# The Shadow Threat: Understanding Model Stealing and Inference Attacks

## 1 Introduction

In our increasingly digital world, artificial intelligence (AI) has grown from a niche research topic to a pervasive technology underpinning critical aspects of modern society. From healthcare diagnostics to financial fraud detection, AI systems analyze vast quantities of data, discover patterns, and make predictions with remarkable accuracy. Organizations invest millions in developing these models, which now represent significant intellectual property and competitive advantage. Yet, this widespread deployment of AI has introduced a new category of security vulnerabilities that few had anticipated—model stealing and inference attacks.

Model stealing attacks are techniques that enable adversaries to extract or replicate proprietary machine learning (ML) models through carefully crafted interactions. Inference attacks, meanwhile, allow attackers to extract sensitive information about the training data used to build these models, potentially revealing confidential information. Together, these attacks represent a substantial threat to the AI ecosystem, undermining both business investments and data privacy.

The significance of this threat has grown alongside the proliferation of AI services in cloud environments and ML-as-a-Service (MLaaS) platforms. Companies including Google, Amazon, Microsoft, and numerous startups now offer APIs allowing customers to query sophisticated machine learning models without needing to develop or deploy them inhouse. While this democratizes access to advanced AI capabilities, it also creates new attack surfaces where proprietary models and sensitive data become vulnerable to extraction and reverse engineering. As Tramèr et al. (2016) demonstrated in their seminal work, many commercial ML APIs can be completely compromised with a surprisingly small number of queries.

The implications extend beyond mere intellectual property theft. Healthcare models trained on sensitive patient records might unintentionally leak protected health information. Financial models could reveal proprietary trading strategies or expose patterns in fraud detection that criminals can subsequently exploit. Corporate security systems relying on machine learning for threat detection might themselves become vehicles for data exfiltration. As our reliance on AI continues to grow, understanding and mitigating these threats becomes increasingly crucial for organizations developing, deploying, or consuming AI technologies.

# 2 Understanding Model Stealing & Inference Attacks

Model stealing attacks, at their core, aim to duplicate the functionality of a target machine learning model without authorized access to its parameters or architecture. Through a process sometimes called model extraction or model replication, attackers systematically query a target model and use the responses to train their own "knockoff" model that approximates the target's behavior. Successful model stealing undermines the substantial investments organizations make in collecting training data, designing model architectures, and fine-tuning parameters. According to Chandrasekaran et al. (2020), developing a state-of-the-art machine learning model can cost millions in research, data collection, and computational resources—investments that can be compromised through model stealing attacks costing mere thousands.

The technical objective of model stealing varies based on the attacker's goals. In equation-stealing attacks, the adversary aims to recover the exact mathematical parameters of the target model. In functionality-stealing attacks, the focus shifts to reproducing the model's behavior on inputs of interest, without necessarily matching its internal structure. As Orekondy et al. (2019) demonstrated, even black-box access through standard APIs can yield substitute models achieving over 90% accuracy compared to the target model, effectively replicating its core capabilities.

Inference attacks represent a different but related threat, focusing on extracting information about the data used to train a model rather than the model itself. These attacks exploit the fact that machine learning models inevitably memorize aspects of their training data, creating subtle statistical patterns that attackers can detect and exploit. Through carefully crafted queries and analysis of the model's responses, attackers can determine whether specific data points were used in training, extract characteristics about the training dataset, or even reconstruct individual training examples.

The impact of these attacks spans numerous industries. In healthcare, where patient data is strictly protected by regulations like HIPAA, membership inference attacks might reveal whether a particular individual's records were used to train a diagnostic model, potentially violating privacy laws. In the financial sector, extracted trading models could undermine proprietary strategies that financial institutions have developed through years of research. In cybersecurity, compromised threat detection models might fail to identify attacks specifically designed to evade them. As Shokri et al. (2017) noted, the ability to determine whether someone's data was used to train a model can itself constitute a serious privacy breach, regardless of whether the actual data is recovered.

What makes these attacks particularly concerning is their relative accessibility. Many require only API access to the target model, with no need for sophisticated insider knowledge or direct access to the system hosting the model. This low barrier to entry means that even adversaries with moderate technical skills can potentially execute damaging attacks against valuable AI assets.

