

FlowGuard: Flow Matching for Identity-Independent Detection of Data-Free Model Stealing Attacks on Energy System Intrusion Detection Systems

Maxime Schwarzer
Karlsruhe Institute of Technology,
Thales Deutschland GmbH
Karlsruhe, Germany
maxime.schwarzer@as-systeme.de

Laurin Holz
CortAIx Labs, Thales Deutschland
Ditzingen, Germany
laurin.holz@thalesgroup.com

Tobias Huerten
CortAIx Labs, Thales Deutschland
Ditzingen, Germany
tobias.huerten@thalesgroup.com

Johannes Loevenich
CortAIx Labs, Thales Deutschland
Ditzingen, Germany
johannes.loevenich@thalesgroup.com

Thies Moehlenhof
CortAIx Labs, Thales Deutschland
Ditzingen, Germany
thies.moehlenhof@thalesgroup.com

Roberto Rigolin
CortAIx Labs, Thales Deutschland
Ditzingen, Germany
roberto.rigolin@thalesgroup.com

Veit Hagenmeyer
Karlsruhe Institute of Technology
Karlsruhe, Germany
veit.hagenmeyer@kit.edu

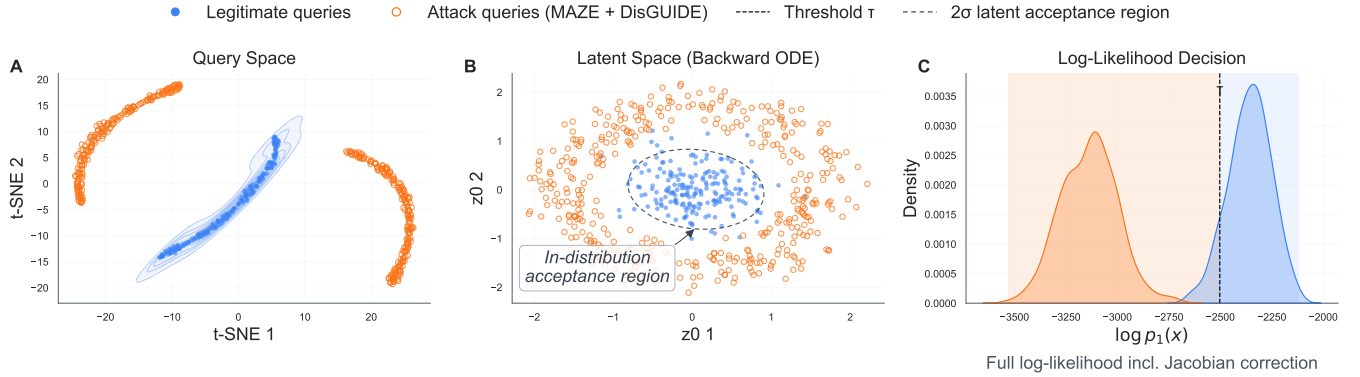


Figure 1: Flow Matching OOD detection. (A) t-SNE of legitimate and synthetic attack queries in input space. (B) Latent representations z_0 after backward ODE integration; attack queries fall outside the 2σ acceptance region. (C) Log-likelihood distributions (Eq. 1) with decision threshold τ . Each query is evaluated individually, independent of client identity.

Abstract

Artificial Intelligence (AI)-based Intrusion Detection Systems (IDS) deployed in energy infrastructure are vulnerable to model theft attacks, which allow adversaries to create evasive traffic offline. Current defences against model extraction rely either on identity-bound query monitoring, which is ineffective against distributed attackers (Sybil), or on prediction poisoning through soft-label perturbation, which is inapplicable to hard-label IDS deployments. Therefore, we propose FlowGuard, an identity-independent defence

based on flow matching that classifies incoming queries as out-of-distribution (OOD) prior to IDS processing. This approach exploits the fact that queries generated synthetically for data-free model stealing attacks occupy a lower-dimensional manifold than real network traffic. This results in measurably lower log-likelihoods when using a Continuous Normalizing Flow that has been trained on legitimate data. We evaluate our method against PRADA and FDI Net using MAZE and DisGUIDE attacks in single-client and distributed (100-client Sybil) settings. While PRADA’s detection rate dropped to 0% when the distribution changed, our defence maintained a stable detection rate across both settings without relying on identity information. We discuss the scope and limitations of the approach, and outline potential applications to data-dependent attacks.



