ARTEMIS: Adaptive Reweighing for Transferable Evasion via Meta-learning in Zero-Query Network Intrusion Detection Systems

Lei Wang[1], Qingsong Zou[1,2], Qing Li[2], Jianping Zhang[3], Yong Jiang[1,2]
[1] Tsinghua Shenzhen International Graduate School, Shenzhen, China [2] Peng Cheng Laboratory, Shenzhen, China
[3] The Chinese University of Hong Kong, Hong Kong, China

2025 IEEE 33rd International Conference on Network Protocols (ICNP) | 979-8-3315-0376-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICNP65844.2025.11192405

***Abstract*—Machine Learning-based Network Intrusion Detection Systems (ML-NIDSes) are vital for cyber-security, yet their inherent vulnerability to adversarial attacks poses a persistent challenge. Among these, zero-query transfer-based attacks present a particularly realistic and formidable threat, as adversaries operate with no knowledge of or interaction with the target NIDS. However, the efficacy of such attacks is critically hampered by poor adversarial transferability across diverse NIDS model architectures and by strict protocol-defined constraints on network traffic modifications. To probe the robustness of NIDSes under zero-query attacks, we introduce ARTEMIS, a novel hybrid attack framework engineered to dramatically enhance adversarial transferability for zero-query attacks. ARTEMIS leverages the generalization capabilities of meta-learning to adapt to unknown target models by simulating diverse black-box transfer tasks. Simultaneously, it combines a reinforcement learning-inspired adaptive reweighing mechanism to maximize transfer potential by promoting the effective use of heterogeneous substitute ensembles. Extensive evaluations on the BCCC-CIC-IDS2017/2018 datasets across closed-set, open-set, and cross-set zero-query scenarios confirm that ARTEMIS significantly outperforms state-of-the-art baselines. Our work presents a powerful methodology for NIDS vulnerability assessment and provides crucial insights for developing defenses against transfer-based evasions.**

***Index Terms*—adversarial attack, zero-query, black-box, NIDS**

I. Introduction

In recent years, the application of Machine Learning-based Network Intrusion Detection Systems (ML-NIDSes) to detect malicious traffic has become a significant trend [1]–[3]. Compared with classical signature-based techniques, ML models yield richer feature representations and improved detection of zero-day attacks [4]–[6]. However, despite the superior detection accuracy and automation capabilities of ML-based NIDSes, their robustness are usually overlooked, exhibiting significant vulnerability, particularly when facing adversarial attacks [7]–[9]. Once a malicious flow is misclassified as benign, adversaries can easily bypass defense, leading to severe consequences. Therefore, investigating adversarial attack mechanisms targeting ML-NIDSes, and providing insights for developing defenses against such attacks is of paramount importance.

Currently, most research on adversarial attacks against NIDSes assumes a white-box setting, where the adversary has access to the model architecture and even the training data [4], [10]. However, this is often unrealistic in practice. Due to security and privacy concerns, deployed NIDSes typically operate as closed systems, only outputting final prediction labels (normal or abnormal) without disclosing model structures or returning any confidence scores or probability information, representing a typical decision-based hard-label black-box scenario [11]–[13]. Research on black-box attacks primarily employs two strategies: 1) **Query-based attacks** [14]–[16], which use model outputs (scores/labels) from crafted queries to estimate gradients or decision boundaries; and 2) **Transferability-based attacks** [17]–[19], which generate adversarial examples using local substitute models and exploit their transferability to unknown target models. While effective in domains like image processing, query-based methods face challenges when generating adversarial traffic targeting NIDSes. NIDSes often return only hard labels and are sensitive to abnormal query patterns or excessive query rates, making query-based attacks easily detectable and likely to trigger defenses [20], [21]. Therefore, studying how the stealthier, non-interactive transferability-based adversarial methods would affect the defensive performance of NIDSes become a more realistic proposition.

However, effectively applying such transferability-based black-box attack strategies to NIDSes presents numerous unique challenges. **First**, NID models exhibit high diversity, ranging from traditional machine learning models (e.g., decision tree ensembles [22]) to complex deep neural networks [1], [23]–[25]. These models vary significantly in structure, decision boundaries, and sensitivity to inputs [7], [26], making it exceptionally difficult to find universally effective adversarial perturbations. Moreover, most prior transfer-based attacks [27], [28] rely on gradient information to probe transferability between neural networks, and therefore break down when confronted with non-differentiable tree models. **Second**, unlike the image domain [29], [30], modifications to network traffic are subject to strict constraints compared to pixel values [31], [32]. Traffic features exhibit complex interdependencies and

Corresponding author: Yong Jiang (jiangy@sz.tsinghua.edu.cn)

must adhere to the specifications of underlying network protocols (e.g., TCP/IP) to be transmitted and processed correctly. Furthermore, many feature values have fixed numerical ranges or can only take discrete values. Any non-compliant modification can render the sample invalid or fail to achieve the attack objective. **Lastly**, as previously mentioned, black-box NIDS typically provide only hard-label outputs and are highly sensitive to query behavior. Therefore, any queries to the target NIDS is impractical, making attack methods difficult to obtain valuable information for generating adversarial traffic.

To address these challenges, we propose ARTEMIS, a novel hybrid adversarial attack framework under the zero-query setting. ARTEMIS systematically tackles these critical issues through its unique design: **First**, to counter the zero-query unknown target model problem, it employs a meta-learning paradigm. This involves constructing a diverse pool of substitute models, comprising both differentiable and non-differentiable architectures to reflect NIDS heterogeneity, and simulating varied black-box transfer scenarios to learn generalizable evasion strategies. **Second**, ARTEMIS addresses the challenge of substitute ensemble heterogeneity by incorporating an RL-inspired adaptive reweighing mechanism. This mechanism dynamically optimizes the influence of each model, guiding the adversarial example generation towards shared adversarial subspaces. The integration of diverse model types is further achieved by a unified momentum accumulation approach, which consolidates directional information from both differentiable models (e.g., attacked via MI-FGSM with Bayesian weight sampling [33]) and non-differentiable tree models (e.g., attacked via Leaf Tuple Attack [34] with input Gaussian sampling). **Third**, to ensure the generated adversarial traffic adheres to strict network data modification constraints, ARTEMIS incorporates a dedicated Protocol-Aware projection constraint module that enforces protocol specifications and feature validity.

Through this integrated approach, ARTEMIS iteratively refines perturbations. By learning to exploit common vulnerabilities across numerous simulated transfer tasks, it significantly enhances the ability of adversarial examples to successfully transfer to unknown NIDSes in zero-query settings.

The main contributions of this paper can be summarized as follows:

1) **Hybrid Zero-Query Framework.** We introduce a meta-learning–reinforcement-learning framework that crafts adversarial traffic *without* querying the target NIDSes. The outer meta-learner simulates diverse black-box conditions, while an inner RL agent adaptively adjusts substitute-model weights, yielding highly transferable perturbations.
2) **Protocol-Aware Transferability Maximization.** Our attack explicitly respects TCP/IP field semantics and discrete feature ranges, yet remains effective across heterogeneous NIDS architectures—including both gradient-based deep networks and non-differentiable tree ensembles—thereby overcoming a key limitation of prior gradient-dependent transfer attacks.
3) **Comprehensive Empirical Validation.** Experiments on standard datasets BCCC-CIC-IDS2017 [35] and BCCC-CSE-CIC-IDS2018 [36] under closed-set, open-set, and cross-set scenarios demonstrate consistent gains in attack success rate over state-of-the-art black-box baselines, confirming both robustness and practical applicability.

II. Background and Related Work

This section introduces the fundamental concepts of Machine Learning-based Network Intrusion Detection Systems and Adversarial Learning techniques.

A. ML-based Network Intrusion Detection Systems

Based on the learning paradigm and detection objectives, ML-based NIDSes can be broadly categorized as follows.

Supervised Learning-based Classifiers: These NIDSes are trained on labeled datasets containing both benign and malicious traffic to learn decision boundaries that distinguish between different traffic classes. Common NID models using Support Vector Machines, Decision Trees, Random Forests, Logistic Regression, and various Deep Neural Network architectures such as Multi-Layer Perceptrons, Convolutional Neural Networks, and Recurrent Neural Networks and their variants (e.g., LSTM [25]). Research efforts like AlertNet [23], IdsNet [37], and DeepNet [24] have explored different DNN architectures for NID applications.

