits 0 through 9 and includes more than 73,000 labeled training samples, 26,000 test samples, and 531,131 additional samples. These datasets provide diverse and well-known benchmarks for evaluating image classification models.

Similar to existing studies in the literature, this thesis assumes that the adversary can access data from the same problem domain as the target model's private dataset. For this purpose, the entire CIFAR-10 and SVHN datasets were used as the unlabeled surrogate datasets. However, to prevent any overlap with the target model's training data, queries for the CIFAR-10 dataset were selected exclusively from its test set, while for the SVHN dataset, the additional set was used. This setup guarantees that the queried samples are distinct from those in the target model's training dataset.

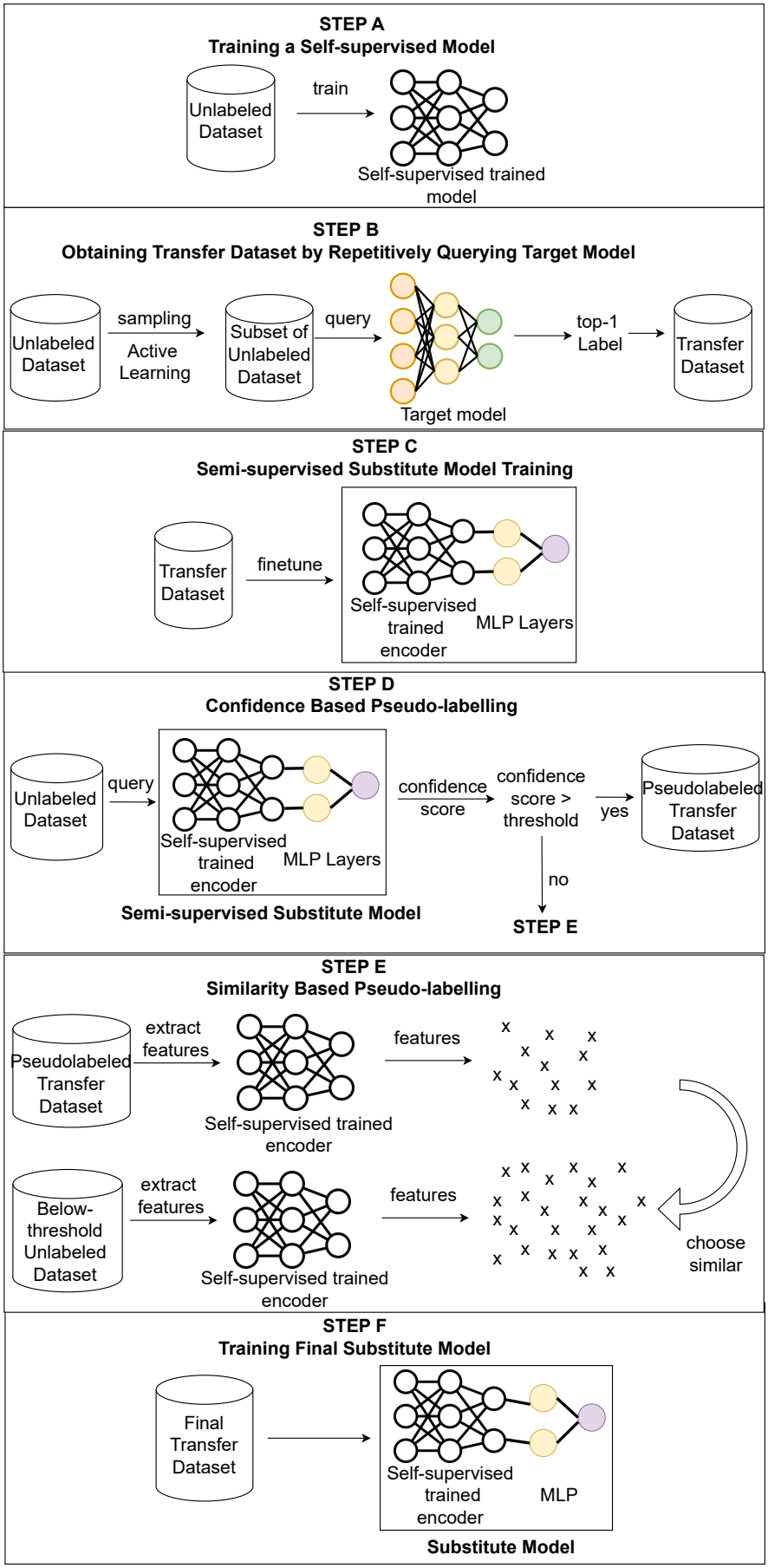


Figure 3.2. Semi-supervised Model Extraction Attack Stages

Therefore, the CIFAR-10 dataset test set was split into two subsets: one for queries and the other as a test set. Although the query dataset sizes vary from 4000, 2000, 1000, and 250 in different experiments, the test set remains fixed at 6000 data points. The results presented in the tables represent the accuracy of the models evaluated on this fixed test set of 6000 samples. For the SVHN dataset, the additional dataset was utilized for query sampling, while the entire test set was used solely for evaluating the model's performance.

A target model with a WideResNet40x10 architecture was trained on the CIFAR-10 train set, achieving an accuracy of 94.4%, while for the SVHN dataset, a target model using the ResNet50 architecture was trained, achieving an accuracy of 94.90%. The selected queries were fed as inputs to the target models, and the top-1 labels produced by the models were collected as outputs to create the transfer datasets.

3.2.1. Baseline Semi-Supervised Model Evaluation

Table 3.1 provides the performance of the semi-supervised baseline substitute model obtained by sequentially implementing Steps A, B, and C as illustrated in Figure 3.2. These results evaluate the models under different query budgets and training configurations. Accordingly, four different query budgets (4000, 2000, 1000, and 250) were examined, and the models were trained with three distinct encoder configurations, which are "all layers frozen," "fine-tuning," and "logits layer trained." Furthermore, the effect on model accuracy was evaluated by comparing two types of classifiers applied on top of the encoder: a Linear Model and a Multi-Layer Perceptron (MLP) Model.

As expected, increasing the query budget leads to higher accuracy, with the best performance achieved at the highest query budget of 4000 across all training methods. This emphasizes the value of having a larger transfer dataset, which provides more diverse and representative data for training the substitute model. However, lower query budgets, such as 250, result in a noticeable drop in accuracy, highlighting the difficulty of constructing effective substitute models with limited labeled data. This demonstrates the trade-off between keeping query costs and detection risks low while ensuring enough data to achieve strong performance.

Table 3.1. Semi-supervised Baseline Substitute Models Test Accuracy Results on CIFAR-10 Dataset for Varying Size Query Budgets

Query Budget	Training Method	Linear Model	MLP Model
4000	All layers frozen	87.23%	87.71%
	Logits layer trained	89.10%	88.85%
	Fine-tuning	**91.01%**	**90.98%**
2000	All layers frozen	86.90%	86.91%
	Logits layer trained	88.25%	87.96%
	Fine-tuning	**89.56%**	**89.71%**
1000	All layers frozen	85.70%	85.25%
	Logits layer trained	86.88%	86.76%
	Fine-tuning	**88.90%**	**88.31%**
250	All layers frozen	82.13%	81.15%
	Logits layer trained	84.33%	82.50%
	Fine-tuning	**84.95%**	**83.65%**

When examining training methods, fine-tuning consistently achieves the best results across all query budgets for both linear and MLP models. This shows that fully updating the model's parameters allows it to adapt better to the transfer dataset, leading to improved accuracy. In contrast, training only the logits layer results in slightly lower accuracy, though it still performs better than freezing all layers, which consistently produces the lowest accuracy. These results underline the limitations of training methods that restrict parameter updates, as they prevent the model from fully utilizing the transfer dataset. Although fine-tuning is more computationally demanding, it provides the most noticeable improvements, particularly when the query budget is sufficient.

When comparing the linear and MLP models, the linear model consistently outperforms the MLP model across most query budgets and training methods, as seen in the table. For example, with a query budget of 4000, the linear model achieves an accuracy of 91. 01% with fine-tuning, slightly higher than the MLP model's 90.98%. A similar trend is observed with lower query budgets, such as 250, where the linear model achieves 84.95% with fine-tuning, compared to 83.65% for the MLP model. This trend can be attributed to several factors. First, the simplicity of the linear model reduces the risk of overfitting, especially when the transfer dataset is limited in size due to query budget constraints. Second, the linear model aligns better with the feature space generated during the self-supervised pretraining phase, effectively leveraging the representations without adding unnecessary

complexity. In contrast, the MLP model, with its additional layers, may overfit the limited data or distort the pre-trained features. Lastly, the linear model requires fewer parameters to train, making it less sensitive to hyperparameter tuning and more stable during training. These factors combined make the linear model a robust and efficient choice, particularly in scenarios with restricted query budgets, while still maintaining competitive accuracy even when compared to the more complex MLP architecture.