This work is licensed under a Creative Commons Attribution 4.0 International License.
ACM Sustainability Week Companion '26, Banff, AB, Canada
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2199-1/26/06
<https://doi.org/10.1145/3765611.3815137>

CCS Concepts

• **Security and privacy** → **Software security engineering**; • **Computing methodologies** → *Artificial intelligence*; *Neural networks*.

Keywords

Model Extraction Attack, Intrusion Detection System, Flow Matching, Out-of-Distribution Detection, Sybil Attack, Critical Infrastructure Security

ACM Reference Format:

Maxime Schwarzer, Laurin Holz, Tobias Huerten, Johannes Loevenich, Thies Moehlenhof, Roberto Rigolin, and Veit Hagenmeyer. 2026. FlowGuard: Flow Matching for Identity-Independent Detection of Data-Free Model Stealing Attacks on Energy System Intrusion Detection Systems. In *ACM Sustainability Week 2026 (ACM Sustainability Week Companion '26), June 22–25, 2026, Banff, AB, Canada*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3765611.3815137>

1 Introduction

AI-based IDS are increasingly deployed to protect critical energy infrastructure, including smart grids and SCADA networks [7]. These systems use DNNs to classify network traffic in real time. When such models are exposed through query interfaces, for instance, within a Security Operations Center or via internal APIs in a SDD architecture, they become targets for MEAs [14, 19].

In a MEA, an adversary systematically queries the target model to train a functionally equivalent substitute. This substitute then serves as an offline testbed for crafting adversarial evasion traffic: network packets misclassified as benign by the original IDS [16]. The compound threat of extraction followed by evasion is particularly severe in energy systems. Unlike traditional IT environments, where breaches primarily result in data loss, an undetected intrusion in a smart grid or SCADA network can lead to direct physical consequences, including equipment damage, cascading failures, and widespread power outages.

Existing defenses against MEAs fall into two categories, query detection and poisoning prediction, both with limitations in this setting. Query detection methods such as PRADA [4] and FDINet [22] analyze incoming query patterns to identify anomalous behavior. However, these approaches operate on a per-identity basis, collecting statistics over queries from individual clients. Distributing queries across multiple identities via round-robin scheduling (a Sybil attack) reduces PRADA’s detection to 0%. Even global aggregation variants can be defeated through traffic mixing.

Prediction poisoning methods such as Adaptive Misinformation [6] and MODELGUARD [18] perturb the model’s output probabilities to degrade the substitute model. These defenses require access to the full probability vector (soft labels). In practice, many IDS deployments return only a binary hard label (attack/benign), rendering prediction poisoning inapplicable. Furthermore, even when applicable, model extraction remains feasible despite such perturbations [1].

While generative models and latent space representations have already been successfully applied to detect anomalies and model complex dynamics in energy systems [3, 20], we now adapt this

principle to secure these IDS deployments against MEAs. Specifically, we propose using a CNF trained via Flow Matching [10, 11] on the distribution of legitimate network traffic to classify incoming queries as in-distribution or OOD before they reach the IDS. FlowGuard operates on the query content rather than on query metadata or identity information, making it inherently resilient to Sybil attacks.

The key empirical observation is that data-free model stealing attacks (e.g., MAZE [5], DisGUIDE [17]) generate synthetic queries that are useful for extracting decision-boundary information without necessarily matching the full training distribution. Under a density model trained on genuine traffic, these synthetic queries receive substantially lower log-likelihoods than legitimate queries. FlowGuard therefore treats low likelihood as the OOD signal.