Unsupervised Learning-based Anomaly Detectors: These NIDSes primarily learn the patterns of normal network traffic and identify deviations from these patterns as anomalies or potential attacks. This approach is particularly suited for detecting zero-day attacks. Representative works include KitNET [1], which uses an ensemble of autoencoders, and methods based on isolation forests or clustering [22].

Hybrid or Ensemble Methods: To combine the strengths of different models, many studies employ ensemble learning strategies, such as XGBoost, Deep Forest, or construct systems integrating multiple base models [38], [39]. Some works also explore models based on temporal features, utilizing Markov models [40] or RNNs [41] to analyze packet sequences.

B. Adversarial Attacks against NIDSes

Adversarial attacks aim to deceive ML models by adding carefully crafted, often imperceptible perturbations to input data, causing the model to make incorrect predictions and thus bypass NIDS detection. Based on the adversary's knowledge of the target NIDS, adversarial attacks can be mainly classified as two categories.

1) White-box Attacks: White-box attacks assume the adversary has complete knowledge of the target model's structure, parameters, training data, etc. [4], [10], [42]. Adversaries can leverage this information (e.g., gradients) to precisely compute and generate the most effective adversarial perturbations. Early white-box attacks against NIDSes adapted gradient methods from the image domain to NID models [42]. Authors in [4] evaluate the effectiveness of different gradient attack algorithms against DNN-based NIDSes. Although these studies

revealed the vulnerability of ML-based NIDSes, the white-box assumption is generally unrealistic in real-world scenarios.

2) Black-box Attacks: Black-box typically assumes that the adversary cannot access the target model's internal information and must rely on interaction (queries) or general model vulnerabilities (transferability) to conduct the attack.

Query-based Attacks: The core strategy of these attacks involves interacting with the target NIDSes by sending probe traffic and observing the output (typically a binary label, benign or malicious). This feedback is then used to infer information about the model or directly guide the generation of adversarial examples. Several query-based techniques have been explored within the NIDS domain. For instance, Zolbayar et al. [43] adopts a GAN-based framework that directly leverages feedback from the target model to train a generator, aiming to produce realistic adversarial traffic. Model extraction techniques also represent a significant category, where queries are used to gather data for training a local substitute model that mimics the target [8]. Despite demonstrating the feasibility of black-box attacks, these methods generally suffer from low query efficiency, high cost, and high detectability, making them challenging to apply against well-defended NIDSes.

Transfer-based Attacks: These attacks exploit the property that adversarial examples carefully-crafted against one substitute model can often successfully deceive other unknown target models [11], [44]. Adversaries train or obtain one or more substitute models locally, generate adversarial examples on them, and then directly use these examples to attack the target NIDSes without any queries or interactions.The success of such attacks heavily depends on the transferability of the adversarial examples. **Zero-query attack research specifically for NIDS is relatively scarce and more challenging.** Han et al. [45] combined GANs for adversarial feature generation with PSO for traffic vector optimization, improving attack effectiveness in gray/black-box settings. Nasr et al. [46] sought universal perturbations under traffic constraints against DNN-based traffic analysis systems. However, due to the strong heterogeneity of NIDS models and strict constraints on traffic features, achieving high transferability in zero-query attacks remains difficult. Zhang et al. [7] noted that existing black-box attacks struggle with low success rates due to weak transferability between different model types (e.g., neural networks and tree models).

In summary, while ML-based NIDSes, encompassing diverse architectures from neural networks to tree-based ensembles, offer advanced threat detection capabilities, their vulnerability to adversarial attacks is a significant concern. White-box attacks, though demonstrating model weaknesses, rely on unrealistic assumptions of complete knowledge regarding the target NIDS. Black-box attacks present a more practical threat model but are themselves challenged: query-based approaches often suffer from inefficiency and high detectability when applied to NIDSes, which typically provide limited feedback. Consequently, transfer-based attacks, which generate adversarial examples on local substitute models without querying the target, are more aligned with realistic zero-query scenarios. However, the efficacy of such attacks is often hindered by the poor transferability of adversarial examples across heterogeneous NIDS models and the strict constraints imposed by network traffic validity. This paper introduces ARTEMIS, a framework designed to specifically address these critical challenges in achieving effective zero-query, transfer-based adversarial attacks against diverse ML-NIDSes.

III. Threat Model

In this section, we define the threat model considered for adversarial attacks against ML-based NIDSes. As a conclusion, the threat model investigated in this paper is a realistic and challenging **zero-query black-box scenario**. The adversary attempts to evade detection by constructing substitute models and leveraging the transferability of adversarial examples, while ensuring that the generated adversarial traffic adheres to network protocol and functional constraints.

A. Adversary's Knowledge

We consider a **black-box** attacker who *cannot* access the target NIDS architecture, parameters, or decision scores and receives *no* feedback during the attack (zero-query setting). Within this constraint, the adversary has the following concrete capabilities:

Training Data. The attacker cannot obtain the exact dataset used to train the target NIDS. Instead, she can access public traffic corpora (e.g., BCCC-CIC-IDS2017/2018 [35], [36]) or collect her own traffic to train local substitute models.

Feature Extractor. We assume the target NIDS relies on the flow-level statistics. Because these features are publicly documented, the adversary can reproduce the same extractor offline. When the feature sets coincide, our experiments give a strong-attacker upper bound; if the target employs additional proprietary fields, the shared subset still provides a realistic transfer channel.

Building Substitute Models. Using the available data and replicated features, the adversary trains one or more local substitute models to approximate the unseen target NIDS.

B. Notation

Key symbols used in this paper to describe the attack are summarized in Table I.

C. Formal Problem Statement

The adversary's primary objective is to generate a perturbation Δx that, when applied to an original malicious sample X, creates an adversarial sample $\hat{x} = X + \Delta x$. This sample $\hat{x}$ should be misclassified by the unknown target NIDS f_{tgt} while the perturbation remains imperceptible, i.e., $||\Delta x||_\infty \leq \epsilon$, and $\hat{x}$ adheres to network traffic constraints.

In the zero-query black-box setting, since f_{tgt} is unknown and unqueriable, the challenge lies in crafting an adversarial example $\hat{x}$ that maximizes transferability from a diverse ensemble of locally accessible substitute models $\mathcal{F}_{pool}$. The adversary's goal is to find a final adversarial sample $\hat{x}$ that maximizes the expected attack success rate against unseen

TABLE I: Summary of Notations

Symbol	Description
X	Original malicious network traffic sample
y	True label of the original malicious sample
$\hat{x}$	Adversarial example (final output)
ϵ	Maximum perturbation magnitude (L_∞ norm)
f_{tgt}	Unknown true target black-box NIDS model
$\mathcal{F}, \mathcal{F}_{pool}$	Pool of n (or N) pre-trained substitute models
$\mathcal{F}_{diff}$, $\mathcal{F}_{nondiff}$	Subsets of differentiable and non-differentiable models in $\mathcal{F}$, respectively
$\mathcal{F}_{src}$	Subset of m models sampled as source ensemble
W	Weights $\{w_1, \ldots, w_m\}$ for models in $\mathcal{F}_{src}$
f_{sim_tgt}	Model sampled from $\mathcal{F} \setminus \mathcal{F}_{src}$ as simulated target for a meta-task
$J(\cdot,\cdot)$	Loss function (e.g., cross-entropy)
I	Number of meta-task iterations
T	Number of outer loop (reweighing) iterations
K	Number of inner loop (perturbation generation) iterations
α	Step size for perturbation updates
μ	Momentum decay factor
Project$(\cdot, X, \epsilon)$	Projection function for constraints
$x^{(i)}$, $\bar{x}^{(t)}$, $\tilde{x}^{(k)}$	Adversarial samples at meta-task i, outer loop t, and inner loop k, respectively
$g_0, \bar{g}_0, g_t, \tilde{g}_k$	Momentum terms
f_ξ	Model sampled from $\mathcal{F}_{src}$ (index ξ)
$\nabla J_\xi^{(k)}$	Gradient for a differentiable model f_ξ
L_{sim}	Simulated logits for a non-differentiable model
$\nabla J_{sim}^{(k)}$	Pseudo-gradient for a non-differentiable model
R_t	Reward signal at outer loop t (based on $\mathcal{F}_{src}$ loss)
$g_{i,\text{target}}$	Gradient w.r.t. simulated target f_{sim_tgt} (for meta-task update)

target models drawn from a distribution $\mathcal{P}_{target}$. This can be formulated as:

$$\hat{x} \approx \arg\max_{x'} \mathbb{E}_{f'_{tgt} \sim \mathcal{P}_{target}} [\mathbb{I}(f'_{tgt}(x') \neq y)] \tag{1}$$

subject to $||x' - X||_\infty \leq \epsilon$ and traffic validity. $\mathbb{I}(\cdot)$ is the indicator function, and f'_{tgt} here represents a model drawn from the distribution of potential true targets. How such an optimization for $\hat{x}$ can be approached, particularly by leveraging strategies like ensemble reweighing internally within a meta-learning structure to improve the quality of x', will be detailed in the subsequent sections.