Considering the different query budgets presented in the table, the encoder+linear model fine-tuned with the transfer dataset obtained from 4000 queries achieves the highest accuracy. This indicates that a target model with an accuracy of 94.40% can be effectively replicated with a substitute model that achieves 91.01% accuracy. The relative difference between the two models in accuracy is 3.39%.

3.2.2. Performance Analysis of Proposed Method on Problem Domain Query Data

In this section, we analyze the performance of the proposed method in the problem domain by evaluating the impact of different query budgets on substitute model accuracy. The selection of query data differs based on the dataset: for the CIFAR-10 domain, queries are sampled from the test dataset, ensuring that the queried samples are distinct from those in the training set of the target model. In contrast, for the SVHN dataset, the additional dataset is used as the source of queries, also ensuring that the queried samples are distinct from those in the training set of the target model. This deliberate choice guarantees that the extracted substitute model does not rely on data points that were originally used to train the target model, preserving the integrity of the black-box attack scenario.

Analyzing the data presented in Table 3.2, we observe that the fully supervised method struggles significantly at lower query budgets, with a substitute model accuracy of only 48.78% at 250 queries. This substantial drop in accuracy demonstrates the challenge of training a high-quality model when limited labeled data is available. Conversely, the baseline and proposed methods consistently outperform the fully supervised approach. The baseline method achieves 91.01% accuracy at 4000 queries, with an absolute difference of 3.39% compared to the target model. Meanwhile, the proposed method further improves upon this result, reaching 91.93% accuracy at 4000 queries with an absolute difference of only 2.47%.

Table 3.2. Experimental Results on CIFAR-10 Dataset

Method	Query Budget	Substitute Model Test Accuracy	Target Model Test Accuracy	Absolute Diff. Between Target and Substitute Models	Relative Diff. Between Target and Substitute Models
Fully-supervised	4000	83.93%	94.40%	10.47%	88.91%
	2000	73.13%	94.40%	21.27%	77.47%
	1000	68.99%	94.40%	25.41%	73.08%
	250	48.78%	94.40%	45.62%	51.67%
Baseline Method	4000	91.01%	94.40%	**3.39%**	96.04%
	2000	89.56%	94.40%	4.84%	95.03%
	1000	88.90%	94.40%	5.50%	93.55%
	250	84.95%	94.40%	9.45%	86.37%
Proposed Method	4000	91.93%	94.40%	**2.47%**	97.38%
	2000	90.86%	94.40%	3.54%	96.25%
	1000	90.35%	94.40%	4.05%	95.71%
	250	86.00%	94.40%	8.40%	91.10%

Similarly, when the proposed method is applied to a different dataset, such as SVHN, it continues to demonstrate superior performance compared to the fully supervised and baseline methods. As seen in Table 3.3, the proposed method achieves 90.27% accuracy at 4000 queries, outperforming the fully supervised method, which reaches only 84.00%. This trend is consistent across different query budgets, with the proposed method maintaining higher accuracy at 2000 queries (82.08%) and 1000 queries (80.29%), while the fully supervised method lags behind at 78.59% and 70.87%, respectively. This underscores the effectiveness of leveraging self-supervised learning and pseudo-labeling techniques to extract high-quality substitute models even in scenarios where labeled data is scarce.

As the query budget decreases, the proposed method continues to outperform the baseline method. At 1000 queries, it achieves 80.29% accuracy compared to the baseline method's 74.80%, reflecting a significant advantage. This performance gap highlights the benefits of integrating confidence-based and similarity-based pseudo-labeling to enhance dataset diversity and improve model generalization. The relative difference between the target and substitute models remains consistently higher for the proposed method, demonstrating its robustness in handling lower query budgets without a drastic drop in performance.

Table 3.3. Experimental Results on SVHN Dataset

Method	Query Budget	Substitute Model Test Accuracy	Target Model Test Accuracy	Absolute Diff. Between Target and Substitute Models	Relative Diff. Between Target and Substitute Models
Fully-supervised	4000	84.00%	94.90%	10.90%	88.51%
	2000	78.59%	94.90%	16.31%	82.81%
	1000	70.87%	94.90%	24.03%	74.68%
Baseline Method	4000	87.12%	94.90%	7.78%	91.80%
	2000	76.09%	94.90%	18.81%	80.18%
	1000	74.80%	94.90%	20.10%	78.82%
Proposed Method	4000	90.27%	94.90%	4.63%	95.12%
	2000	82.08%	94.90%	12.82%	86.49%
	1000	80.29%	94.90%	14.61%	84.60%

This improvement is attributed to the combination of confidence-based and similarity-based pseudo-labeling, which allows the model to leverage additional unlabeled data effectively. While high-confidence pseudo-labeling ensures reliable labels, similarity-based pseudo-labeling enhances dataset diversity by assigning labels to uncertain samples based on feature similarity. This process enables the substitute model to generalize better and achieve higher accuracy even with a reduced query budget.

The experimental analysis highlights the efficiency of the proposed method in performing model extraction with a constrained query budget. The use of self-supervised learning in conjunction with pseudo-labeling techniques enables the extracted substitute model to closely approximate the target model's performance, even under resource-constrained conditions. The method demonstrates superior performance compared to fully supervised approaches and achieves competitive results against benchmark semi-supervised techniques, particularly in low-query settings. These findings validate the effectiveness of the proposed framework in real-world model extraction attack scenarios.

3.2.3. Benchmark Comparison and Performance Analysis of Proposed Method on Private Dataset Query Data

Table 3.4 offers a comprehensive comparison between the method proposed in this thesis and the approach introduced by Jagieslki*et al.* (Jagielski et al. 2020), which

employs the MixMatch semi-supervised learning method (Berthelot et al. 2019) for model extraction. The comparison assesses the attack efficiencies regarding the absolute and relative differences between target and substitute models, focusing on performance across varying query budgets (4000, 1000, and 250).

To ensure a fair comparison with the benchmark approach, we followed the same setup by selecting query data from the CIFAR-10 training dataset, aligning with the methodology used in the benchmark paper. Unlike our previous experiments, where query samples were chosen from the test dataset to ensure they were distinct from the training data of the target model, in this comparison, the query dataset is drawn directly from the training set of the target model. This setup enables a direct evaluation under identical conditions, allowing a more accurate assessment of performance differences between the two approaches.

Table 3.4. Classification Test Results of Benchmark Semi-Supervised Learning Approaches and Proposed Method Across Different Query Budgets on CIFAR-10 Dataset

Method	**Query Budget**	**Substitute Model Test Accuracy**	**Target Model Test Accuracy**	**Absolute Diff. Between Target and Substitute Models**	**Relative Diff. Between Target and Substitute Models**
(Jagielski et al. 2020)	4000	93.29%	95.75%	**2.46%**	97.43%
	1000	90.63%	95.75%	5.12%	94.65%
	250	87.98%	95.75%	7.77%	91.89%
Proposed Method	4000	92.50%	94.40%	**1.90%**	97.99%
	2000	91.33%	94.40%	3.07%	96.75%
	1000	90.36%	94.40%	4.04%	95.72%
	250	87.68%	94.40%	6.72%	92.88%

The results in Table 3.4 indicate that MixMatch appears to achieve higher accuracy with a query budget of 4000 due to its target model having a higher test accuracy rather than an inherently superior extraction performance. In reality, the proposed method surpasses MixMatch across all query budgets when considering absolute and relative differences. At 4000 queries, the proposed method achieves an absolute difference of 1.90% compared to MixMatch's 2.46%, already demonstrating better efficiency. As the query budget decreases, MixMatch's performance deteriorates more significantly, with an absolute difference of 7.77% at 250 queries, while the proposed method maintains a lower

absolute difference of 6.72%. This consistent trend confirms that the proposed method outperforms MixMatch across all settings, achieving better substitute model accuracy while being more robust to reductions in query budgets.

These findings highlight that while MixMatch benefits from a larger query budget, its reliance on a WideResNet28-2 architecture with 1.5M parameters leads to declining performance under constrained query budgets. Conversely, the proposed method not only remains competitive, but also outperforms MixMatch when fewer queries are available, reinforcing its superiority in resource-constrained scenarios and proving its effectiveness in real-world model extraction attacks.