The motivation for applying Flow Matching comes from recent work on FlowPure [2], which demonstrates that CNFs trained with Conditional Flow Matching can effectively distinguish adversarial examples from clean inputs by measuring the magnitude of the learned velocity field. We adapt this principle to the distributed data-free model stealing detection problem.

The main contributions of this paper are as follows:

- (1) We propose FlowGuard, an identity-independent defense against data-free model stealing attacks based on Flow Matching OOD detection.
- (2) We evaluate the defense against MAZE and DisGUIDE attacks on CIFAR-10 [8], a standard benchmark dataset for model stealing attacks, and compare against PRADA and FDINet.
- (3) We discuss the scope of the approach, its current limitation to data-free attacks, and outline future extensions.

This paper is organized as follows. Section 2 provides the technical background on model stealing attacks, existing defenses, and the mechanics of normalizing flows. The proposed FlowGuard methodology is detailed in Section 3, explaining how Flow Matching is used to compute log-likelihoods for individual queries to detect synthetic attack data. Section 3.1 defines a threat model involving a black-box attacker utilizing data-free extraction methods. Section 4 discusses the experimental evaluation, benchmarking FlowGuard against PRADA and FDINet in both single-client and distributed Sybil attack settings. Finally, Section 5 summarizes the findings and suggests future work.

2 Background & Related Work

2.1 Model Stealing Attacks (MEAs)

A MEA targets a victim model $f_V : \mathcal{X} \rightarrow \mathcal{Y}$ accessible through a query interface. The adversary trains a substitute model f_S by querying f_V with inputs $x \in \mathcal{X}$ and observing responses $\hat{y} \in \mathcal{Y}$. The goal is typically fidelity extraction, where f_S replicates the decision boundary of f_V [19]. Data-free attacks generate these queries synthetically: MAZE [5] co-trains a generator with the substitute model, and DisGUIDE [17] extends this with an ensemble-based disagreement and diversity loss. Both produce queries from noise without access to the original training data.

2.2 Defenses Against Model Stealing

Query detection. PRADA [4] applies the Shapiro-Wilk test to the distribution of pairwise distances among queries from a single client, flagging deviations from normality. FDINet [22], the current state-of-the-art for detecting distributed attacks, computes a Feature Distortion Index from internal model activations to detect colluding clients. Both methods require accumulating a sufficient number of queries from individual identities before detection is possible.

Prediction poisoning. Methods such as Reverse Sigmoid [9], Prediction Poisoning/MAD [15], and MODELGUARD [18] modify the returned probability vector to mislead the substitute model's training. These require soft-label access and do not prevent extraction under hard-label settings [1].

2.3 Normalizing Flows and OOD Detection

At their core, Normalizing Flows [11] are bijective transformations that continuously map a complex, unknown data distribution (such as network traffic) into a simple, tractable base distribution (typically a standard Gaussian). This invertibility allows for the exact likelihood computation of new samples. Building on this, Flow Matching [10] trains CNFs by regressing a neural network onto the conditional velocity field that transports samples between a source and target distribution. Unlike traditional normalizing flows, Flow Matching does not require expensive Jacobian computations during training, while still enabling exact density evaluation via the instantaneous change of variables formula at inference:

$$\log p_1(x) = \log p_0(z_0) - \int_0^1 \text{tr} \left(\frac{\partial f_\theta}{\partial z}(z_t, t) \right) dt \quad (1)$$

where p_1 denotes the data distribution, and a sample $x \sim p_1$ is interpreted as z_1 and mapped to a latent representation z_0 by integrating the learned ODE $\frac{dz}{dt} = f_\theta(z, t)$ backward from $t = 1$ to $t = 0$. The base distribution p_0 is typically chosen as a standard Gaussian.