IV. SYSTEM DESIGN

A. Overall Framework

To solve the optimization problem of enhancing zero-query black-box attack transferability against diverse ML-NIDSes (Section III-C, Eq. 1), we propose ARTEMIS, a novel hybrid adversarial attack framework (Fig. 1, Algorithm 1). ARTEMIS systematically addresses critical challenges: it counters the zero-query unknown target model issue by performing meta-learning over an ensemble of substitute models ($\mathcal{F}_{pool}$) to simulate black-box attacks; it tackles ensemble model heterogeneity, including differentiable and non-differentiable types, using an RL-inspired reweighing mechanism to explore shared adversarial subspaces, with a unified momentum accumulation approach bridging the gap between model types; and it respects network data modification limits via a dedicated projection constraint module.

The meta-learning strategy is operationalized through a series of meta-tasks. Each meta-task is divided into a training phase and a testing phase. In the **training phase**, given a source ensemble $\mathcal{F}_{src}$ sampled from $\mathcal{F}_{pool}$, ARTEMIS performs a joint optimization. An *inner loop* iteratively crafts an adversarial perturbation. This loop uses momentum-based attacks (e.g., MI-FGSM) with Bayesian sampling for differentiable models in $\mathcal{F}_{src}$, and Leaf Tuple Attack for non-differentiable models, with a unified momentum term guiding the process. Concurrently, an *outer loop* uses an RL-inspired mechanism to adaptively reweigh models in $\mathcal{F}_{src}$ based on the inner loop's output. This phase yields an optimized adversarial sample $\bar{x}^{(T)}$ and ensemble weights $W^{(T)}$. In the subsequent **testing phase**, $\bar{x}^{(T)}$ is evaluated against a simulated target model f_{sim_tgt} (sampled from $\mathcal{F}_{pool} \setminus \mathcal{F}_{src}$). This black-box evaluation feedback is used to refine the main adversarial sample for the next meta-task.

Through iterative execution of these meta-tasks, ARTEMIS learns to produce adversarial samples with significantly improved transferability to unknown true target NIDS f_{tgt}. The detailed mechanisms of each component are elaborated in the

Algorithm 1 Hybrid Adversarial Attack

Input: Malicious sample X, true label y, surrogate ensemble $\mathcal{F} = \{f_1, \ldots, f_n\}$, initial weights $W = \{w_1, \ldots, w_n\}$, loss function J, iterations I, T, K, step size α, decay factor μ, bound ϵ, projection function Project$(\cdot, X, \epsilon)$.
Output: Final adversarial sample $\hat{x}$.

Initialize main sample $x^{(0)} \leftarrow X$; $W = W_{initial}$
for $i = 0$ to $I - 1$ **do** ▷ Meta-task loop
  Sample source models $\mathcal{F}_{src} \subseteq \mathcal{F}$, target $f_t \in \mathcal{F} \setminus \mathcal{F}_{src}$
  $\bar{x}^{(0)} \leftarrow x^{(i)}$; $g_0 \leftarrow 0$
  for $t = 0$ to $T - 1$ **do** ▷ Outer loop (Reweighing)
    $\tilde{x}^{(0)} \leftarrow \bar{x}^{(t)}$; $\tilde{g}_0 \leftarrow 0$
    for $k = 0$ to $K - 1$ **do** ▷ Inner loop (Perturbation generation)
      Sample model f_ξ from $\mathcal{F}_{src}$ (index ξ)
      if f_ξ is differentiable **then**
        Compute gradient $\nabla J_\xi^{(k)}$ using Bayesian sampling (Eq. 10)
      else (f_ξ is non-differentiable)
        Compute simulated logits L_{sim} (Eq. 13)
        Compute pseudo-gradient $\nabla J_{sim}^{(k)}$ based on $J(w_\xi^{(t)} L_{sim}(\tilde{x}^{(k)}), y)$
        Let $\nabla J_\xi^{(k)} \leftarrow \nabla J_{sim}^{(k)}$
      end if
      $\tilde{g}_{k+1} \leftarrow \mu \cdot \tilde{g}_k + \frac{\nabla J_\xi^{(k)}}{||\nabla J_\xi^{(k)}||_1}$
      $\tilde{x}'^{(k+1)} \leftarrow \tilde{x}^{(k)} + \alpha \cdot \text{sign}(\tilde{g}_{k+1})$
      $\tilde{x}^{(k+1)} \leftarrow \text{Project}(\tilde{x}'^{(k+1)}, X, \epsilon)$
    end for ▷ Result $\tilde{x}^{(K)}$, final inner momentum $\tilde{g}_K$
    Calculate Reward R_t (Eq. 6)
    Update weights $W^{(t+1)}$ based on Reward R_t(Eq. 7)
    $g_{t+1} \leftarrow \mu \cdot g_t + \frac{\tilde{g}_K}{||\tilde{g}_K||_1}$
    $\bar{x}'^{(t+1)} \leftarrow \bar{x}^{(t)} + \alpha \cdot \text{sign}(g_{t+1})$
    $\bar{x}^{(t+1)} \leftarrow \text{Project}(\bar{x}'^{(t+1)}, X, \epsilon)$
  end for ▷ Result $\bar{x}^{(T)}$ after T reweighing steps
  $g_{i,\text{target}} \leftarrow \nabla_{\bar{x}^{(T)}} J(f_t(\bar{x}^{(T)}), y)$
  $x'^{(i+1)} \leftarrow \bar{x}^{(T)} + \alpha \cdot \text{sign}(g_{i,\text{target}})$
  $x^{(i+1)} \leftarrow \text{Project}(x'^{(i+1)}, X, \epsilon)$
end for ▷ Final result $x^{(I)}$ after I meta-tasks
$\hat{x} \leftarrow x^{(I)}$
return $\hat{x}$

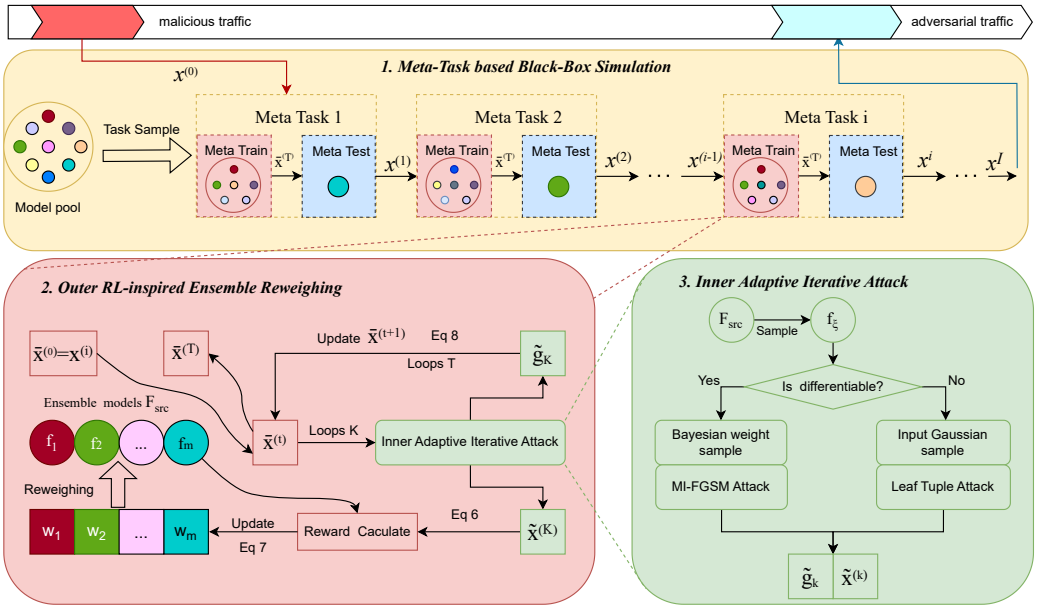

Fig. 1: The overview of our hybrid adversarial attack framework.

subsequent sections.