In conclusion, the proposed method demonstrates superior accuracy, robustness, and efficiency compared to MixMatch. Its ability to maintain high performance across varying query budgets, particularly in limited query budget scenarios, makes it a practical and effective alternative for semi-supervised learning applications.

3.3. Ablation Study

The ablation study section evaluates the performance and impact of various components of our methodology. Specifically, it explores the integration of pseudo-labeling techniques, utilizing both confidence-based and similarity-based approaches. In addition, the influence of the threshold value on the confidence-based method is thoroughly investigated. By systematically removing or modifying individual components, we assess their importance in improving the performance of the substitute model. All ablation experiments are conducted on the CIFAR-10 dataset to ensure a controlled evaluation environment and maintain consistency across comparisons. Additionally, all the results are given for a 4000 query budget. The ablation study also includes a detailed analysis of the time required to execute the model extraction attack, excluding the development phase.

3.3.1. Effect of Pseudo-Labeling Strategies

This section examines the impact of different pseudo-labeling strategies and provides justification for utilizing both confidence-based and similarity-based approaches rather than relying solely on one. By analyzing their individual contributions, we demonstrate that confidence-based pseudo-labeling ensures high-precision labels, while

similarity-based pseudo-labeling enhances dataset diversity, and their combined application leads to a more balanced and effective training set for the substitute model.

3.3.1.1 Pseudo-Label by Confidence

In this experiment, we analyze the effect of pseudo-labeling the entire 50,000 unlabeled samples using the baseline substitute model without applying any confidence filtering. When all samples are labeled in this manner, the total dataset achieves an overall pseudo-label accuracy of 92.88%, meaning that a portion of the dataset still contains incorrect labels. Training a new model on this fully pseudo-labeled dataset results in a final test accuracy of 91.18%, which is lower than the baseline substitute model trained with a more selective pseudo-labeling strategy. This indicates that while having a larger dataset contributes to model generalization, the presence of incorrect labels limits the potential accuracy improvements.

Given this observation, an effective strategy would be to increase the accuracy of the pseudo-labeled dataset by filtering samples based on the baseline substitute model's confidence scores. Rather than labeling all 50,000 samples, applying a threshold-based selection method—where only samples with high-confidence predictions are included—could significantly reduce label noise while maintaining a sufficient dataset size for training. By prioritizing high-confidence outputs, the dataset's accuracy can be improved beyond 92.88%, leading to a cleaner training signal for the substitute model. This suggests that incorporating confidence-based filtering is a more effective approach than blindly labeling all available data, as it enables better trade-offs between dataset size and label quality, ultimately leading to improved substitute model accuracy.

3.3.1.2 Pseudo-Label by Similarity

In similarity measurement, a cosine similarity value of 1 indicates that two feature vectors are perfectly aligned, meaning they are identical in direction. This measure helps evaluate how similar two images are in feature space. Accordingly, we identify the most similar unlabeled images for each labeled image in the transfer dataset by comparing their similarity values. We select the top N most similar images from the pool of unlabeled images that demonstrate a similarity score above the defined threshold to the transfer dataset samples. These selected images are then included in the pseudo-labeled dataset,

ensuring that only the most relevant samples are utilized for labeling.

In this experiment, all 50,000 unlabeled samples were assigned pseudo-labels based on their cosine similarity to samples in the transfer dataset. Feature representations were extracted using the baseline substitute model, and each unlabeled sample was assigned the label of the most similar instance from the transfer dataset. The overall accuracy of this pseudo-labeled dataset was measured at 89%, which is notably lower than the 92.88% accuracy achieved using confidence-based pseudo-labeling. This decline suggests that similarity-based labeling introduces more errors, likely due to the limitations of relying purely on feature-space distance for label assignment. While cosine similarity is effective in grouping visually and semantically related samples, it does not guarantee that the nearest neighbor in the feature space shares the same class, especially in cases where the model's learned representations do not perfectly separate pneumonia and normal cases.

One primary reason for the reduced pseudo-label accuracy is the lack of class separability in the feature space, particularly for borderline cases. If the baseline substitute model has not learned fully discriminative representations, visually similar samples from different classes may still be assigned incorrect labels. Additionally, as cosine similarity only considers distance within the feature space and does not factor in the confidence of the model's prediction, samples with ambiguous or noisy embeddings may receive incorrect labels, further reducing accuracy. Compared to confidence-based pseudo-labeling, which directly leverages the model's prediction probability, similarity-based labeling is more prone to errors when feature representations are not highly distinct between classes. This suggests that while similarity-based pseudo-labeling is valuable for expanding the dataset, applying additional filtering, such as thresholding on similarity scores, could help improve label quality and overall model performance

3.3.2. Impact of Threshold Values on Pseudo-Labeling Strategies

The threshold values used in pseudo-labeling strategies play a crucial role in determining the trade-off between dataset size and label accuracy. Higher thresholds ensure that only high-confidence samples are included, reducing label noise, while lower thresholds increase the number of pseudo-labeled samples but introduce a greater risk of incorrect annotations. In this section, we investigate the impact of threshold values on

pseudo-labeling strategies and their effect on dataset quality and model performance. By adjusting the threshold, we analyze how different selection criteria influence the trade-off between dataset size and label accuracy, ultimately shaping the effectiveness of the proposed method.

3.3.2.1 Impact of Threshold Values on Confidence-Based Pseudo-Labeling

In this section, we analyze the dataset accuracy and, consequently, the accuracy of the substitute model trained on this dataset for different threshold values. Table 3.5 presents the results of confidence-based pseudo-labeling for a query budget of 4000 samples, where the unlabeled dataset is labeled using the baseline self-supervised substitute model. In this approach, only predictions with confidence scores above a specified threshold are assigned as pseudo-labels to the corresponding inputs. The table illustrates how varying confidence thresholds affect the size and accuracy of the pseudo-labeled dataset, as well as the test accuracy of a new substitute model trained on the generated dataset. The experimental results presented in Table 3.4 were obtained using a threshold value of 0.98.

Table 3.5. Accuracy of the Confidence-based Pseudo-labeled Dataset Generated Using the Baseline Self-Supervised Substitute Model and Test Accuracy of the Substitute Model Trained on the Confidence-based Pseudo-labeled Dataset. This table represents the results obtained by following the sequential process outlined in Steps A-B-C-D as illustrated in Figure 3.2.

Confidence Threshold	**Data Size**	**Dataset Accuracy**	**Substitute Model Accuracy**
0.98	44328	98.34%	91.48%
0.95	46612	97.74%	91.21%
0.9	48270	97.01%	91.30%
0.8	50007	96.14%	91.06%
0.7	51217	95.36%	91.18%
0.6	52215	94.57%	91.53%
0	54000	92.82%	91.18%

Table 3.5 presents the relationship between confidence threshold, dataset size, dataset accuracy (true label accuracy), and substitute model accuracy. As the confidence

threshold decreases from 0.98 to 0, the dataset size steadily increases from 44,328 to 54,000. This is expected because lower confidence thresholds allow more samples to be pseudo-labeled, thereby increasing the overall dataset size. However, this comes at the cost of dataset accuracy—i.e., the proportion of correctly pseudo-labeled samples decreases. For instance, at a 0.98 confidence threshold, the dataset accuracy is 98.34%, whereas at 0, it drops to 92.82%. This decline indicates that including more pseudo-labeled samples leads to a higher likelihood of incorrect labels contaminating the dataset.

Despite the decreasing dataset accuracy, the substitute model accuracy remains relatively stable, ranging between 91.06% and 91.53%, with minimal fluctuations. This suggests that as the dataset grows, the model compensates for the lower label accuracy by leveraging the increased diversity of training samples. Notably, even at the lowest threshold where dataset accuracy drops significantly, the substitute model maintains competitive performance. This indicates that the model is robust to some degree of label noise, likely benefiting from the larger training set, which enables it to generalize better despite the presence of incorrect labels.

The stability of the substitute model's performance can be attributed to several factors. First, lowering the confidence threshold introduces noise into the pseudo-labeled dataset, as predictions with lower confidence are more likely to be incorrect. This noise reduces the model's ability to learn precise decision boundaries, limiting its performance. Second, inputs with high confidence scores are often the ones the baseline self-supervised model has effectively memorized. For example, inputs with high confidence scores often come from certain classes that the baseline self-supervised model has memorized. This can cause an uneven distribution in the pseudo-labeled dataset, where some classes are overrepresented and others are underrepresented. As a result, the dataset lacks diversity, making it harder for the substitute model to learn about all classes equally. This imbalance limits the model's ability to improve and helps explain why its performance stays the same. In conclusion, for a 4000 query budget, confidence-based pseudo-labeling effectively leverages the baseline model to generate labeled data, but its performance is constrained by the trade-off between dataset size and label quality.