The ability to compute exact log-likelihoods makes Flow Matching a candidate for OOD detection, but likelihood direction must be calibrated for the concrete data and model. Deep generative models can assign higher likelihood to OOD data than training data in certain cases (e.g., SVHN vs. CIFAR-10) [13]. In our setting, the calibrated signal is the lower tail of the likelihood distribution: legitimate validation queries define the accepted range, and synthetic extraction queries are flagged when their likelihood falls below that range.

3 FlowGuard: Flow Matching for OOD Detection

Our defense inserts a density-based filter between the query interface and the IDS. The filter operates as follows:

- (1) A Continuous Normalizing Flow f_θ is trained via Flow Matching on the same data distribution used to train the IDS. The training follows the Conditional Flow Matching objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, x_0, x_1} [\|f_\theta(x_t, t) - u_t(x_t | x_0, x_1)\|^2], \quad (2)$$

where $x_0 \sim p_0$ (base Gaussian), $x_1 \sim p_{\text{data}}$, and u_t is the target conditional vector field along straight paths from x_0 to x_1 .

- (2) When a query x_q arrives at the IDS interface, we treat it as a data-space sample at $t = 1$ and integrate the learned

ODE $\frac{dz}{dt} = f_\theta(z, t)$ backward to $t = 0$ to obtain its latent representation z_0 . The model then computes $\log p_1(x_q)$ via Equation (1) as the base log-density $\log p_0(z_0)$ (with p_0 chosen as a standard Gaussian) corrected by the accumulated divergence term along the trajectory.

- (3) If $\log p_1(x_q) < \tau$, the query is classified as OOD and blocked. The threshold τ is a lower-tail threshold calibrated on a held-out validation set of legitimate queries.

3.1 Threat Model

We consider an adversary targeting an AI-based IDS deployed to protect energy infrastructure.

Attacker capabilities. The attacker has black-box query access to the IDS and receives only hard labels (attack/benign). Given sufficient resources to coordinate queries across multiple identities (Sybil attack), the attacker uses data-free model stealing methods that generate queries from noise or co-trained generators without access to the original training data distribution.

Attacker objective. Extract a substitute model with high fidelity which enables crafting adversarial evasion traffic which bypasses the original IDS decisions.

Defender capabilities. The defender has access to the legitimate training data distribution (or a representative sample) to train a density model. The defender can inspect each incoming query before it reaches the IDS. The defender does not rely on any identity or session information.

3.2 Why this works for data-free attacks.

Data-free model stealing methods generate queries through processes that do not have access to the full complexity of the real data distribution. MAZE [5] uses a generator co-trained with the substitute model, producing samples that approximate useful regions of the decision boundary rather than the full data manifold. DisGUIDE [17] adds diversity through ensemble disagreement but still generates from noise. These objectives can produce queries that are informative to the victim classifier while remaining unlikely under a density model trained on legitimate data.

Unlike classifier confidence, the CNF score measures compatibility with the legitimate data distribution. Data-free attacks optimize queries to extract target-model behavior, but they do not explicitly optimize likelihood under the defender's density model. As a result, their synthetic samples can be useful for extraction while still lying in low-density regions of the training distribution.

3.3 Identity independence.

The defense evaluates each query individually based on its content. No identity information, session tracking, or query history is required. This makes the defense inherently resilient to Sybil attacks: even if an attacker distributes queries across thousands of identities, each individual query is still evaluated against the density model.

4 Evaluation

We evaluate whether content-based density filtering remains effective under distributed, identity-hiding model extraction, where identity-based detectors fail.

4.1 Experimental Design

Our evaluation follows the threat model from Section 3: a hard-label black-box adversary performing data-free extraction. We benchmark three query defenses under identical query budgets:

- PRADA [4]: Per-identity query distribution analysis (Shapiro-Wilk test on pairwise distances).
- FDI_{Net} [22]: Feature Distortion Index computed from internal model activations.
- FlowGuard (ours): Log-likelihood threshold on the trained CNF.