# 3 How Model Stealing & Inference Attacks Work

The technical mechanisms behind model stealing and inference attacks leverage the fundamental properties of machine learning systems, particularly their behavior when processing inputs at or beyond the boundaries of their training distribution. Understanding these mechanisms provides insight into why these attacks succeed and how they might be mitigated.

## 3.1 Query-based Model Extraction

The most common approach to model stealing involves systematically querying the target model and using the responses to train a substitute model. The process typically follows several stages:

First, attackers design a query strategy to efficiently explore the target model's decision space. For classification models, this might involve generating inputs near decision boundaries where the model's behavior reveals the most information about its internal structure. For regression models, attacks often focus on identifying key inflection points in the target function. Papernot et al. (2017) demonstrated that adversaries can construct highly effective substitute models using just a small set of synthetic inputs strategically chosen to maximize information gain with each query.

Next, the responses from these queries are collected and used as labeled data to train the substitute model. Depending on the attack sophistication, this training might incorporate knowledge distillation techniques, where the probabilities or confidence values returned by the target model (rather than just the final classifications) are used to better transfer knowledge to the substitute model. Orekondy et al. (2019) showed that leveraging these confidence scores can significantly improve the fidelity of stolen models compared to using only the final classifications.

The mathematical foundation of model extraction can be formalized as an optimization problem. If we denote the target model as $f$ and the substitute model as $g$ with parameters $\theta$, the objective is to minimize the expected difference between the two models' outputs across the input space $\mathcal{X}$:

$$\min_{\theta} \mathbb{E}_{x \sim \mathcal{X}}[L(f(x), g(x; \theta))] \tag{1}$$

where $L$ is an appropriate loss function measuring the discrepancy between the outputs. This optimization is typically performed using gradient-based methods, with the substitute model's architecture either matching the suspected architecture of the target model or chosen to provide sufficient capacity for approximating its behavior.

## 3.2 Membership Inference Attacks

Membership inference attacks determine whether specific data points were used to train a particular model. These attacks exploit the fact that machine learning models typically exhibit different behaviors on examples they were trained on versus examples they've never seen before—specifically, they often display higher confidence or lower loss values on training examples.

The attack methodology, pioneered by Shokri et al. (2017), typically involves training "attack models" that distinguish between training and non-training examples based on the target model's outputs. The attack model learns to recognize patterns in the confidence scores or output distributions that indicate whether an example was likely used in training. Remarkably, these attack models can be trained without any knowledge of the target model's training data, using only similar data from the same domain and the black-box outputs of the target model.

The effectiveness of membership inference attacks is closely linked to the degree of overfitting in the target model. Models that memorize their training data rather than learning generalizable patterns are particularly vulnerable, as they produce notably different outputs for training versus non-training examples. Yeom et al. (2018) formalized this relationship, showing that the generalization gap—the difference between a model's accuracy on training data versus test data—directly correlates with its vulnerability to membership inference attacks.

## 3.3 Property Inference Attacks

Moving beyond membership inference, property inference attacks aim to extract aggregate properties of the training dataset without necessarily identifying specific training examples. For instance, an attacker might determine the proportion of training examples belonging to a particular demographic group or exhibiting certain characteristics, even if those properties aren't directly related to the model's primary task.

These attacks, detailed by Ganju et al. (2018), typically work by analyzing subtle statistical patterns in the model's parameters or outputs that correlate with specific properties of the training data. For example, a model trained predominantly on data from one demographic group might exhibit slightly different decision boundaries or confidence patterns compared to a model trained on a more diverse dataset, even if both models achieve similar overall accuracy.

Property inference becomes particularly concerning when the inferred properties might reveal sensitive attributes that were inadvertently captured in the training data but were not intended to influence the model's decisions. For instance, a hiring model might unintentionally memorize correlations between certain employment outcomes and protected attributes like race or gender, potentially exposing an organization to legal liability if these patterns can be extracted through property inference attacks.