Figure 3.3 illustrates the relationship between the confidence threshold, dataset size, and dataset accuracy across different query budgets. Consistent with the Table 3.5, Figure 3.3 shows a clear inverse relationship between confidence threshold and dataset

size—lowering the threshold increases the dataset size, as more samples are accepted into the pseudo-labeled dataset. However, this comes at the cost of dataset accuracy, as indicated by the downward trend in accuracy across all query budgets. At higher confidence thresholds (e.g., 0.98, 0.95), the dataset remains smaller but maintains high accuracy. As the threshold decreases (e.g., 0.80, 0.70, 0.60), dataset size increases, but accuracy declines more steeply, aligning with the previous observation that incorporating more samples introduces more incorrectly labeled data.

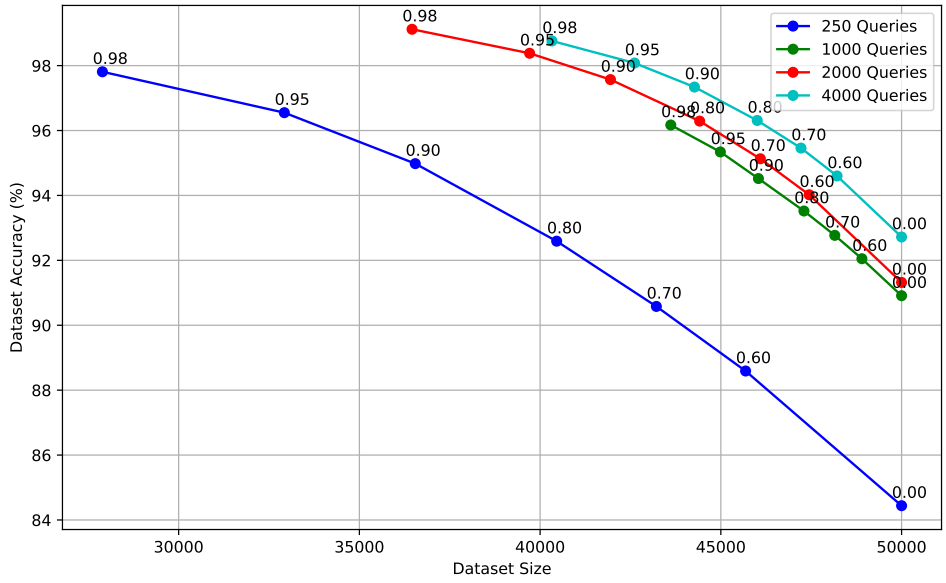


Figure 3.3. Effect of Confidence Threshold on Dataset Size and Dataset Accuracy Across Varying Query Budgets.

3.3.2.2 Impact of Threshold Values on Similarity-Based Pseudo-Labeling

Table 3.6 presents the results of a cosine similarity-based pseudo-labeling approach, where labels are assigned to an unlabeled dataset based on feature similarity. Features for both the labeled and unlabeled datasets are extracted using the semi-supervised baseline substitute model, and cosine similarity is used to find the closest matches between data points. For each unlabeled sample, the label of the most similar sample(s) from the

labeled dataset is assigned. The similarity threshold determines how strict the matching process is, directly impacting the size and quality of the pseudo-labeled dataset.

The table presents the effect of threshold values and the number of sampling iterations on dataset accuracy and dataset size in similarity-based pseudo-labeling. The results indicate that increasing the similarity threshold leads to higher dataset accuracy but significantly reduces the number of pseudo-labeled samples available for training. This trade-off is crucial in determining the effectiveness of pseudo-labeling, as an extremely high threshold ensures only the most confidently labeled samples are included, but at the cost of a drastically reduced dataset size.

For instance, at a threshold of 0.8, the dataset accuracy consistently remains above 99.7%, demonstrating that selecting only highly similar samples minimizes the risk of introducing incorrect labels. However, the dataset size decreases considerably, making it potentially insufficient for training a robust model. In contrast, at a threshold of 0.7, the dataset size is substantially larger, but the dataset accuracy declines slightly, though it is still maintaining a high level. This suggests that while lower thresholds introduce more samples, they also bring in additional label noise. The final row of the table, which includes all pseudo-labeled data without threshold filtering, shows a dataset accuracy of 89.01%, highlighting how including all available samples without filtering leads to a significant drop in accuracy. This result reinforces the necessity of balancing dataset size and accuracy, while higher thresholds improve label quality, they may not provide enough training data, and lower thresholds, while increasing sample size, introduce label noise that can degrade the substitute model's performance.

These findings emphasize that a well-chosen threshold is necessary to achieve an optimal balance between the accuracy and size of the data set. If the dataset size is too small, the model may fail to generalize effectively, whereas a dataset with excessive label noise could limit performance gains. Thus, the pseudo-labeling strategy must consider both factors to ensure effective model training.

The last row of the table represents the scenario where the entire unlabeled dataset is labeled using cosine similarity without applying any threshold. In this case, all 50,000 samples are included in the pseudo-labeled dataset, resulting in a dataset accuracy of 89.01%. This is significantly lower than the accuracies achieved with threshold-based sampling, highlighting the importance of filtering samples based on confidence. When all

samples are labeled regardless of similarity, including noisy and low-confidence matches introduces substantial errors, reducing the overall quality of the dataset. To address this, the approach adopted in this thesis involves initially performing high-accuracy pseudo-labeling using a substitute model. Subsequently, the remaining samples are labeled based on their cosine similarity to the transfer dataset. This combined strategy has yielded the best results.

Table 3.6. Effect of Threshold Values and Number of Sampling on Dataset Accuracy and Sample Selection in Similarity-Based Pseudo-labeling. The table indicates the results for a transfer dataset size of 4000 and an unlabeled dataset size of 50,000.

Number of sampling	**Threshold**	**Dataset Size**	**Dataset Accuracy**
1	0.7	6401	99.25%
	0.8	4744	99.83%
2	0.7	8373	99.10%
	0.8	5169	99.81%
3	0.7	10115	99.00%
	0.8	5501	99.75%
4	0.7	11700	98.91%
	0.8	5779	99.71%
5	0.7	13149	98.91%
	0.8	6016	99.70%
6	0.7	14486	98.91%
	0.8	6234	99.71%
7	0.7	14982	98.56%
	0.8	6430	99.72%
8	0.7	16066	98.59%
	0.8	6612	99.73%
9	0.7	10618	99.34%
	0.8	6774	99.73%
10	0.7	11013	99.35%
	0.8	6928	99.74%
All	0	50000	89.01%

The results highlight a clear trade-off between dataset size and accuracy when using cosine similarity-based pseudo-labeling. Applying a stricter threshold produces smaller, more accurate datasets by filtering out low-confidence matches. Conversely, including the entire unlabeled dataset without a threshold maximizes the dataset size but significantly

compromises accuracy due to the presence of noisy labels.

3.3.3. Time Analysis of the Model Extraction Process

This evaluation provides insights into the computational efficiency of the attack, focusing solely on the time taken for key stages such as querying the target model, constructing the transfer dataset, performing pseudo-labeling, and training the final substitute model. By breaking down each stage, the study quantifies the feasibility of executing the attack under real-world constraints, demonstrating how an adversary with sufficient computational resources and an unlabeled dataset can extract a high-performing substitute model within a practical timeframe. The reported time measurements highlight the efficiency of the proposed approach, emphasizing its applicability in low-query settings while maintaining minimal computational overhead. Table 3.7 presents a detailed breakdown of the time required for each step of the attack. The measurements were conducted on a system equipped with an Intel Core i7 13th Generation CPU, 16GB RAM, and an NVIDIA RTX 4080 GPU, ensuring consistent performance evaluation across experiments.

Table 3.7. Execution Time Breakdown for Each Stage of the Model Extraction Attack

Query Budget	**Transfer Dataset, Creation (sec)**	**Substitute Baseline Training (min)**	**Pseudolabeling (min)**	**Final Substitute Model Training (min)**	**Total Time (min)**
4k	39.88	3.98	1.13	5.13	10.63
2k	26.02	2.41	1.17	4.63	8.47
1k	22.08	1.91	1.03	4.85	8.00
250	21.76	2.91	1.24	4.17	8.53

Table 3.7 shows that higher query budgets generally lead to faster execution, but the 250-query budget scenario has the longest pseudo-labeling time (1.24 min) despite fewer initial labeled samples. This occurs because more samples require similarity-based pseudo-labeling, increasing computational time. The results highlight how reliance on pseudo-labeling grows in low-query settings, though the overall attack remains efficient.