B. Meta-Task based Black-Box Simulation

We employ meta-learning to simulate the zero-query transfer challenge via I iterative meta-tasks. Each meta-task samples a source ensemble $\mathcal{F}_{src}^{(i)}$ and a simulated target $f_{sim\_tgt}^{(i)}$ from a diverse model pool $\mathcal{F}_{pool}$.

Within meta-task i, an inner joint optimization (detailed in Sections IV-D) seeks an effective adversarial example $\overline{x}_i^{(T)}$ and optimized ensemble weights $W^{(T)}$ for $\mathcal{F}_{src}^{(i)}$. Conceptually, this aims to achieve:

$$(\overline{x}_i^{(T)}, W^{(T)}) \approx \underset{\mathbf{x}', \mathbf{w}'}{\text{argmax}} J\left(\sum_{f_j \in \mathcal{F}_{src}^{(i)}} w_j' f_j(\mathbf{x}'), y\right) \quad (2)$$

Here, $W^{(T)}$ represents the weights after T reweighing iterations within this meta-task.

The resulting $\overline{x}_i^{(T)}$ then guides the update of the main adversarial sample from $x^{(i)}$ to $x^{(i+1)}$, based on its evaluation against $f_{sim\_tgt}^{(i)}$. Let $g_{src\_ens}^{(i)}$ represent the effective gradient direction obtained from attacking the source ensemble $\mathcal{F}_{src}^{(i)}$ with weights $W^{(T)}$ at sample $\overline{x}_i^{(T)}$:

$$g_{src\_ens}^{(i)} = \text{sign}(\nabla_{\overline{x}_i^{(T)}} J(\sum_{f_j \in \mathcal{F}_{src}^{(i)}} W_j^{(T)} f_j(\overline{x}_i^{(T)}), y)) \quad (3)$$

This meta-test conceptually optimizes $x^{(i+1)}$ by maximizing the loss on $f_{sim\_tgt}^{(i)}$ after taking a step from $\overline{x}_i^{(T)}$ in the direction $g_{src\_ens}^{(i)}$:

$$x^{(i+1)} \approx \underset{\mathbf{x}''}{\text{argmax}} J\left(f_{sim\_tgt}^{(i)}\left(\overline{x}_i^{(T)} + \alpha \cdot g_{src\_ens}^{(i)}\right), y\right) \quad (4)$$

The overall objective across I meta-tasks is to find the final adversarial sample $\hat{x} = x^{(I)}$ that maximizes expected attack success against unseen true target models $f_{tgt} \sim \mathcal{P}_{target}$:

$$\hat{x} \approx \underset{\mathbf{x}'}{\text{argmax}} \mathbb{E}_{f_{tgt} \sim \mathcal{P}_{target}}[\mathbb{I}(f_{tgt}(\mathbf{x}') \neq y)] \quad (5)$$

subject to $||\mathbf{x}' - X||_\infty \leq \epsilon$ and traffic validity. This bi-level structure aims to approximate this objective for enhanced zero-query transferability.

C. Outer RL-inspired Ensemble Reweighing

To dynamically harness the diverse contributions of models in $\mathcal{F}_{src}^{(i)}$, the outer loop adjusts the ensemble weights $\mathbf{w}^{(t)} = (w_1^{(t)}, \ldots, w_m^{(t)})$ using an RL-inspired approach and refines the adversarial sample $\overline{x}^{(t)}$. The update for weights is driven by a reward signal $R_t(\mathbf{w}^{(t)})$, which is precisely the weighted ensemble loss of the inner loop's output $\tilde{x}^{(K)}$ (generated from $\overline{x}^{(t)}$ using weights $\mathbf{w}^{(t)}$) on the source ensemble $\mathcal{F}_{src}^{(i)}$:

$$R_t(\mathbf{w}^{(t)}) = J\left(\sum_{j=1}^{m} w_j^{(t)} f_j(\tilde{x}^{(K)}), y\right) \quad (6)$$

The weights are updated to maximize this reward:

$$\mathbf{w}^{(t+1)} = \arg\max_{\mathbf{w}'} R_t(\mathbf{w}') \quad (7)$$

To practically implement the weight update, we transform this reward function into a minimization problem by considering the loss $J_R = -\ln(R_t(\mathbf{w}^{(t)}))$, which is then optimized using Stochastic Gradient Descent (SGD) to update the weights $\mathbf{w}^{(t)}$ to $\mathbf{w}^{(t+1)}$.

Concurrently, within this outer loop, the adversarial sample $\overline{x}^{(t)}$ is updated to $\overline{x}^{(t+1)}$. The conceptual objective for this

update is to find an $\overline{x}^{(t+1)}$ that maximizes the loss against the source ensemble $\mathcal{F}_{src}^{(i)}$ using the current weights $\mathbf{w}^{(t)}$:

$$\overline{x}^{(t+1)} \approx \underset{\mathbf{x}'}{\text{argmax}} J\left(\sum_{j=1}^{m} w_j^{(t)} f_j(\mathbf{x}'), y\right) \tag{8}$$

subject to $||\mathbf{x}' - X||_\infty \leq \epsilon$ and traffic validity constraints. This maximization is practically achieved by taking a gradient-based step, leveraging the aggregated momentum from the inner loop's attack, as detailed in Algorithm 1 (lines 23-25).

This adaptive reweighing of models and iterative refinement of the adversarial sample allows the framework to focus on promising subspace directions for generating universally effective adversarial examples. After T iterations, this outer loop yields an optimized sample $\overline{x}_i^{(T)}$ and corresponding weights $W^{(T)}$ for the current meta-task i.

D. Inner Adaptive Iterative Attack

Given a source ensemble $\mathcal{F}_{src}^{(i)}$ and weights $\mathbf{w}^{(t)}$, the inner loop iteratively generates an adversarial example $\tilde{x}^{(K)}$ from a starting point $\overline{x}^{(t)}$ over K steps. At each step k, a model f_ξ is sampled from $\mathcal{F}_{src}^{(i)}$. The perturbation Δx_k is computed using strategies tailored to the model type, aiming to maximize the weighted loss $J(w_\xi^{(t)} f_\xi(\tilde{x}^{(k)}), y)$, where y is the original malicious label. A key aspect is the unified accumulation of directional information via a momentum term $\tilde{g}_k$, even when handling non-differentiable models.

1) Attacking Differentiable Models ($\mathcal{F}_{diff}$): Inspired by [33], when f_ξ is differentiable, we enhance transferability by attacking a distribution over model parameters. We aim to optimize the expected loss over an approximate Bayesian posterior $p(\mathbf{w}|\mathcal{D})$:

$$\max_{\|\Delta x\|_p \leq \epsilon} \mathbb{E}_{\mathbf{w}\sim p(\mathbf{w}|\mathcal{D})}[J(w_\xi^{(t)} f_\xi(x_0 + \Delta x; \mathbf{w}), y)] \tag{9}$$

Solving this optimization question directly is challenging, so we use a Monte Carlo approach within the MI-FGSM [47] iterative framework. At each inner step k, we first sample a specific model instance by drawing weights $\mathbf{w}_\xi^{(t,k)} \sim p(\mathbf{w}|\mathcal{D})$. We then compute the gradient of the loss for this sampled instance:

$$\nabla J_\xi^{(k)} = \nabla_{\tilde{x}^{(k)}} J(w_\xi^{(t)} f_\xi(\tilde{x}^{(k)}; \mathbf{w}_\xi^{(t,k)}), y) \tag{10}$$