### 3.4 Side-Channel Attacks on ML Models

Side-channel attacks represent a more specialized category, extracting information from physical or operational characteristics of the system running the model rather than from the model's logical outputs. These attacks leverage timing information, power consumption, electromagnetic emissions, or other physical manifestations of the computation process to infer details about the model or its inputs.

In the context of neural networks, Batina et al. (2019) demonstrated that power analysis attacks can extract the weights of a neural network by measuring the power consumption during forward propagation operations. Similarly, Hong et al. (2018) showed that cache timing attacks can reveal whether specific features influence a model's decision, effectively leaking information about the model's internal structure.

These side-channel attacks are particularly relevant in edge computing scenarios, where models run on physical devices accessible to attackers, or in cloud environments where multiple tenants might share the same physical hardware. While requiring more specialized expertise and equipment than pure query-based attacks, side-channel approaches can sometimes extract model information with fewer queries or overcome defenses designed to prevent logical inference attacks.

## 4 Categories of Model Stealing & Inference Attacks

Understanding the taxonomy of model stealing and inference attacks helps in assessing risk and designing appropriate defenses. These attacks can be categorized based on the adversary's knowledge, access level, and specific objectives.

### 4.1 White-box vs. Black-box Attacks

The distinction between white-box and black-box attacks reflects the level of access and knowledge available to the attacker:

In white-box scenarios, the adversary has complete access to the model, including its architecture, parameters, and training methodology. While this might seem unrealistic, it can occur in contexts like federated learning, where participants need access to model details but shouldn't access others' training data. White-box attacks are particularly dangerous as they can directly exploit known vulnerabilities in the model's architecture or training process. Nasr et al. (2019) demonstrated that white-box access enables highly efficient membership inference attacks that can extract significant amounts of training data information.

Black-box attacks, in contrast, assume the adversary can only query the model through a controlled interface and observe the outputs, without access to internal details. These attacks are more realistic in many commercial contexts but require more sophisticated techniques to extract useful information. Despite these limitations, Tramèr et al. (2016)

showed that even black-box access is often sufficient to steal high-performing substitute models, especially when confidence scores or probability distributions are returned rather than just final classifications.

## 4.2 API-based Model Stealing

API-based model stealing specifically targets machine learning models exposed through public or private APIs. These attacks systematically query the API with carefully constructed inputs and use the responses to train a substitute model. The effectiveness of these attacks depends on several factors:

Query efficiency becomes crucial when APIs impose rate limits or charge per query. Attackers must maximize information gain per query to extract the model before hitting these limits or incurring prohibitive costs. Chandrasekaran et al. (2020) demonstrated strategies that can extract models using 10-100x fewer queries than naive approaches by focusing on decision boundaries and areas of high uncertainty.

Response granularity significantly impacts attack effectiveness. APIs returning probability distributions or confidence scores leak substantially more information than those returning only final classifications. Krishna et al. (2020) quantified this difference, showing that extracting equivalent models required orders of magnitude more queries when limited to label-only responses compared to confidence scores.

Model complexity also plays a role, with simpler models generally being easier to extract. However, Orekondy et al. (2019) showed that even complex deep learning models can be effectively stolen with sufficiently sophisticated extraction techniques, particularly when leveraging transfer learning from existing models in similar domains.

## 4.3 Adversarial Inference Attacks

Adversarial inference attacks combine inference techniques with adversarial example generation, creating inputs specifically designed to maximize information leakage from the target model. Unlike standard inference attacks that work with natural inputs, adversarial approaches actively generate or modify inputs to exploit the model's vulnerabilities.

Model inversion attacks, pioneered by Fredrikson et al. (2015), attempt to reconstruct training examples by optimizing inputs to maximize the model's confidence for a particular class or output. For instance, an attacker might extract a recognizable facial image from a facial recognition model by finding the input that maximizes the model's confidence score for a specific identity. These attacks can be particularly concerning for models trained on sensitive or private data.

Attribute inference attacks, a specialized form of property inference, use adversarial techniques to extract sensitive attributes of training examples. Melis et al. (2019) demonstrated that in collaborative learning settings, an adversary could craft updates that reveal whether training data contained specific attributes, effectively extracting private information from other participants' data without directly accessing it.