CHAPTER 4

USE CASE ON CHEST X-RAY DATA

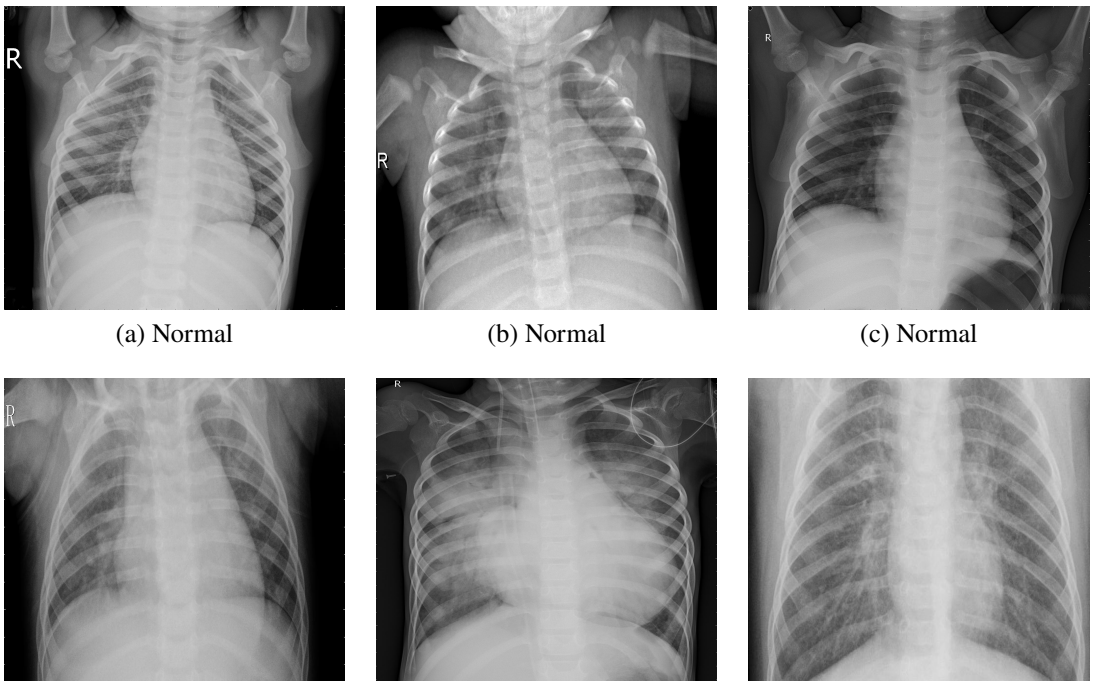
In many domains, acquiring labeled data is both expensive and challenging, as training a model typically requires a large number of annotated samples. One such domain is medical imaging, where expert-annotated datasets are difficult to obtain due to the need for specialized medical knowledge and extensive manual review. This section demonstrates that an adversary with access to a sufficiently large pool of unlabeled data can perform a low-budget model extraction attack and obtain a high-accuracy substitute model. By leveraging self-supervised learning and pseudo-labeling, the proposed method effectively minimizes the reliance on labeled queries while maintaining strong model performance, highlighting a critical vulnerability in black-box AI systems within data-scarce environments.

To evaluate the feasibility of this approach, a pre-trained DenseNet121 model (Densenet121-res224-nih) from the TorchXrayVision framework (Cohen et al., 2022) was used as the target model. This model, trained on the NIH Chest X-ray dataset (Wang et al., 2017), provides classification outputs for 18 different diseases, making it a suitable benchmark for medical imaging applications. The primary focus of this study was on pneumonia classification, where the target model produces probability scores for the presence of pneumonia. Any input receiving an output score of 0.5 or higher was classified as pneumonia-positive, aligning with standard clinical interpretation thresholds.

The query dataset was derived from Kaggle's Chest X-Ray Images (Pneumonia) dataset (Kermany, 2018), an open-access dataset containing 5617 chest X-ray images, categorized into pneumonia and normal cases. These images were obtained from retrospective cohorts of pediatric patients aged one to five years at the Guangzhou Women and Children's Medical Center, and their accuracy was verified by medical experts. A total of 600 images, equally distributed between pneumonia and normal cases, were randomly sampled as the query dataset. The remaining 4616 images were treated as unlabeled data, forming the foundation for the self-supervised learning and pseudo-labeling processes, while an additional 624 samples were held out as a test set for final evaluation. To illustrate the dataset used in this study, Figure 4.1 presents representative normal and

pneumonia cases from Kaggle's Chest X-Ray dataset.

To construct the transfer dataset, the 600 selected samples from Kaggle's Chest X-Ray Images (Pneumonia) dataset were used as queries to the DenseNet121 target model. Each sample, consisting of 300 pneumonia and 300 normal cases, was sent to the target model, which returned its classification outputs. These outputs, paired with their corresponding input images, formed the transfer dataset, serving as the initial labeled data for training the substitute model. This dataset plays a crucial role in the model extraction process, as it provides a foundation for fine-tuning the self-supervised representations learned from the unlabeled dataset. Figure 4.2 illustrates the complete workflow of this process, detailing how the transfer dataset is derived from the query interactions and subsequently utilized in substitute model training.



(a) Normal (b) Normal (c) Normal

(d) Pneumonia (e) Pneumonia (f) Pneumonia

Figure 4.1. Sample Images from Kaggle's Chest X-Ray Images (Pneumonia) dataset (Kermany 2018)

To develop a baseline substitute model, an ImageNet-pretrained SimCLR model was fine-tuned using the transfer dataset, which was constructed from the 600 queried

samples and their corresponding target model outputs. Since SimCLR is a self-supervised representation learning framework, it extracts meaningful features from unlabeled data, but to adapt it for the specific task of pneumonia classification, only the final layer was retrained using the transfer dataset. This ensures that the learned representations remain general while aligning the output space with the target model's decision boundaries. The resulting baseline substitute model achieved a test accuracy of 85.73%, demonstrating its ability to approximate the target model's behavior using a minimal amount of labeled data.

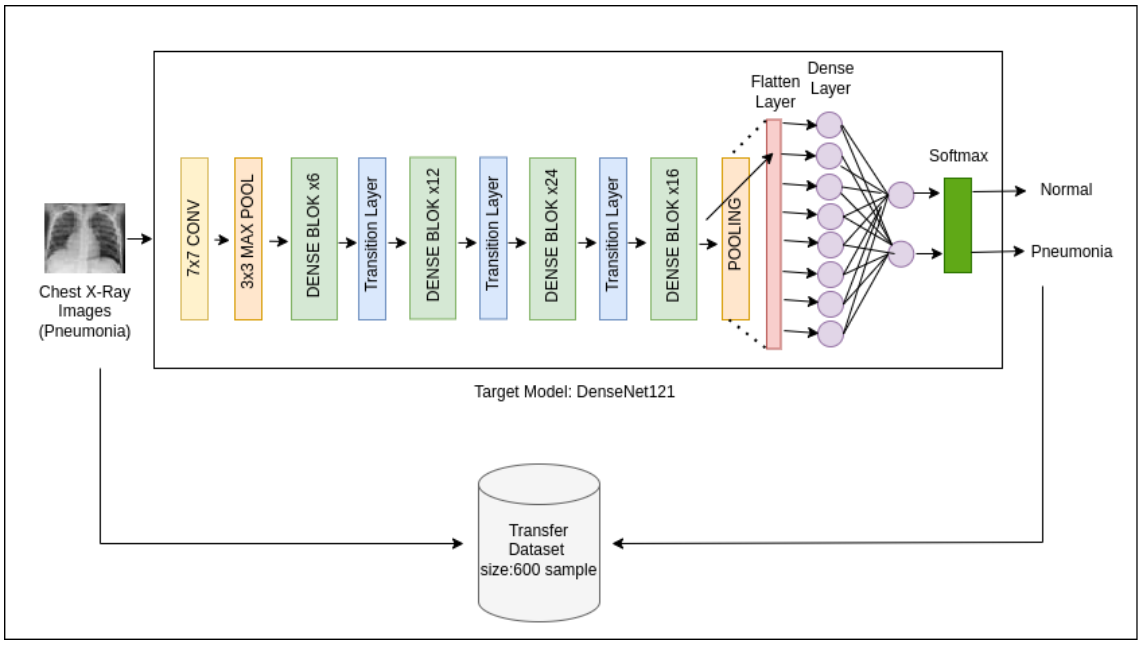


Figure 4.2. Process of Transfer Dataset Creation from Query Interactions

In order to enhance the model performance, pseudo-labeling algorithms were applied to the 4616 unlabeled samples, leveraging the predictions of the baseline substitute model. Instead of manually labeling this large dataset, pseudo-labeling assigned class labels to each sample based on confidence thresholds and similarity-based techniques, effectively expanding the labeled training data. The substitute model was then retrained using this extended dataset, incorporating both the initially labeled transfer dataset and the newly pseudo-labeled samples. This progressive refinement process enabled the model to generalize better, ultimately resulting in an improved test accuracy of 88.78%. Figure

4.3 provides a visual representation of this process, illustrating how the combination of self-supervised learning, transfer dataset training, and pseudo-labeling contributes to the development of a high-performing substitute model.