This per-sample gradient $\nabla J_\xi^{(k)}$ is used to update the momentum term $\tilde{g}_k$:

$$\tilde{g}_{k+1} = \mu \tilde{g}_k + \frac{\nabla J_\xi^{(k)}}{||\nabla J_\xi^{(k)}||_1} \tag{11}$$

Finally, the adversarial example is updated using the momentum direction and projected back into the allowed space:

$$\tilde{x}^{(k+1)} = \text{Project}_{X,\epsilon}(\tilde{x}^{(k)} + \alpha \cdot \text{sign}(\tilde{g}_{k+1})) \tag{12}$$

The projection function $Project_{X,\epsilon}(\cdot)$ is crucial for ensuring generated adversarial examples are both evasive and realistic, embodying our **protocol-aware** approach by enforcing key "traffic validity" constraints. Specifically, $Project_{X,\epsilon}(\cdot)$ ensures that adversarial examples remain within the ϵ-ball of the original sample X and adhere to necessary feature constraints. All numerical feature values are clipped to the minimum and maximum values observed in the training set. Flag-based features are kept unchanged to maintain protocol validity. Rate-based features (e.g., byte rate, down up rate) are recalculated based on the perturbed flow features (like total packets, total bytes) and the (potentially perturbed) duration features after each update step. Other features with inherent mathematical dependencies (e.g., max $>$ mean $>$ min for packet sizes) are adjusted post-perturbation to preserve these valid relationships. By applying these constraints, $Project_{X,\epsilon}(\cdot)$ ensures ARTEMIS generates adversarial traffic that is effective, adheres to network communication rules.

2) Attacking Non-Differentiable Models ($\mathcal{F}_{nondiff}$): For tree ensembles $f_\xi \in \mathcal{F}_{nondiff}$, where gradients are undefined, the primary mechanism to find an effective perturbation Δx is the Leaf Tuple Attack [34]. This method efficiently searches the discrete leaf space to find a nearby adversarial leaf tuple $\mathcal{C}^*$ minimizing $dist_p(\mathcal{C}^*, x_0)$ such that $f_\xi(x_0 + \Delta x) \neq y$. During the attack, input Gaussian sampling is performed.

To integrate these models into the unified momentum framework (Eq. 11), we derive *simulated logits*, denoted as $L_{sim}(\tilde{x}^{(k)})$, from the ensemble's output probabilities $P_\xi(y|\tilde{x}^{(k)})$. For binary classification, this can be the logit function:

$$L_{sim}(\tilde{x}^{(k)}) = \text{logit}(P_\xi(y|\tilde{x}^{(k)})) = \log\left(\frac{P_\xi(y|\tilde{x}^{(k)})}{1 - P_\xi(y|\tilde{x}^{(k)})}\right) \tag{13}$$

While the Leaf Tuple Attack identifies the actual perturbation, we can compute a pseudo-gradient, denoted as $\nabla J_{sim}^{(k)}$, based on the loss using these simulated logits, $J(w_\xi^{(t)} L_{sim}(\tilde{x}^{(k)}), y)$. This pseudo-gradient $\nabla J_{sim}^{(k)}$ is then used in the momentum update (Eq. 11), replacing $\nabla J_\xi^{(k)}$. This allows the momentum term $\tilde{g}_k$ to consistently aggregate directional information across both differentiable and non-differentiable models, guiding the overall search direction, even though the step Δx itself for tree models is determined by the Leaf Tuple Attack. The sample update then follows Eq. 12.

V. Experiments

In this section, we conduct a series of comprehensive experiments to rigorously evaluate the performance and characteristics of our proposed ARTEMIS framework. We begin by detailing the common experimental setup in Section V-A Subsequently, our evaluation is structured as follows: First, we assess the transferability of adversarial examples generated by ARTEMIS under three distinct scenarios(detailed in Sections V-B, V-C, and V-D), representing varying degrees of data distribution similarity between substitute and target models:

- **Closed-set scenario**: Substitute and target models are trained and tested on data from the same dataset distribution.

- **Open-set scenario**: Substitute and target models are trained on mutually exclusive data subsets drawn from the same underlying dataset distribution, simulating a common practical challenge.
- **Cross-set scenario**: Substitute models are trained on one dataset and target models on another, and vice-versa, to evaluate robustness against significant distributional shifts.

Second, we compare ARTEMIS against several state-of-the-art baseline attack methods (Section V-A4) to demonstrate its relative advantages in the zero-query black-box setting. Finally, we delve into the internal workings of ARTEMIS through a **framework analysis**(Section V-E), which includes an examination of the meta-task optimization process and a **component ablation study** to quantify the contributions of its key mechanisms.

A. Experimental Setup

1) Datasets: Our experiments utilize the BCCC-CIC-IDS2017 (henceforth IDS2017) [35] and BCCC-CSE-CIC-IDS2018 (henceforth IDS2018) [36] datasets. These datasets represent enhanced versions of the original CIC-IDS2017/2018 benchmarks [6].

We employ the NTLFlowLyzer [35] as the feature extractor. From its extensive feature set, we select 60 commonly used features for network intrusion detection, covering categories such as time-based, rate-based, payload-based, flag-based, and header-based characteristics, ensuring relevance and comparability. For the IDS2018 dataset, we randomly sample 50,000 instances for each attack type. Due to smaller data volumes for the Brute force and Botnet attack types in the IDS2017 dataset, all available instances are utilized for these specific categories; for other attack types in IDS2017, 50,000 instances are also randomly sampled. All datasets are subsequently partitioned into training and testing sets using a 4:1 ratio.

For the closed-set and open-set scenarios, we generate adversarial examples from 500 malicious samples per attack type (Brute_force, Botnet, DDoS, DoS Hulk), which are all initially correctly classified by the respective target models. For the cross-dataset experiment—for example, transferring attacks from IDS2017 (source) to IDS2018 (target)—we randomly sample 500 malicious flows from IDS2017, retain only those that the target NIDS trained on IDS2018 initially classifies correctly, generate adversarial versions of this filtered set with substitute models trained on IDS2017, and then submit the resulting adversarial flows to the IDS2018 model to measure cross-dataset transferability.

2) Models: Our substitute-model pool $\mathcal{F}_{\text{pool}}$ comprises five diverse architectures commonly used in NIDS: XGBoost (XGB), Random Forest (RF), Multi-Layer Perceptron (MLP), LeNet [48], and Long Short-Term Memory (LSTM) networks. These cover both non-differentiable (ND) models (XGB, RF) and differentiable (D) models (MLP, LeNet, LSTM).

We evaluate transferability against 14 target models: five in the white-box (WB) setting and nine in the black-box (BB) setting. Their names, types, and differentiability status are summarized in Table II.

TABLE II: Target Model Definitions. WB denotes white-box; BB denotes black-box.

White-box Models			Black-box Models		
ID	**Name**	**Type (ND/D)**	**ID**	**Name**	**Type (ND/D)**
WB1	XGBoost-WB	Ensemble (ND)	BB1	RF-BB	Ensemble (ND)
WB2	RF-WB	Ensemble (ND)	BB2	XGBoost-BB	Ensemble (ND)
WB3	MLP-WB	Neural Net (D)	BB3	KitNET [1]	Autoencoder (D)
WB4	LeNet-WB	CNN (D)	BB4	MAMPF [40]	Markov (ND)
WB5	LSTM-WB	RNN (D)	BB5	FS-Net [41]	RNN (D)
			BB6	AlexNet-NIDS	CNN (D)
			BB7	AlertNet [23]	CNN (D)
			BB8	DeepNet [24]	DNN (D)
			BB9	IdsNet [37]	DNN (D)

3) Attack Implementation: We implement ARTEMIS as detailed in Algorithm 1. The source code for our framework is publicly available to facilitate reproducibility[1]. We adopt the default hyper-parameters summarized in Table III.

TABLE III: Default hyper-parameters for all attack experiments

Parameter	**Default setting**
Perturbation constraint (L_∞)	$\epsilon \in \{0.05,\ 0.10\}$
Meta-task iterations (I)	10
Weight-update iterations (T)	5
Adaptive attack iterations(K)	10
Bayesian weight samples (diff.)	5
Gaussian input samples (non-diff.)	5
Step size (α)	ϵ/T
Momentum decay (μ)	1.0

4) Baseline Methods: We compare our hybrid attack framework against five representative adversarial attack methods. All baselines are implemented under a unified black-box setting: adversarial examples are generated by attacking ensemble substitute models and then transferred to multiple target NID models for evaluation. Four of the attacks (JSMA, C&W, ZOO, HSJA) are implemented using the Adversarial Robustness Toolbox (ART v1.16.0 [49]), whereas ETA is re-implemented from its original publication.