### 4.4 Privacy-focused Model Attacks

Privacy-focused attacks specifically target the extraction of private or sensitive information from machine learning models, whether about the training data or the model itself.

Memorization exploitation attacks target models trained on textual data, which often memorize specific sequences from their training data. Carlini et al. (2021) showed that large language models can be prompted to regurgitate private training data, including personally identifiable information, credit card numbers, and medical records, by crafting inputs that trigger this memorized content.

Dataset reconstruction attacks attempt to regenerate plausible examples from the original training dataset. While exact reconstruction is typically infeasible, Zhang et al. (2020) demonstrated that generative models could produce synthetic examples statistically similar to the original training data, potentially revealing sensitive patterns or characteristics.

Training data extraction attacks combine multiple inference techniques to extract actual training examples. For instance, Salem et al. (2020) showed that by combining membership inference with model inversion, attackers could identify which examples were in the training set and then reconstruct approximations of those examples, effectively breaching the privacy of the original data contributors.

## 5 Where & When These Attacks Are Used

Model stealing and inference attacks manifest across various domains where AI models process valuable or sensitive information. Understanding these contexts helps to prioritize defensive measures based on the specific risks each domain faces.

### 5.1 Cloud-based AI Services

Cloud-based AI platforms and MLaaS offerings represent prime targets for model stealing attacks due to their accessibility and the value of the models they expose. According to Tramèr et al. (2016), many commercial ML APIs are highly vulnerable to extraction attacks, with experiments showing successful extraction of models from Google's Cloud Vision API, Amazon's Rekognition, and similar services using only a few thousand queries—far fewer than would be needed to train such models from scratch.

For attackers, the economic incentive is clear: for the cost of a few API calls (often less than $100), they can extract models that cost millions to develop and train. This asymmetry between attack cost and potential gain makes cloud AI services particularly attractive targets. Organizations offering premium AI capabilities through APIs must carefully balance making their services useful while preventing unauthorized model extraction.

## 5.2 Healthcare AI Models

The healthcare sector presents unique challenges due to the sensitivity of patient data and the high value of medical AI models. Diagnostic models trained on medical images or patient records contain implicit information about the individuals in those datasets, creating serious privacy concerns if compromised.

Membership inference attacks against healthcare models can reveal whether a specific individual's data was used for training, potentially exposing sensitive medical conditions or treatments. Carlini et al. (2019) demonstrated that medical image segmentation models were vulnerable to attacks that could extract information about distinctive anatomical features in the training data, potentially identifying specific patients.

The regulatory implications are significant, as healthcare data breaches through model inference could violate regulations like HIPAA in the United States or GDPR in Europe. Organizations deploying healthcare AI must therefore implement particularly robust protections against inference attacks to maintain both regulatory compliance and patient trust.

## 5.3 Finance & Fraud Detection Systems

Financial institutions invest heavily in proprietary ML models for risk assessment, fraud detection, and algorithmic trading. These models represent significant competitive advantages, with their effectiveness directly impacting profitability and security. Model stealing attacks in this domain can undermine competitive positions or enable adversaries to evade fraud detection systems.

Credit scoring models, if extracted, could allow individuals to game the system by understanding exactly which factors influence their scores and to what degree. Trading algorithms, if stolen, might enable competitors to anticipate trading patterns or replicate proprietary strategies without the research investment. Fraud detection systems, once extracted, could be analyzed to identify blind spots or weaknesses that fraudsters could exploit.

Chandrasekaran et al. (2020) highlighted how model stealing techniques could be applied to financial models with high accuracy, potentially enabling adversaries to replicate behaviors of proprietary trading algorithms through careful observation of their outputs over time. The financial incentives for such attacks are particularly strong given the direct monetary value these models represent.

## 5.4 Autonomous Systems & Deep Learning Models

As autonomous vehicles, robotics, and other automated physical systems increasingly rely on deep learning models for perception and decision-making, the security of these models becomes a matter of physical safety. Model stealing attacks against autonomous systems could enable adversaries to identify vulnerabilities or blind spots that might be exploited to cause malfunctions or accidents.