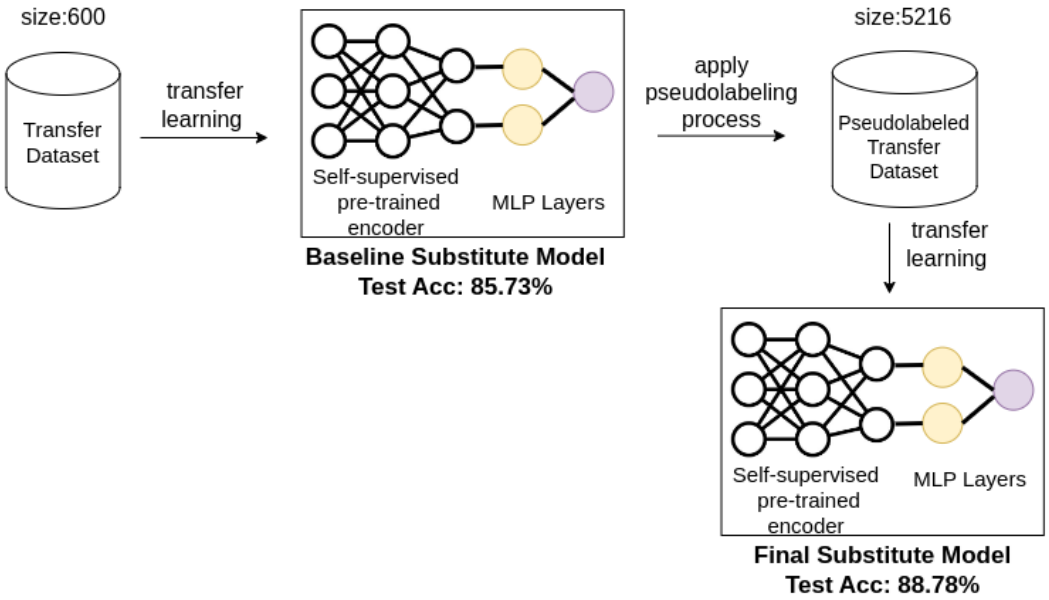


Figure 4.3. Workflow for Constructing the Final Substitute Model through Pseudo-Labeling and Training

These findings underscore the effectiveness of the proposed method in medical imaging applications. Despite the challenges associated with class imbalances and limited labeled data, the methodology demonstrates strong generalization capabilities. The integration of self-supervised learning and pseudo-labeling enhances model performance while reducing reliance on extensive labeled datasets, emphasizing the practical applicability of the approach for real-world deployment.

CHAPTER 5

THREAT MODEL

A threat model provides a structured framework for identifying vulnerabilities within a system, understanding how various threats might exploit these weaknesses, and determining effective strategies to mitigate or defend against potential attacks. By systematically analyzing the system's components, threat modeling helps assess the impact of threats and guides the development of robust security measures. It also outlines possible impacts if the model were to be replicated or reverse-engineered by an attacker. This structured approach allows us to evaluate defenses and create strategies to protect the model's functionality and proprietary value. Without a clear threat model, it can be difficult to identify the model's potential vulnerabilities and anticipate how it might be attacked, which leaves it more exposed to risk.

In the following subsections, we analyze the assets of the model extraction attack, categorize potential adversaries, identify system components that may be targeted, and examine the attack surfaces of the system. Additionally, we discuss prominent model extraction threats and their impact on the system and the model owner and propose effective countermeasures to mitigate these risks.

5.1. Assets

The main assets in this threat model are the target model and the private dataset used to train it. The target model, hosted on an API, is the result of a significant investment in time, expertise, and computational resources. It has commercial value, as companies earn revenue by providing access to their models through APIs, often charging subscription fees or per-query costs. The private dataset is also crucial, as it contains proprietary or sensitive information and is often difficult to collect or replicate, especially in specialized fields like medical imaging or finance. These assets are vital to the model owner's business and intellectual property.

5.2. Adversary

We classify adversaries into two categories: insider and outsider adversaries. Insider adversaries include the model developers or API workers, while outsider adversaries are the users of the API. Insider adversaries can access detailed information about the model, enabling them to carry out white-box attacks. On the other hand, outsider adversaries are generally weaker, which limits them to potentially launching black-box attacks. In the following sections, we provide a detailed analysis of the adversary's goals, motivations, and capabilities in executing this attack.

5.2.1. Adversary's Motivation

An adversary may have two primary objectives: obtaining a local copy of the model or using that copy as a basis for further attacks. To bypass black-box API restrictions, like per-query fees or daily query limits, replicating the model hosted on the API and training a local version can be an effective approach. Another motivation could be to transform black-box models into white-box ones, allowing for more invasive attacks, such as evasion or poisoning. Here, the obtained local model serves as a preliminary step to enable these white-box attacks. For example, at the simplest level, an adversarial sample can be generated from the white-box model to mislead the target model.

5.2.2. Adversary's Capability

Adversaries are categorized by their capabilities as either weak or strong. A strong adversary has access to details such as the model's architecture or private training dataset. Model extraction attacks are typically classified into three types: white-box, grey-box, and black-box. In a white-box attack, the adversary has complete knowledge of the model's training process, including its hyperparameters. In a grey-box attack, the adversary has partial information; for example, they may know the model's architecture but lack access to the private dataset. Finally, in a black-box model extraction attack, the adversary only has access to the model's outputs, which can vary depending on the API—some APIs provide confidence scores, while others only return the top-1 label.

The black-box model extraction attack is the strongest scenario for this type of attack and is particularly suited to APIs. In this scenario, the adversary and the API interaction are only based on queries. Hence, it is called a ”query-based attack.” The limited interaction relies on analyzing the model's output to reconstruct or approximate the target model's functionality without direct access to its internal information.

5.2.3. Adversary's Goal

An adversary can have two aims while conducting a model extraction attack: fidelity extraction or accuracy extraction, as outlined in the study (Jagielski et al. 2020). Fidelity extraction focuses on replicating the target model's behavior as closely as possible. This type of attack does not require ground-truth labeled samples, as it evaluates the substitute model's performance based solely on alignment with the target model's outputs. The substitute model's upper accuracy bound is constrained by the target model's accuracy in fidelity extraction. In contrast, accuracy extraction aims to achieve outputs that align with ground-truth labels, allowing the substitute model's accuracy to potentially exceed that of the target model.

The success of the attack is often measured by the number of queries, known as the query budget, since it directly affects the attack cost, as the adversary typically pays per query. Additionally, APIs may enforce daily query limits or flag high query volumes as suspicious, labeling them as adversarial. Thus, reducing the query budget is a key focus for adversaries looking to extract the model efficiently.

5.3. Trust Model

In the context of a model extraction attack, the system consists of several key actors with distinct roles and trust levels. These actors include the user, the AI developer (model developer), and the API service provider. The AI developer is responsible for designing, training, and deploying the machine learning model, while the API service provider hosts the model and manages its accessibility through the API. Users access the system via the API and can be categorized as either benign users who use the API as intended or adversaries who attempt to exploit the system.

This thesis considers a black-box attack scenario. In this context, the AI developer,

API service provider, and benign users are classified as trusted actors. This is because, in a black-box scenario, neither the API service provider nor the AI developer leaks any internal information about the system. However, the adversary is an untrusted actor, as their goal is to exploit the API through a model extraction attack.