We test against two data-free attacks:

- MAZE [5]: Generator-based data-free attack with co-training.
- DisGUIDE [17]: Ensemble-based generator with disagreement/diversity loss.

We evaluate two attacker regimes: (1) *Single-client*, where all attack queries are issued under one identity, and (2) *Distributed* (Sybil setting), where the same total budget is spread across 100 client identities. This split tests whether a defense depends on identity-level temporal statistics (PRADA, FDI_{Net}) or on per-query content (FlowGuard).

4.2 Protocol

We use CIFAR-10 with a VGG16-BN victim model (85.03% test accuracy) as the initial benchmark, following the standard evaluation protocol in model stealing research [4, 18]. For each defense attack pair, we run one malicious and one benign episode: (1) 200 attack queries generated by MAZE or DisGUIDE, and (2) 200 clean reference queries sampled from legitimate data. Metrics are computed at query level over the combined set.

Flow model. We train a CNF using Meta’s `flow_matching` library for 1000 epochs on CIFAR-10. The velocity field is parameterized by a U-Net backbone and optimized with the Conditional Flow Matching objective (MSE between predicted and target velocity) using a Conditional Optimal Transport probability path from Gaussian noise to data. The threshold τ is selected empirically as a lower-tail cutoff from log-likelihood score distributions on validation data.

We report Detection Rate (equivalent to TPR), FPR, Precision, F1, Macro-F1, and ROC-AUC. This includes both threshold-dependent metrics (TPR, FPR, F1) and threshold-independent ranking quality (ROC-AUC).

4.3 Results

Table 1 summarizes the full aggregate detection experiment, while Fig. 2 illustrates the corrected lower-tail score separation on a representative 10-query MAZE deep dive.

Corrected likelihood direction. The MAZE deep dive confirms that FlowGuard blocks low-likelihood queries. In the 10-query example shown in Fig. 2, benign CIFAR-10 queries receive scores between 8478.75 and 16885.64, while MAZE queries receive scores between -1504.09 and -1349.31. With $\tau = 5000$, all benign queries remain above the threshold and all MAZE queries fall below it. This corrected direction is the basis for the aggregate metrics in Table 1.

Single-client setting. Both PRADA and FlowGuard detect data-free attacks, but with different trade-offs. For MAZE, PRADA reaches TPR 0.840, FPR 0.000, and F1 0.913, while FlowGuard achieves higher recall (TPR 0.965) at moderate FPR (0.170), yielding F1 0.904. For DisGUIDE, FlowGuard reaches TPR 1.000 and ROC-AUC 1.000 with F1 0.919, whereas PRADA reaches TPR 0.833 and F1 0.909. FDI_{Net} shows comparable detection performance for DisGUIDE (F1 0.784) but fails to detect MAZE queries (ROC-AUC 0.488).

Distributed setting (100 clients). PRADA collapses to no detection at all (TPR 0.000, F1 0.000, ROC-AUC 0.500) for both attacks. This is expected: distributing queries across identities prevents sufficient per-client accumulation for the statistical test.

FlowGuard remains stable across both settings. For MAZE it keeps TPR 0.965 and F1 0.904 (ROC-AUC 0.922); for DisGUIDE it keeps TPR 1.000 and F1 0.919 (ROC-AUC 1.000). The per-query density evaluation is unaffected by how queries are distributed across identities.

FDI_{Net} shows mixed results. It detects DisGUIDE queries with high sensitivity (TPR 1.000), but at high false-positive rate (FPR 0.530), reducing precision. For MAZE under distributed querying, its TPR drops to 0.500 with F1 0.493, indicating lower robustness than FlowGuard in this setup.