- **Jacobian-based Saliency Map Attack (JSMA)** [50]: A white-box attack that perturbs input features based on their saliency with respect to the target class.
- **Carlini & Wagner (C&W)** [9]: An optimization-based attack that minimizes the L_2 norm of perturbations while achieving misclassification.
- **Zeroth Order Optimization (ZOO)** [14]: A score-based attack that approximates gradients using finite differences on output confidence scores.
- **HopSkipJumpAttack (HSJA)** [51]: A decision-based attack that finds adversarial examples by iteratively approximating the decision boundary using hard-label outputs.
- **Explainable and Transferable Attack (ETA)** [7]: A recent saliency-guided transfer attack designed for NIDS

[1]https://github.com/wanglei0208/ARTIMIS

that perturbs inputs based on ensemble gradient attributions.

All attacks are applied to a shared subset of 500 correctly classified malicious flows. ART-based attacks are configured with consistent parameters: C&W and ZOO use 50 optimization steps with confidence set to 0.0; JSMA uses $\theta = 0.05$ and $\gamma = 0.1$; HSJA runs for 30 boundary evaluations. All adversarial examples are clipped to the $[0, 1]$ range and refined using protocol-aware constraints to ensure semantic plausibility.

5) Evaluation Metrics: 1. **Detection Performance (Baseline):** We report the standard classification Accuracy and F1-Score of the target models on the original (non-adversarial) test data to establish their baseline detection capabilities. It is noteworthy that all models, when trained on their respective training sets as defined in Section V-A1, achieve both F1-scores and Accuracy values exceeding 0.98 on the corresponding clean test sets.

2. **Attack Success Rate (ASR):** The Attack Success Rate (ASR) quantifies the effectiveness of the adversarial attack by measuring the proportion of generated examples that successfully mislead the target model to output an incorrect prediction. It can be calculated as:

$$\text{ASR} = \frac{\text{Number of successful attacks}}{\text{Total number of attack attempts}} \tag{14}$$

Here, the "Total number of attack attempts" refers to the set of malicious samples selected for adversarial example generation, as detailed in Section V-A1.

B. Transferability Evaluation: Closed-set Scenario

1) Performance of ARTEMIS: In the closed-set scenario, our proposed ARTEMIS framework demonstrated strong performance. As shown in Table IV, on the IDS2018 dataset with $\epsilon = 0.1$, the average Attack Success Rate (ASR) across all 14 target models is 86.7%, and 88.9% on the IDS2017 dataset. Even with a smaller perturbation ($\epsilon = 0.05$), the average ASRs remained 80.3% on IDS2018 and 83.4% on IDS2017.

2) Comparison with Baseline Methods: Figure 2 further illustrates that ARTEMIS consistently achieves a higher average ASR across all four evaluated attack types compared to the other black-box attack methods. This superior performance stems from our framework's integration of meta-learning for generalizable attack strategies, RL-inspired dynamic weighting of substitute models, and a hybrid attack mechanism tailored for both differentiable and non-differentiable models, all operating in a zero-query manner. In contrast, baseline methods either rely on query feedback (e.g., ZOO, HSJA), which is not applicable here, or exhibit lower transferability in this challenging zero-query NIDS environment.

C. Transferability Evaluation: Open-set Scenario

In the open-set scenario, our method continues to demonstrate effective transferability, albeit with an expected slight decrease in performance compared to the closed-set scenario, reflecting the challenges of data distribution shifts.

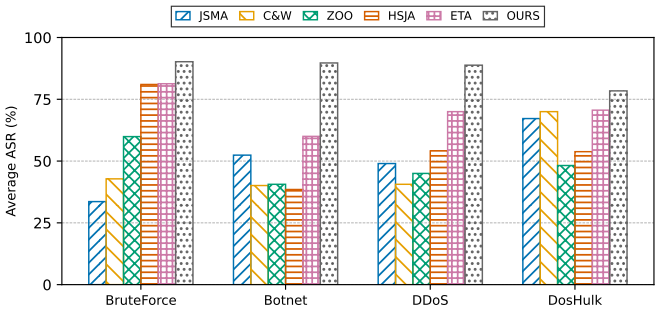

Fig. 2: Average attack success rate (ASR) of six black-box methods on four attack types.

Comparing these results with the closed-set performance on IDS2018 (Table IV, $\epsilon = 0.1$), where the average ASR for Brute force is 90.2% and for Botnet is 89.7%, we observe a modest reduction. This reduction highlights the challenge posed by variations in training data. Nevertheless, the consistently high ASRs underscore our framework's robustness and its ability to generate transferable adversarial examples even when the adversary lacks access to the target model's exact training data distribution.

D. Transferability Evaluation: Cross-set Scenario

The cross-set scenario represents the most challenging test for transferability. Results in Table VI (with $\epsilon = 0.1$) show *Filtered Num* (malicious samples out of 500 correctly identified by the target model from the other dataset) and the ASR on these filtered samples.

Transfer from IDS2017 to IDS2018 (17→18): When attacking IDS2018 targets using substitutes trained on IDS2017, the initial susceptibility of these targets to the out-of-distribution IDS2017 samples varies significantly, as evidenced by considerable fluctuations in *Filtered Num*. This variability highlights differing decision boundary sensitivities of models trained on distinct data distributions. Despite this, for the *Filtered Num* samples that are initially correctly identified, our method often achieves high ASRs, indicating ARTEMIS effectively crafts potent adversarial examples against recognized out-of-distribution attacks.

Transfer from IDS2018 to IDS2017 (18→17): Conversely, we observe that transfer from IDS2018 to IDS2017 generally yields stronger results. In this direction, *Filtered Num* is often higher, and subsequent ASRs are consistently strong across attack types. This suggests that adversarial examples derived from models trained on more extensive data (IDS2018) exhibit higher potency against models trained on less extensive data (IDS2017).

Overall, these cross-set findings highlight the inherent difficulties of transferring attacks across disparate data distributions. Nevertheless, ARTEMIS consistently achieves high ASRs on recognized out-of-distribution samples, and the observed asymmetry in transferability suggests that dataset characteristics, such as volume and diversity, play a crucial role in attack potency.

TABLE IV: ASR (%) of ARTEMIS for IDS2017 in the Closed-set Scenario

Attack Type	ϵ	White-Box Models					Black-Box Models								
		XGB	RF	MLP	LeNet	LSTM	XGB	RF	MAMPF	FSNET	KITNET	AlexNet	AlertNet	DeepNet	IdsNet
brute_force	0.05	97.6	100.0	87.2	85.8	85.2	97.6	99.0	94.0	77.2	100.0	73.8	88.8	79.2	89.8
	0.1	94.6	99.2	96.2	90.6	95.8	94.6	97.4	94.6	90.0	100.0	73.6	95.2	94.2	95.8
botnet	0.05	84.6	83.8	64.0	98.4	89.2	84.6	96.8	90.4	37.2	96.8	32.4	95.8	49.0	96.8
	0.1	92.8	86.4	91.4	99.4	99.8	92.8	95.8	97.4	62.4	99.6	30.4	99.8	63.8	99.8
ddos	0.05	98.8	97.0	93.2	60.2	70.2	97.0	99.0	96.2	94.0	97.8	56.0	96.4	90.2	92.2
	0.1	99.0	97.2	93.4	65.8	95.2	99.4	97.0	90.6	95.2	98.8	64.8	90.8	93.8	90.8
dos_hulk	0.05	90.8	90.0	70.4	45.0	82.0	92.8	92.2	89.2	53.8	98.6	66.8	80.0	60.0	65.0
	0.1	96.8	92.0	75.4	65.2	94.8	94.8	94.2	92.2	65.8	98.8	75.8	82.0	70.6	85.0