5.4. Attack Surface and Attack Vectors

In model extraction attacks, the primary attack vector focuses on the queries sent to the target model, which are tools that attackers use to explore the model's decision-making process and understand how it operates. Moreover, the query source is critical in determining the attack's success, as it controls the information extracted from the model.

Queries are generally categorized into four types: original data, problem domain data, non-problem domain data, and artificially generated data. Original data refers to the private dataset, and adversaries may gain access to a portion of this dataset to use it for querying. This allows the attacker to exploit the model's learned patterns effectively, as the queries are well-aligned with the problem space. Alternatively, adversaries may use datasets from the problem domain, which can reveal significant details about the model's behavior. For example, in a medical image classification model, adversaries might query the model using medical image data, where the feature space remains the same, and the marginal probability distribution is quite similar to that of the model's training data but not identical. This alignment helps attackers gain insights into the model's decision boundaries and functionality.

In some cases, an adversary may use non-problem domain data that shares the same feature space but has a different marginal probability distribution compared to the private dataset. In this type of query source, the modality of the private dataset and query are the same, such as using image data for image classification models or text data for text classification. While less efficient, such data can still provide valuable insights into the model's decision-making process by revealing how it responds to unfamiliar data distributions. Lastly, adversaries might employ artificially generated data, leveraging techniques such as generative models to create inputs specifically designed to explore the model's vulnerabilities or less commonly visited regions of its decision space. This approach allows attackers to probe the model's behavior in a controlled manner, identifying

weaknesses that could be exploited in a more focused attack.

The effectiveness of the attack depends on the quality, diversity, and alignment of these queries with the target model's domain. By carefully selecting or generating query data, adversaries can maximize the information gained from each interaction, making the query source a fundamental parameter in the attack vector. The attack surface of a model extraction attack is primarily defined by the API interface through which the adversary interacts with the target model. This interface provides an entry point for submitting queries and retrieving outputs, such as class labels or probability distributions. Adversaries exploit the openness of this interface to systematically query the model, collecting outputs to infer its internal structure and replicate its functionality.

Through the API interface, attackers can interact with the model without requiring direct access to its parameters or training data. This makes it a critical vulnerability point, particularly for publicly accessible machine learning models. The scope of the attack surface is influenced by factors such as the level of detail in the outputs provided by the API. For instance, returning class probabilities offers more information to an attacker than hard labels. Additionally, unrestricted query access enables attackers to iterate through large datasets or generate synthetic inputs to refine their understanding of the model.

5.5. Threat Impact

Protecting the target model and private dataset is crucial for maintaining the model owner's revenue, intellectual property, and competitive position. If the target model is stolen, attackers can replicate its functionality without paying for API access, leading to a loss of revenue. This also reduces the provider's ability to offer unique services and weakens their competitive advantage. Furthermore, an adversary could profit from the stolen model, creating direct competition for the original provider.

The private dataset is equally valuable due to the effort and cost involved in its creation. Specialized datasets, such as those in medical imaging, require expertise from professionals, making them expensive and time-intensive to develop. Additionally, datasets containing sensitive or personal information, like medical records, must comply with privacy regulations such as HIPAA. If the outputs of the target model reveal patterns derived from this data, it could result in privacy violations, a loss of trust, and potential

legal or financial penalties.

In conclusion, the theft of the target model or private dataset poses serious risks, including financial losses, damage to reputation, and reduced competitiveness. These risks emphasize the need for robust security measures to protect these critical assets.

5.6. Countermeasures

While countermeasures to black-box model extraction attacks are critical in practice, they are beyond the primary scope of this thesis. This study focuses on developing a simple model extraction attack for an adversary with easy access to unlabeled data, leveraging self-supervised learning methods while maintaining a low query budget rather than exploring defensive strategies. However, for completeness, key countermeasures from the literature are summarized below, along with references to studies that address these challenges.

Defenses against model extraction attacks can be grouped into reactive and proactive approaches. Reactive defenses focus on identifying and responding to attacks either during or after they happen. For example, ownership verification methods, like Dataset Inference (Maini, Yaghini, and Papernot 2021), check if a substitute model was trained using the original dataset by measuring the distance of training samples from the decision boundary. However, this method has limitations when the dataset is publicly available, as models trained on similar data might be incorrectly flagged as stolen (Li et al. 2022). Another reactive method is watermarking, which embeds hidden information into the model during training. Techniques like DAWN (Szyller et al. 2021) change the model's output for specific queries, while DynaMarks (Chakraborty et al. 2022) adds random changes to confidence scores, embedding watermarks without reducing usability. Monitoring systems, such as PRADA (Juuti et al. 2019), analyze query patterns and detect suspicious behavior when query distributions deviate from normal ones.

Proactive defenses aim to stop effective model stealing by making stolen models less useful. Output perturbation methods limit the information adversaries can collect, like rounding confidence scores or returning only top-k labels (Tramèr et al. 2016). (Orekondy, Schiele, and Fritz 2019) introduced Maximizing Angular Deviation (MAD), which modifies confidence scores to distort gradients, making extraction harder.

Combining reactive approaches, such as monitoring and watermarking, with proactive methods like perturbation can make model stealing much harder. These defenses come with trade-offs, like reduced accuracy or higher computational costs, but they effectively protect machine learning models.

The proposed model extraction attack introduces novel challenges for existing countermeasures due to its reliance on self-supervised learning and pseudo-labeling techniques. Traditional monitoring-based defenses, such as PRADA, primarily detect attacks based on query distribution deviations, which may be ineffective against this method as it minimizes the number of queries and strategically selects samples that resemble natural data distributions. Similarly, ownership verification techniques struggle to detect model theft when an attacker constructs a substitute model using unlabeled data and self-supervised learning, rather than relying heavily on the target model's labeled outputs. Watermarking techniques like DAWN and DynaMarks, which modify specific outputs, may also be circumvented if the pseudo-labeling process effectively smooths these alterations over multiple training iterations.

A potential countermeasure against this approach could involve adversarial query obfuscation, where the API model returns adversarially perturbed predictions to disrupt feature extraction and similarity-based pseudo-labeling. Another direction could be active response mechanisms, where models introduce controlled inconsistencies in responses to suspected adversarial queries, making it difficult to derive useful representations through self-supervised learning. Given the stealth and efficiency of the proposed attack, countermeasure development must consider both query minimization strategies and self-supervised feature extraction, which are currently underexplored in existing defenses.

CHAPTER 6

CONCLUSION

This thesis presented a new strategy for executing model extraction attacks in black-box machine learning environments, focusing on minimizing the number of queries while ensuring high accuracy in the substitute model. The approach utilizes self-supervised learning through the SimCLR framework to derive meaningful feature representations from large collections of unlabeled data. By combining this foundation with a novel pseudo-labeling process that incorporates both confidence-based and similarity-based techniques, the method effectively expands the transfer dataset with high-quality labels, improving the substitute model's generalization ability.

A key contribution of this research is its ability to surpass the existing benchmark method MixMatch, especially in scenarios with constrained query budgets. Unlike MixMatch, which relies on intricate semi-supervised techniques, the proposed method achieves similar or better accuracy with a more streamlined and accessible implementation. The robustness and scalability of this approach are evident in its performance under low-query conditions, achieving state-of-the-art results while maintaining computational efficiency. This makes the method not only effective but also practical for real-world applications in black-box settings.

The uniqueness of this work lies in its combination of self-supervised learning and pseudo-labeling for model extraction. SimCLR serves as the backbone of the method, reducing the reliance on labeled data and addressing a major limitation in existing techniques. The two-stage pseudo-labeling strategy enhances the dataset quality: high-confidence predictions provide reliable labels, while similarity-based assignments ensure greater diversity and adaptability. The approach's effectiveness has been demonstrated through rigorous experiments, consistently outperforming benchmarks across various query budgets and scenarios.

Furthermore, this thesis demonstrates a use case scenario on medical imaging data, showcasing the feasibility of the proposed method in a low-budget model extraction setting. By applying the approach to chest X-ray classification, we highlight how self-supervised learning and pseudo-labeling can effectively extract a substitute model with

limited labeled data, reinforcing its applicability in real-world domains where acquiring labeled datasets is challenging.

This study also offers a comprehensive threat model that examines the potential risks and impacts of model extraction attacks. The analysis highlights the importance of addressing adversarial threats and developing appropriate safeguards by placing the proposed method within the broader context of machine learning security.

6.1. Future Work

Although this thesis makes substantial progress in improving the efficiency and accuracy of model extraction attacks, it opens avenues for further exploration. Future research could focus on developing defenses against such attacks, including methods such as query monitoring, output obfuscation, and watermarking to deter adversarial behavior. Expanding the approach to cross-domain settings, where the surrogate and target datasets differ significantly, could further test its adaptability and generalization.