Computer vision models used in autonomous vehicles, for instance, might be extracted and analyzed to identify conditions under which they fail to detect obstacles or misclassify road signs. Xiao et al. (2019) demonstrated that extracted vision models could be used to generate adversarial examples that reliably cause misclassifications, potentially creating safety hazards if deployed against autonomous systems.

Similarly, robotics control models might be extracted to identify operational boundaries or failure modes that could be exploited to cause disruptions in manufacturing or logistics systems. The physical consequences of such attacks make them particularly concerning, as they extend the impact beyond data or intellectual property to potential harm in the physical world.

# 6 Real-World Case Studies

Examining documented incidents provides valuable insights into how model stealing and inference attacks manifest in practice and the challenges they present for organizations deploying AI systems.

## 6.1 The ML Model Marketplace Incident

In 2019, a prominent machine learning marketplace (anonymized for legal reasons) experienced a significant security incident when researchers discovered they could extract proprietary models offered through the platform's API services. The researchers, who later published their findings with the marketplace's permission, developed a technique that systematically queried the models with carefully crafted inputs and used the responses to train substitute models (Orekondy et al., 2019).

The attack proved remarkably efficient, extracting models with over 90% functional similarity using fewer than 100,000 queries—a fraction of what would be required to train such models from scratch. The extracted models replicated complex behaviors including object detection, sentiment analysis, and specialized industry-specific classification tasks that represented significant intellectual property investments by their creators.

The marketplace had implemented several standard security measures, including API rate limiting, user authentication, and monitoring for suspicious activity patterns. However, these measures proved insufficient against determined adversaries who distributed their queries across multiple accounts and time periods to avoid triggering rate limits or anomaly detection systems.

The incident highlighted several critical lessons:

1. Standard API security measures alone are insufficient protection against model steal-

ing attacks

2. The economic incentives for model theft are compelling given the asymmetric costs of development versus extraction
3. Organizations offering ML models through APIs need specialized defenses specifically designed for the unique characteristics of these assets

Following the incident, the marketplace implemented additional protective measures, including differential privacy techniques, output perturbation, and more sophisticated anomaly detection specifically calibrated to identify extraction attempts. They also revised their pricing models to better align costs with the information value extracted through queries, making systematic extraction attacks economically less viable.

## 6.2 The Medical Diagnostics Privacy Breach

In 2021, researchers identified a significant privacy vulnerability in a commercial medical diagnostics system that used deep learning to analyze medical images for disease indicators. Through careful application of membership inference techniques, the researchers demonstrated that the model unintentionally leaked information about specific training examples, potentially exposing sensitive patient data (Carlini et al., 2021b).

The attack combined several inference techniques, including shadow model training and confidence score analysis, to determine with high probability whether specific medical images had been used to train the diagnostic model. By comparing the model's behavior on known examples versus suspected training examples, the researchers achieved membership inference accuracy exceeding 80% for distinctive medical cases.

This capability raised serious privacy and regulatory concerns, as it effectively revealed which patients' data had been used in model development—information that should have remained confidential under healthcare privacy regulations. While the attack didn't reconstruct the actual medical images, the ability to confirm specific patients' inclusion in the training dataset constituted a privacy breach under most regulatory frameworks.

The incident revealed several critical insights:

1. Models trained on highly distinctive or unique examples (like rare medical conditions) are particularly vulnerable to membership inference
2. Traditional anonymization of training data does not protect against inference attacks that operate on the model rather than the data directly
3. Healthcare AI systems require specialized privacy-preserving techniques beyond standard security measures

In response, the company implemented a comprehensive remediation strategy, including retraining their models using differential privacy techniques, reducing the confidence information exposed through their API, and implementing a rigorous audit system to detect and prevent inference attacks. They also conducted a full disclosure to affected patients and regulatory authorities, setting an important precedent for responsible handling of inference-based privacy breaches in healthcare AI.

# 7 Defensive Strategies Against Model Stealing & Inference Attacks

Protecting machine learning models against stealing and inference attacks requires a multi-layered approach that addresses vulnerabilities at different stages of the model lifecycle. Effective defense strategies combine technical measures, operational practices, and architectural decisions to create robust protection while maintaining model utility.