4.4 Interpretation

The results confirm that identity-independent, content-based detection is crucial for securing models against distributed extraction. PRADA, which relies on per-client statistical assumptions, breaks down under Sybil coordination. FDI_{Net}, while designed to handle distributed attacks by analyzing internal feature distortions, faces a different limitation. Although its performance does not degrade under Sybil coordination, it struggles with an excessively high false-positive rate (FPR 0.530) and fails to reliably detect queries generated by MAZE (ROC-AUC dropping to 0.473). This indicates a lack of robustness across different data-free generators. FlowGuard demonstrates that density-based filtering overcomes both challenges. Because it evaluates each query’s log-likelihood independently and blocks lower-tail outliers, it is inherently immune to Sybil partitioning (unlike PRADA). Furthermore, by anchoring the decision boundary to the legitimate data distribution rather than relying on internal model activations, it maintains high detection rates across different attack generators.

The difference between ROC-AUC and fixed-threshold metrics is worth noting. High ROC-AUC indicates good ranking of benign vs. malicious queries across all thresholds, while FPR and F1 depend on the chosen operating point τ . A defense with near-perfect ROC-AUC can still exhibit non-zero false positives at a specific deployment threshold.

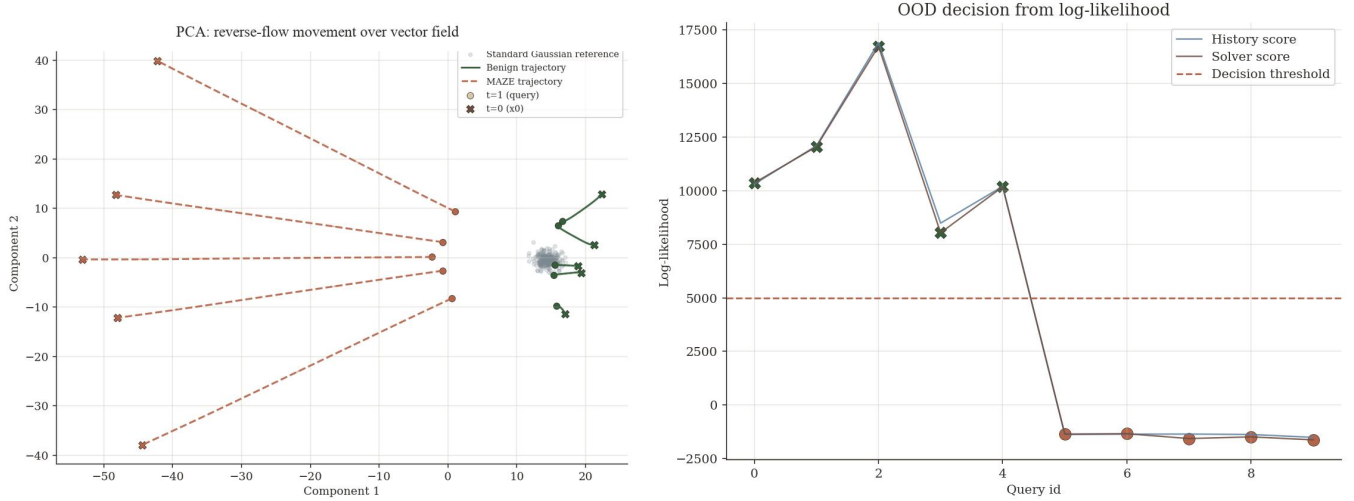
4.5 Limitations

The evaluation is limited to one dataset (CIFAR-10), one victim architecture, and one run per condition (no confidence intervals). The attack set covers data-free extraction only (MAZE, DisGUIDE). These results should be interpreted as evidence for the effectiveness against distributed data-free model stealing, not as a universal guarantee across all extraction families or data modalities. In particular, an adaptive attacker that explicitly optimizes against the defender’s

Table 1: Detection performance of query defenses against model-extraction attacks on CIFAR-10 (VGG16-BN target). Values are query-level metrics. Higher is better except false-positive rate (FPR), where lower is better.