In addition, incorporating other self-supervised frameworks, such as BYOL or MoCo, could reveal alternative methods to achieve similar objectives. Exploring the application of this approach to generative models, such as stable diffusion and VAEs, presents another potential research direction.

Another promising research direction is the application of self-supervised learning techniques for model extraction attacks on text-based models. Similar to vision-based self-supervised learning, natural language processing (NLP) models can leverage unlabeled text corpora to learn high-quality feature representations. Frameworks such as SimCSE (Gao, Yao, and Chen 2021) utilize contrastive learning-based pretext tasks tailored for text data, enabling the extraction of robust sentence embeddings without requiring extensive labeled datasets. These embeddings can then be fine-tuned for downstream applications, allowing adversaries to construct substitute models with minimal query interactions. Unlike traditional model extraction attacks that rely on direct queries to an API, self-supervised learning facilitates the training of substitute models using publicly available text data, significantly reducing dependency on target model output. This approach makes detection more challenging, as adversaries can refine extracted representations offline before making strategic queries. As large-scale language models continue to dominate NLP applications,

understanding their vulnerabilities to self-supervised extraction techniques is crucial to developing effective countermeasures. Future research should explore methods to detect and mitigate these risks, ensuring that proprietary models remain protected against emerging self-supervised model extraction strategies.

Ethical considerations also remain an essential topic, as the development of regulatory policies will be critical to safeguarding intellectual property and preventing misuse of such techniques.

In conclusion, this thesis delivers a well-rounded framework for performing model extraction attacks using self-supervised learning and pseudo-labeling. Advancements in the field, both methodologically and practically, lay a solid foundation for future work in adversarial machine learning, while emphasizing the need for ethical practices and robust defenses in AI systems.

BIBLIOGRAPHY

Bastanlar, Yalin, and Semih Orhan. 2022. "Self-supervised contrastive representation learning in computer vision." In *Artificial Intelligence Annual Volume 2022*. IntechOpen.

Berthelot, David, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in neural information processing systems* 32.

Caron, Mathilde, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. "Unsupervised learning of visual features by contrasting cluster assignments." *Advances in neural information processing systems* 33:9912–9924.

Chakraborty, Abhishek, Daniel Xing, Yuntao Liu, and Ankur Srivastava. 2022. "Dynamarks: Defending against deep learning model extraction using dynamic watermarking." *arXiv preprint arXiv:2207.13321*.

Chandrasekaran, Varun, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. 2020. "Exploring connections between active learning and model extraction." In *29th USENIX Security Symposium (USENIX Security 20),* 1309–1326.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. "A simple framework for contrastive learning of visual representations." In *International conference on machine learning,* 1597–1607. PMLR.

Gao, Tianyu, Xingcheng Yao, and Danqi Chen. 2021. "Simcse: Simple contrastive learning of sentence embeddings." *arXiv preprint arXiv:2104.08821*.

Genç, Didem, Mustafa Özuysal, and Emrah Tomur. 2023. "A taxonomic survey of model extraction attacks." In *2023 IEEE International Conference on Cyber Security and Resilience (CSR),* 200–205. IEEE.

Gong, Xueluan, Yanjiao Chen, Wenbin Yang, Guanghao Mei, and Qian Wang. 2021. "InverseNet: Augmenting Model Extraction Attacks with Training Data Inversion." In *IJCAI,* 2439–2447.

Grill, Jean-Bastien, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. "Bootstrap your own latent-a new approach to self-supervised learning." *Advances in neural information processing systems* 33:21271–21284.

He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. "Momentum contrast for unsupervised visual representation learning." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 9729–9738.

Iyer, Rishabh, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. 2021. "Submodular combinatorial information measures with applications in machine learning." In *Algorithmic Learning Theory,* 722–754. PMLR.

Jagielski, Matthew, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. "High accuracy and high fidelity extraction of neural networks." In *29th USENIX security symposium (USENIX Security 20),* 1345–1362.

Juuti, Mika, Sebastian Szyller, Samuel Marchal, and N Asokan. 2019. "PRADA: protecting against DNN model stealing attacks." In *2019 IEEE European Symposium on Security and Privacy (EuroS&P),* 512–527. IEEE.

Kariyappa, Sanjay, Atul Prakash, and Moinuddin K Qureshi. 2021. "Maze: Data-free model stealing attack using zeroth-order gradient estimation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 13814–13823.

Krizhevsky, Alex, Geoffrey Hinton, et al. 2009. "Learning multiple layers of features from tiny images."

Li, Yiming, Linghui Zhu, Xiaojun Jia, Yong Jiang, Shu-Tao Xia, and Xiaochun Cao. 2022. "Defending against model stealing via verifying embedded external features." In *Proceedings of the AAAI conference on artificial intelligence,* 36:1464–1472. 2.

Lowd, Daniel, and Christopher Meek. 2005. "Adversarial learning." In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining,* 641–647.

Maini, Pratyush, Mohammad Yaghini, and Nicolas Papernot. 2021. “Dataset inference: Ownership resolution in machine learning.” *arXiv preprint arXiv:2104.10706.*

Miura, Takayuki, Toshiki Shibahara, and Naoto Yanai. 2024. “Megex: Data-free model extraction attack against gradient-based explainable ai.” In *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems,* 56–66.

Netzer, Yuval, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. 2011. “Reading digits in natural images with unsupervised feature learning.” In *NIPS workshop on deep learning and unsupervised feature learning,* 2011:4. 2. Granada.

Oliynyk, Daryna, Rudolf Mayer, and Andreas Rauber. 2023. “I know what you trained last summer: A survey on stealing machine learning models and defences.” *ACM Computing Surveys* 55 (14s): 1–41.

Orekondy, Tribhuvanesh, Bernt Schiele, and Mario Fritz. 2019. “Knockoff nets: Stealing functionality of black-box models.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 4954–4963.

Pal, Soham, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. 2020. “Activethief: Model extraction using active learning and unannotated public data.” In *Proceedings of the AAAI Conference on Artificial Intelligence,* 34:865–872. 01.

Papernot, Nicolas, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. “Practical black-box attacks against machine learning.” In *Proceedings of the 2017 ACM on Asia conference on computer and communications security,* 506–519.

Reith, Robert Nikolai, Thomas Schneider, and Oleksandr Tkachenko. 2019. “Efficiently stealing your machine learning models.” In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society,* 198–210.

Shi, Yi, Yalin Sagduyu, and Alexander Grushin. 2017. “How to steal a machine learning classifier with deep learning.” In *2017 IEEE International symposium on technologies for homeland security (HST),* 1–5. IEEE.

Szyller, Sebastian, Buse Gul Atli, Samuel Marchal, and N Asokan. 2021. “Dawn: Dynamic adversarial watermarking of neural networks.” In *Proceedings of the 29th ACM International Conference on Multimedia,* 4417–4425.

Tramèr, Florian, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. “Stealing machine learning models via prediction {APIs}.” In *25th USENIX security symposium (USENIX Security 16),* 601–618.

Truong, Jean-Baptiste, Pratyush Maini, Robert J Walls, and Nicolas Papernot. 2021. “Data-free model extraction.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 4771–4780.

VITA

DİDEM GENÇ

Teaching and Professional Experience

- Research and Teaching Assistant, İzmir Institute of Technology, 2016–2024.

Higher Education

- MSc in Computer Engineering, İzmir Institute of Technology, 2019.
- BSc in Electrical and Electronics Engineering, Fatih University, 2013.

Selected Publications

- Genç D, Özuysal M, Tomur E. **A Taxonomic Survey of Model Extraction Attacks**. 2023 IEEE International Conference on Cyber Security and Resilience (CSR). 2023; pp.200-205. https://doi.org/10.1109/CSR57506.2023.10224959.
- Ercan AT, Genç D, Tomur E. **Endüstriyel Nesnelerin İnterneti Uygulamaları için FPGA Destekli ve Bağlam Tabanlı Erişim Kontrol Güvenlik Sistemi**. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi. 2023; 25(75): 551-558. https://doi.org/10.21205/deufmd.2023257503.
- Genç D, Tomur E, Erten YM. **Context-Aware Operation-Based Access Control for Internet of Things Applications**. 2019 International Symposium on Networks, Computers and Communications (ISNCC). 2019; pp.1-6. https://doi.org/10.1109/ISNCC.2019.8909196.