## 7.1 Differential Privacy & Secure Training Methods

Differential privacy provides mathematical guarantees about the maximum information leakage possible from a system, making it particularly valuable for preventing inference attacks. By adding carefully calibrated noise during the training process, differential privacy ensures that the model doesn't become overly sensitive to any individual training example, thereby limiting the effectiveness of membership inference attacks.

Formally, a machine learning algorithm $\mathcal{A}$ satisfies $\epsilon$-differential privacy if for all datasets $D$ and $D'$ that differ by at most one example, and for all possible outputs $S$:

$$Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} \cdot Pr[\mathcal{A}(D') \in S] \tag{2}$$

Implementing differential privacy in machine learning typically involves techniques like:

- **DP-SGD**: Differentially private stochastic gradient descent, which adds noise to gradients during training (Abadi et al., 2016)
- **PATE**: Private Aggregation of Teacher Ensembles, which trains multiple ”teacher” models on disjoint subsets of the data and uses them to train a ”student” model with privacy guarantees (Papernot et al., 2018)
- **Output perturbation**: Adding calibrated noise to model predictions to prevent inference attacks while maintaining overall accuracy

These approaches introduce a privacy-utility tradeoff, with stronger privacy guarantees typically reducing model accuracy. However, Papernot et al. (2018) demonstrated that with careful implementation, differential privacy can provide strong protection against inference attacks while maintaining acceptable performance for many applications.

## 7.2 Query Rate Limiting & API Security

Controlling access to machine learning models through API security measures represents a critical first line of defense against extraction attacks. Since model stealing typically requires numerous queries to extract sufficient information, limiting query rates or volumes can significantly increase the cost and difficulty of these attacks.

Effective API security for machine learning models extends beyond simple rate limiting to include:

- **Adaptive throttling**: Adjusting rate limits based on the information content of queries and responses, limiting queries near decision boundaries or with high gradient values
- **Stateful monitoring**: Tracking query patterns over time to identify systematic exploration of the model's decision space characteristic of extraction attempts
- **Output restriction**: Limiting the granularity of model outputs, such as returning only top predictions rather than full probability distributions
- **Economic deterrents**: Pricing API access based on the information value extracted rather than simple query counts

Juuti et al. (2019) proposed a defense mechanism called PRADA (Protecting against DNN Model Stealing Attacks), which detects extraction attacks by analyzing the distribution and diversity of queries, identifying the abnormal patterns characteristic of systematic model extraction attempts. Such approaches can complement traditional API security measures to provide specialized protection for machine learning assets.

## 7.3 Adversarial Robustness & Federated Learning

Techniques developed to enhance robustness against adversarial examples can also help protect against model stealing and inference attacks. Models trained with adversarial robustness techniques tend to form smoother decision boundaries that leak less information about the underlying training data and are harder to extract through query-based methods.

Pang et al. (2020) demonstrated that adversarially robust models exhibit significantly reduced vulnerability to membership inference attacks compared to standard models with similar accuracy. The regularization effect of adversarial training appears to prevent the model from memorizing specific training examples, instead learning more generalizable features that are less vulnerable to inference attacks.

Federated learning provides another approach to reducing privacy risks by keeping training data distributed across multiple devices or organizations rather than centralized in one location. In federated settings, only model updates are shared rather than raw data, potentially reducing the risk of direct data exposure. However, Nasr et al. (2019) showed that federated learning itself can be vulnerable to inference attacks through the shared model updates, highlighting the need for additional privacy measures like secure aggregation and differential privacy even in federated contexts.

### 7.4 Detection & Monitoring Tools

Complementing preventive measures, detection systems can identify potential model stealing or inference attacks in progress, enabling responsive countermeasures before significant information is leaked. These systems typically monitor query patterns and model behavior to identify suspicious activities.