TABLE V: ASR(%) of the ARTEMIS for IDS2018 in the Open-set Scenario

Attack Type	ϵ	White-Box Models					Black-Box Models								
		XGB	RF	MLP	LeNet	LSTM	XGB	RF	MAMPF	FSNET	KITNET	AlexNet	AlertNet	DeepNet	IdsNet
brute_force	0.05	98.6	100.0	79.0	77.4	85.2	98.6	96.8	85.0	90.0	90.0	60.0	71.2	79.0	84.6
	0.10	99.4	99.8	85.8	70.6	94.4	95.2	99.4	87.2	95.6	94.0	66.2	77.2	87.6	91.6
botnet	0.05	87.0	92.4	15.2	29.8	70.8	89.8	87.0	53.2	79.8	93.0	7.4	38.2	43.0	15.0
	0.10	99.0	99.2	56.6	80.2	98.2	93.6	99.0	96.8	98.6	94.0	28.4	93.8	77.2	70.4
dos_hulk	0.05	84.6	96.6	35.6	21.2	42.0	84.6	97.8	29.0	32.4	92.0	33.2	35.6	31.6	28.6
	0.10	84.6	96.6	69.2	46.0	72.8	79.8	72.0	63.0	75.4	90.0	50.2	70.4	62.2	72.8
ddos	0.05	99.8	100.0	92.4	26.2	90.6	99.4	99.8	95.2	77.2	92.0	54.0	94.2	92.0	94.2
	0.10	99.6	99.8	88.8	25.6	95.2	99.0	99.6	88.8	87.2	94.0	61.2	88.8	88.4	88.4

TABLE VI: ASR(%) of the ARTEMIS for IDS2018 in the Cross-set Scenario

Cross-Set	Attack	Metric	White-Box Models					Black-Box Models								
			XGB	RF	MLP	LeNet	LSTM	XGB	RF	MAMPF	FSNET	KITNET	AlexNet	AlertNet	DeepNet	IdsNet
17→18	brute_force	Filtered Num	82	77	190	157	134	80	82	141	196	79	214	132	131	135
		ASR (%)	100.0	100.0	83.2	73.8	32.0	100.0	100.0	84.4	66.8	98.8	68.2	58.4	67.2	87.4
	botnet	Filtered Num	0	59	65	78	63	0	63	435	65	350	435	65	435	435
		ASR (%)	—	100.0	96.0	69.2	95.2	—	96.8	77.7	100.0	100.0	2.3	64.6	94.9	53.5
18→17	brute_force	Filtered Num	372	0	81	115	150	372	0	134	269	0	33	271	146	165
		ASR (%)	82.8	—	100.0	49.6	75.3	82.8	—	78.4	34.9	—	78.8	83.4	100.0	90.3
	botnet	Filtered Num	246	0	246	246	246	246	5	246	246	246	190	246	246	246
		ASR (%)	61.8	0.0	100.0	97.9	100.0	61.8	100.0	100.0	100.0	100.0	76.0	100.0	100.0	100.0

E. Framework Analysis and Ablation Study

1) Effectiveness of the Meta-task Optimization: To verify that our meta-learning strategy effectively guides adversarial samples towards the adversarial subspace intersection of the substitute models, we track six metrics after each meta-task. These metrics, averaged over 50 randomly selected malicious flows from the IDS-2018 dataset (Brute-Force attacks) and smoothed using a centered moving average window of three tasks, are defined as follows:

- **Ensemble Loss** — cross-entropy of the weighted substitute ensemble;
- **E2M(Ensemble-to-Member alignment)** — average cosine similarity between each substitute's gradient and the weighted ensemble gradient;
- **VictAlign(Victim-Gradient Alignment)** — cosine similarity between the weighted ensemble gradient and the simulated victim-model gradient;
- **W-PCS(Weight-averaged Pairwise Cosine Similarity)** — mean pairwise cosine similarity of substitute gradients, weighted by their current meta-weights;

As illustrated in Figure 3, we can make the following key observations:

(a) *Loss increase.* The Ensemble Loss climbs from 10.5 to 27.4, confirming that the perturbations are consistently moving to regions of higher ensemble risk.

(b) *Gradient alignment.* Both E2M and VictAlign rise from ≈ 0.85 to > 0.99, indicating that the learned momentum quickly aligns with the ensemble gradient and, more importantly, with the (unseen) simulated target.

These converging trends provide strong evidence that the proposed meta-learning mechanism fulfills its design goal: every meta-task acts as a projection step towards the common adversarial intersection, thereby facilitating high transferability in the zero-query setting.

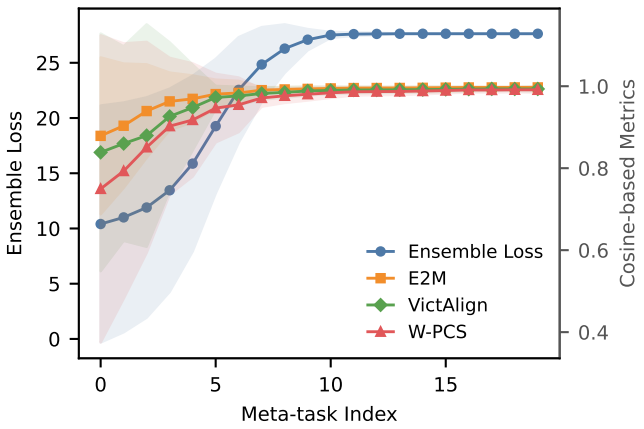

Fig. 3: Convergence of loss- and gradient-based metrics over 20 meta-tasks. Shaded bands represent mean $\pm$ 1 standard deviation.

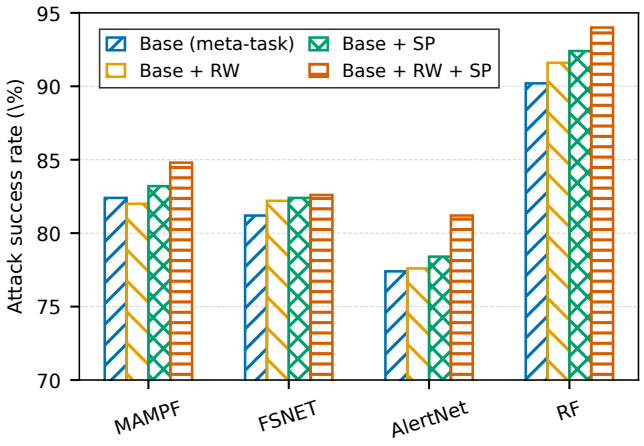

Fig. 4: Impact of key components (RW: adaptive reweighing, SP: Bayesian/Input sampling) on average ASR across four target models using IDS-2018 brute-force flows ($\epsilon = 0.05$, 10 meta-tasks, 500 samples).

2) Component Ablation: To assess the individual contributions of our framework's core components, we conducted an ablation study, with results presented in Figure 4. The *Base* configuration, relying solely on meta-task iterations, achieved an average Attack Success Rate (ASR) of 82.8% across four diverse target models. Introducing the adaptive reweighing mechanism (RW) or the Bayesian/input sampling strategy (SP) independently improved the ASR to 83.4% and 84.1%, respectively. Notably, enabling both RW and SP components concurrently yielded the highest ASR of 85.7%, an overall increase of 2.9 percentage points over the Base. This demonstrates that while each component offers a distinct benefit, their synergy provides the most significant enhancement to attack transferability, with the largest individual gain of 3.8 percentage points observed on AlertNet, underscoring their complementary nature.

VI. Discussion

In this section, we discuss two practical considerations for ARTEMIS: computational cost and the generation of actual attack traffic.

A. On Computational Cost

The computational cost of ARTEMIS is composed of two parts: a one-time, offline training phase for the substitute models, and an online, iterative process to generate each adversarial sample. The overhead of this process, while considerable, is acceptable as it is designed as an offline vulnerability assessment tool, not a real-time attack system. Its purpose is to conduct in-depth robustness analysis, justifying the computational investment.

B. On Generating Actual Attack Traffic

Reverse-mapping feature perturbations to network packets is a known challenge due to the many-to-one relationship. Our framework operates at the feature level to efficiently identify shared vulnerability directions. These findings can then guide the manual or semi-automated modification of real traffic captures. We consider the module to generate real adversarial traffic a parallel research topic, and hence it is not the focus of our study here. A key direction for future work is to establish a more formulaic correlation between the feature and traffic spaces. Such a correlation would allow traffic-domain constraints to be directly translated into the feature space, thus enabling the crafting of adversarial examples that are both more realistic and effective.