Setting	Attack	Defense	Detection Rate	TPR	FPR↓	Precision	F1	Macro-F1	ROC-AUC
Single-client	MAZE	FDINet	0.545	0.545	0.530	0.507	0.525	0.507	0.488
	MAZE	PRADA	0.840	0.840	0.000	1.000	0.913	0.919	0.920
	MAZE	FlowGuard	0.965	0.965	0.170	0.850	0.904	0.897	0.921
	DisGUIDE	FDINet	1.000	1.000	0.530	0.644	0.784	0.712	0.988
	DisGUIDE	PRADA	0.833	0.833	0.000	1.000	0.909	0.918	0.917
	DisGUIDE	FlowGuard	1.000	1.000	0.170	0.850	0.919	0.913	1.000
Distributed (100 clients)	MAZE	FDINet	0.500	0.500	0.530	0.485	0.493	0.485	0.473
	MAZE	PRADA	0.000	0.000	0.000	0.000	0.000	0.333	0.500
	MAZE	FlowGuard	0.965	0.965	0.170	0.850	0.904	0.897	0.922
	DisGUIDE	FDINet	1.000	1.000	0.530	0.644	0.784	0.712	0.989
	DisGUIDE	PRADA	0.000	0.000	0.000	0.000	0.000	0.333	0.500
	DisGUIDE	FlowGuard	1.000	1.000	0.170	0.850	0.919	0.913	1.000

Note. Bold values highlight best-performing entries for key decision metrics within each attack block.

**Figure 2: Representative MAZE deep dive for ten queries: five benign CIFAR-10 queries and five MAZE attack queries. The left panel shows the reverse-flow trajectory in a PCA projection of the learned flow dynamics, while the right panel shows the corresponding OOD decisions from log-likelihood. Benign queries remain above the calibrated lower-tail threshold, whereas MAZE queries fall below it and are blocked.**

density model could reduce the observed likelihood gap. Extending to IDS-specific datasets (e.g., ERENO IEC 61850), repeated seeds, and adaptive attackers is necessary future work.

5 Conclusion & Future Work

We present FlowGuard, a defense against data-free model stealing attacks based on Flow Matching OOD detection. The approach classifies incoming queries by computing their log-likelihood under a CNF trained on the legitimate data distribution. In our experiments, synthetic queries from data-free attacks receive consistently lower likelihoods than legitimate queries and can be reliably detected with a calibrated lower-tail threshold.

Our evaluation on CIFAR-10 shows that while PRADA’s detection drops to 0% under distributed querying (100 clients), FlowGuard maintains stable detection (TPR 0.965–1.000) regardless of how queries are distributed. FDINet shows mixed behavior with high false-positive rates in some configurations. The per-query nature of the density filter makes it inherently resilient to Sybil attacks.

The current scope is limited to data-free attacks. For future work, we plan to:

- Extend the evaluation to realistic IDS datasets (ERENO IEC 61850, CIC-IDS) and additional defenses (D-ADD [12], MeCo [21]).
- Investigate the combination of FlowGuard with techniques used in adversarial purification. FlowPure [2] demonstrates

that CNFs can also detect adversarial examples through velocity field magnitude at $t = 0$. Combining density-based detection of extraction queries with velocity-based detection of adversarial evasion traffic could provide a unified defense against the full model-stealing-then-evasion attack chain.

- Investigate whether more sophisticated generators (e.g., diffusion-based) can produce queries that overcome the density filter and develop corresponding countermeasures.