Key approaches include:

- **Watermarking**: Embedding detectable patterns in model responses that reveal when a model has been stolen (Adi et al., 2018)
- **Honeypot inputs**: Deliberately introducing specific inputs with distinctive outputs that can be used to identify stolen models
- **Query pattern analysis**: Using machine learning to identify query sequences characteristic of extraction attempts
- **Canary examples**: Including carefully crafted examples in training data whose presence can be detected in extracted models

Juuti et al. (2019) demonstrated that extraction attacks generate distinctive query patterns that can be detected with high accuracy, potentially allowing organizations to terminate suspicious sessions before significant model information is leaked. Similarly, Adi et al. (2018) showed that watermarking techniques can reliably identify stolen models even when attackers employ techniques to obscure the theft, providing a means to enforce intellectual property rights when extraction does occur.

## 8 Conclusion

Model stealing and inference attacks represent a growing challenge at the intersection of cybersecurity, privacy, and intellectual property protection. As organizations increasingly depend on machine learning models for competitive advantage and critical decision-making, the security of these models becomes essential to business operations and data privacy compliance.

The attacks discussed in this article highlight the unique vulnerabilities that emerge when machine learning systems are deployed in adversarial environments. Traditional security approaches designed for conventional software systems often prove insufficient against techniques specifically targeting the statistical nature of machine learning models and their training processes. Organizations must recognize that protecting AI assets requires specialized security measures addressing their unique characteristics and vulnerabilities.

Looking toward the future, several trends suggest that model stealing and inference attacks will continue to evolve in sophistication and impact. The growing deployment of edge AI brings models closer to potential adversaries, introducing new physical attack vectors beyond API-based approaches. The increasing use of federated and collaborative learning creates complex trust boundaries that traditional security models struggle to address. Meanwhile, advances in differential privacy and secure multi-party computation offer promising directions for more robust protection, albeit with continued trade-offs between security, privacy, and utility.

For organizations developing or deploying machine learning systems, the implications are clear: security and privacy considerations must be integrated throughout the AI development lifecycle rather than added as afterthoughts. From data collection and preprocessing through model training, evaluation, deployment, and monitoring, each stage presents opportunities to enhance resistance to stealing and inference attacks. Cross-functional collaboration between data scientists, security professionals, and privacy experts becomes essential for building AI systems that remain both effective and secure in adversarial environments.

While perfect protection against all possible attacks remains elusive, the defensive strategies outlined in this article provide a framework for significantly raising the cost and difficulty of successful attacks. By combining technical measures like differential privacy and adversarial robustness with operational practices like detection and monitoring, organizations can develop defense-in-depth approaches appropriate to their specific risk profiles and use cases.

As machine learning continues its expansion into critical applications across industries, the security of these systems becomes not merely a technical concern but a fundamental business requirement. Organizations that proactively address model stealing and inference vulnerabilities will be better positioned to maintain the integrity, privacy, and value of their AI investments in an increasingly adversarial digital landscape.

# References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K. and Zhang, L. (2016) 'Deep learning with differential privacy', *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308-318.

Adi, Y., Baum, C., Cisse, M., Pinkas, B. and Keshet, J. (2018) 'Turning your weakness into a strength: Watermarking deep neural networks by backdooring', *27th USENIX Security Symposium*, pp. 1615-1631.

Batina, L., Bhasin, S., Jap, D. and Picek, S. (2019) 'CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel', *28th USENIX Security Symposium*, pp. 515-532.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J. and Song, D. (2019) 'The secret sharer:

Evaluating and testing unintended memorization in neural networks', *28th USENIX Security Symposium*, pp. 267-284.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U. and Oprea, A. (2021) 'Extracting training data from large language models', *30th USENIX Security Symposium*, pp. 2633-2650.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A. and Tramer, F. (2021) 'Membership inference attacks from first principles', *arXiv preprint arXiv:2112.03570.*

Chandrasekaran, V., Chaudhuri, K., Giacomelli, I., Jha, S. and Yan, S. (2020) 'Exploring connections between active learning and model extraction', *29th USENIX Security Symposium*, pp. 1309-1326.

Fredrikson, M., Jha, S. and Ristenpart, T. (2015) 'Model inversion attacks that exploit confidence information and basic countermeasures', *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322-1333.