VII. Conclusion

In this paper, we propose ARTEMIS, an attack framework that can successfully enhances zero-query black-box attack transferability against diverse NIDSes by synergistically combining meta-learning, RL-inspired adaptive reweighing, and hybrid attack techniques tailored for both differentiable and non-differentiable models. The comprehensive experiments validate its superior performance compared to baselines. Overall, ARTEMIS offers a robust methodology for assessing NIDS vulnerabilities in realistic zero-query settings and underscores the importance of developing defenses resilient to sophisticated, transfer-based attacks. While ARTEMIS shows significant promise, it still faces several limitations, including the limited scope of datasets and substitute models used, the complexity of fully enforcing network traffic constraints, and the computational cost of the meta-learning process. Our future work will focus on broader evaluations on real-world and encrypted traffic, integration with adaptive defenses, advanced constraint optimization and efficiency improvements.

VIII. Acknowledgment

This work is supported by Shenzhen R&D Program under grant NO. KJZD20230923114059020, and in part by Shenzhen Key Laboratory of Software Defined Networking under Grant ZDSYS20140509172959989.

REFERENCES

[1] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[3] C. Fu, Q. Li, M. Shen, and K. Xu, "Realtime robust malicious traffic detection via frequency domain analysis," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 3431–3446.

[4] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38 367–38 384, 2018.

[5] A. Verma and V. Ranga, "Statistical analysis of cidds-001 dataset for network intrusion detection systems using distance-based machine learning," *Procedia Computer Science*, vol. 125, pp. 709–716, 2018.

[6] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[7] H. Zhang, D. Han, S. Zhuang, Z. Wang, J. Sun, Y. Liu, J. Liu, and J. Dong, "Explainable and transferable adversarial attack for ml-based network intrusion detectors," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–18, 2025.

[8] H. Yan, X. Li, W. Zhang, R. Wang, H. Li, X. Zhao, F. Li, and X. Lin, "Automatic evasion of machine learning-based network intrusion detection systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 1, pp. 153–167, 2024.

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. Ieee, 2017, pp. 39–57.

[10] J. Clements, Y. Yang, A. A. Sharma, H. Hu, and Y. Lao, "Rallying adversarial techniques against deep learning for network security," in *2021 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2021, pp. 01–08.

[11] M. Shen, C. Li, Q. Li, H. Lu, L. Zhu, and K. Xu, "Transferability of white-box perturbations: query-efficient adversarial attacks against commercial dnn services," in *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, 2024, pp. 2991–3008.

[12] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," *arXiv preprint arXiv:1807.04457*, 2018.

[13] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.

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

[15] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 742–749.

[16] A. Ilyas, L. Engstrom, and A. Madry, "Prior convictions: Black-box adversarial attacks with bandits and priors," *arXiv preprint arXiv:1807.07978*, 2018.

[17] J. Zhang, Y. Huang, W. Wu, and M. R. Lyu, "Transferable adversarial attacks on vision transformers with token gradient regularization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 16 415–16 424.

[18] J. Zhang, J.-t. Huang, W. Wang, Y. Li, W. Wu, X. Wang, Y. Su, and M. R. Lyu, "Improving the transferability of adversarial samples by path-augmented method," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 8173–8182.

[19] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2730–2739.

[20] S. Chen, N. Carlini, and D. Wagner, "Stateful detection of black-box adversarial attacks," in *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence*, 2020, pp. 30–39.

[21] X. Wang, K. Chen, X. Ma, Z. Chen, J. Chen, and Y.-G. Jiang, "Advqdet: Detecting query-based adversarial attacks with adversarial contrastive prompt tuning," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 6212–6221.

[22] P.-F. Marteau, "Random partitioning forest for point-wise and collective anomaly detection—application to network intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2157–2172, 2021.

[23] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE access*, vol. 7, pp. 41 525–41 550, 2019.

[24] M. Gao, L. Ma, H. Liu, Z. Zhang, Z. Ning, and J. Xu, "Malicious network traffic detection based on deep neural networks and association analysis," *Sensors*, vol. 20, no. 5, p. 1452, 2020.

[25] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *2016 international conference on platform technology and service (PlatCon)*. IEEE, 2016, pp. 1–5.

[26] S. Ennaji, E. Benkhelifa, and L. V. Mancini, "Toward realistic adversarial attacks in ids: A novel feasibility metric for transferability," *arXiv preprint arXiv:2504.08480*, 2025.

[27] Y. Chen, R. Guan, X. Gong, J. Dong, and M. Xue, "D-dae: Defense-penetrating model extraction attacks," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 382–399.

[28] Z. Ge, H. Liu, W. Xiaosen, F. Shang, and Y. Liu, "Boosting adversarial transferability by achieving flat local maxima," *Advances in Neural Information Processing Systems*, vol. 36, pp. 70 141–70 161, 2023.

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

[30] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[31] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, "Modeling realistic adversarial attacks against network intrusion detection systems," *Digital Threats: Research and Practice (DTRAP)*, vol. 3, no. 3, pp. 1–19, 2022.

[32] N. Alhussien, A. Aleroud, A. Melhem, and S. Y. Khamaiseh, "Constraining adversarial attacks on network intrusion detection systems: transferability and defense analysis," *IEEE Transactions on Network and Service Management*, vol. 21, no. 3, pp. 2751–2772, 2024.

[33] Q. Li, Y. Guo, W. Zuo, and H. Chen, "Making substitute models more bayesian can enhance transferability of adversarial examples," *arXiv preprint arXiv:2302.05086*, 2023.

[34] C. Zhang, H. Zhang, and C.-J. Hsieh, "An efficient adversarial attack for tree ensembles," *Advances in neural information processing systems*, vol. 33, pp. 16 165–16 176, 2020.

[35] M. Shafi, A. H. Lashkari, and A. H. Roudsari, "Ntlflowlyzer: Towards generating an intrusion detection dataset and intruders behavior profiling through network and transport layers traffic analysis and pattern extraction," *Computers & Security*, vol. 148, p. 104160, 2025.

[36] ——, "Toward generating a large scale intrusion detection dataset and intruders behavioral profiling using network and transportation layers traffic flow analyzer (ntlflowlyzer)," *Journal of Network and Systems Management*, vol. 33, no. 2, p. 44, 2025.

[37] B.-E. Zolbayar, R. Sheatsley, P. McDaniel, M. J. Weisman, S. Zhu, S. Zhu, and S. Krishnamurthy, "Generating practical adversarial network traffic flows using nidsgan," *arXiv preprint arXiv:2203.06694*, 2022.

[38] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer networks*, vol. 174, p. 107247, 2020.

[39] A. H. Farooqi, S. Akhtar, H. Rahman, T. Sadiq, and W. Abbass, "Enhancing network intrusion detection using an ensemble voting classifier for internet of things," *Sensors*, vol. 24, no. 1, p. 127, 2023.

[40] C. Liu, Z. Cao, G. Xiong, G. Gou, S.-M. Yiu, and L. He, "Mampf: Encrypted traffic classification based on multi-attribute markov probability fingerprints," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–10.

[41] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-net: A flow sequence network for encrypted traffic classification," in *IEEE INFOCOM 2019-IEEE Conference On Computer Communications*. IEEE, 2019, pp. 1171–1179.

[42] M. Rigaki, "Adversarial deep learning against intrusion detection classifiers," 2017.

[43] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," in *Pacific-asia conference on knowledge discovery and data mining*. Springer, 2022, pp. 79–91.
[44] Z. Fang, T. Wang, L. Zhao, S. Zhang, B. Li, Y. Ge, Q. Li, C. Shen, and Q. Wang, "Zero-query adversarial attack on black-box automatic speech recognition systems," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 630–644.
[45] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, and X. Yin, "Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2632–2647, 2021.
[46] M. Nasr, A. Bahramali, and A. Houmansadr, "Defeating {DNN-Based} traffic analysis systems in {Real-Time} with blind adversarial perturbations," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2705–2722.
[47] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.
[48] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[49] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.0.0," 2019. [Online]. Available: https://arxiv.org/abs/1807.01069
[50] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.
[51] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*. IEEE, 2020, pp. 1277–1294.