References

- [1] Varun Chandrasekaran, 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.
- [2] Elias Collaert, Abel Rodriguez, Sander Joos, Lieven Desmet, and Vera Rimmer. 2025. FlowPure: Continuous Normalizing Flows for Adversarial Purification. arXiv:2505.13280 [cs.LG] <https://arxiv.org/abs/2505.13280>
- [3] Benedikt Heidrich, Matthias Hertel, Oliver Neumann, Veit Hagenmeyer, and Ralf Mikut. 2024. Using conditional Invertible Neural Networks to perform mid-term peak load forecasting. *IET Smart Grid* 7, 4 (2024), 460–472. arXiv:<https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/stg2.12169> doi:10.1049/stg2.12169
- [4] Mika Juuti, 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)*. IEEE Computer Society, Los Alamitos, CA, USA, 512–527. doi:10.1109/EuroSP.2019.00044
- [5] Sanjay Kariyappa, 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 (CVPR)*. 13814–13823.
- [6] Sanjay Kariyappa and Moinuddin K Qureshi. 2020. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 770–778.
- [7] Noora Zidan Khalaf, Israa Ibraheem Al Barazanchi, Israa Ibraheem Al Barazanchi, A. D. Radhi, Sushma Parihar, Pritesh Shah, and Ravi Sekhar. 2025. Development of real-time threat detection systems with AI-driven cybersecurity in critical infrastructure. *Mesopotamian Journal of CyberSecurity* 5, 2 (Jun. 2025), 501–513. doi:10.58496/MJCS/2025/031
- [8] Alex Krizhevsky. 2009. *Learning multiple layers of features from tiny images*. Technical Report. University of Toronto.
- [9] Taesung Lee, Benjamin Edwards, Ian Molloy, and Dong Su. 2019. Defending against neural network model stealing attacks using deceptive perturbations. In *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 43–49.
- [10] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. Flow Matching for Generative Modeling.
- [11] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. 2024. Flow Matching Guide and Code. arXiv:2412.06264 [cs.LG] <https://arxiv.org/abs/2412.06264>
- [12] Jian-Ping Mei, Weibin Zhang, Jie Chen, Xuyun Zhang, and Tiantian Zhu. 2025. Defense against model stealing based on account-aware distribution discrepancy. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'25/IAAI'25/EAAI'25)*. AAAI Press, Article 68, 8 pages. doi:10.1609/aaai.v39i1.32041
- [13] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2019. Do deep generative models know what they don't know? (2019).
- [14] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proc. IEEE CVPR*. 4954–4963.
- [15] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2020. Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks. In *ICLR*.
- [16] Nicolas Papernot, 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.
- [17] Jonathan Rosenthal, Eric Enouen, Hung Viet Pham, and Lin Tan. 2023. DisGUIDE: Disagreement-Guided Data-Free Model Extraction. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 8 (Jun. 2023), 9614–9622. doi:10.1609/aaai.v37i8.26150
- [18] Minxue Tang, Anna Dai, Louis DiValentin, Aolin Ding, Amin Hass, Neil Zhenqiang Gong, Yiran Chen, and Hai "Helen" Li. 2024. ModelGuard: Information-Theoretic Defense Against Model Extraction Attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, 5305–5322. <https://www.usenix.org/conference/usenixsecurity24/presentation/tang>
- [19] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction APIs. In *Proc. USENIX Security*. 601–618.
- [20] Marian Turowski, Benedikt Heidrich, Kaleb Phipps, Kai Schmieder, Oliver Neumann, Ralf Mikut, and Veit Hagenmeyer. 2022. Enhancing anomaly detection methods for energy time series using latent space data representations. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems (Virtual Event) (e-Energy '22)*. Association for Computing Machinery, New York, NY, USA, 208–227. doi:10.1145/3538637.3538851
- [21] Zhenyi Wang, Li Shen, Tongliang Liu, Tiehang Duan, Yanjun Zhu, Donglin Zhan, David Doermann, and Mingchen Gao. 2023. Defending against data-free model extraction by distributionally robust defensive training. *Advances in Neural Information Processing Systems* 36 (2023), 624–637.
- [22] Hongwei Yao, Zheng Li, Haiqin Weng, Feng Xue, Zhan Qin, and Kui Ren. 2025. FDI-Net: Protecting Against DNN Model Extraction Using Feature Distortion Index. *IEEE Transactions on Dependable and Secure Computing* 22, 04 (July 2025), 3179–3191. doi:10.1109/TDSC.2024.3520599