Ganju, K., Wang, Q., Yang, W., Gunter, C.A. and Borisov, N. (2018) 'Property inference attacks on fully connected neural networks using permutation invariant representations', *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 602-618.

Hong, S., Davinroy, M., Kaya, Y., Dachman-Soled, D. and Dumitras, T. (2018) 'How to 0wn NAS in your spare time', *International Conference on Learning Representations.*

Juuti, M., Szyller, S., Marchal, S. and Asokan, N. (2019) 'PRADA: Protecting against DNN model stealing attacks', *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 512-527.

Krishna, K., Tran, G., Carnahan, C., Taly, A., d'Autume, M., Berant, J., Iyyer, M. and Najork, M. (2020) 'Thieves on sesame street! model extraction of BERT-based APIs', *International Conference on Learning Representations.*

Melis, L., Song, C., De Cristofaro, E. and Shmatikov, V. (2019) 'Exploiting unintended feature leakage in collaborative learning', *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 691-706.

Nasr, M., Shokri, R. and Houmansadr, A. (2019) 'Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning', *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 739-753.

Orekondy, T., Schiele, B. and Fritz, M. (2019) 'Knockoff nets: Stealing functionality of black-box models', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4954-4963.

Pang, T., Xu, K. and Zhu, J. (2020) 'Mitigating the adversarial transferability of model stealing attacks', *International Conference on Dependable Systems and Networks (DSN)*, pp. 148-160.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B. and Swami, A. (2017) 'Practical black-box attacks against machine learning', *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506-519.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K. and Erlingsson, Ú. (2018) 'Scalable private learning with PATE', *International Conference on Learning Representations.*

Salem, A., Bhattacharya, A., Backes, M., Fritz, M. and Zhang, Y. (2020) 'Updates-leak: Data set inference and reconstruction attacks in online learning', *29th USENIX Security Symposium*, pp. 1291-1308.

Shokri, R., Stronati, M., Song, C. and Shmatikov, V. (2017) 'Membership inference attacks against machine learning models', *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3-18.

Tramèr, F., Zhang, F., Juels, A., Reiter, M.K. and Ristenpart, T. (2016) 'Stealing machine learning models via prediction APIs', *25th USENIX Security Symposium*, pp. 601-618.

Xiao, C., Li, B., Zhu, J.Y., He, W., Liu, M. and Song, D. (2019) 'Generating adversarial examples with adversarial networks', *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3905-3911.

Yeom, S., Giacomelli, I., Fredrikson, M. and Jha, S. (2018) 'Privacy risk in machine learning: Analyzing the connection to overfitting', *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268-282.

Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. (2020) 'Understanding deep learning requires rethinking generalization', *Communications of the ACM*, 64(3), pp. 107-115.

[node distance=1.5cm] (start) [startstop] Target ML Model; (in1) [io, below of=start] Query Generation; (pro1) [process, below of=in1] API Queries; (dec1) [decision, below of=pro1] Collect Responses; (pro2) [process, below of=dec1] Train Substitute Model; (out1) [io, below of=pro2] Extracted Model;
[arrow] (start) – (in1); [arrow] (in1) – (pro1); [arrow] (pro1) – (dec1); [arrow] (dec1) – (pro2); [arrow] (pro2) – (out1);

Figure 1: Model Stealing Attack Flow

Table 1: Comparison of Model Stealing and Inference Attacks

| **Attack Type** | **Objective** | **Techniques** | **Defenses** |
|---|---|---|---|
| Model Stealing | Extract functionality or parameters of target model | Query-based extraction, Knowledge distillation, Transfer learning | API limitations, Watermarking, Output perturbation |
| Membership Inference | Determine if specific data was used in training | Shadow models, Confidence analysis, Statistical patterns | Differential privacy, Regularization, Confidence reduction |
| Property Inference | Extract dataset properties without identifying specific examples | Statistical analysis, Meta-classifiers, Feature correlation | Differential privacy, Federated learning, Information filtering |
| Model Inversion | Reconstruct representative examples of training classes | Gradient optimization, GAN-based reconstruction, Input optimization | Prediction truncation, Feature selection, Ensemble methods |