

DYNAMITE: Dynamic Defense Selection for Enhancing Machine Learning-based Intrusion Detection Against Adversarial Attacks

Jing Chen*, Onat Gungor*, Zhengli Shang, Elvin Li, Tajana Rosing

Department of Computer Science and Engineering, University of California, San Diego

{*jic128, ogungor, z4shang, ell009, tajana*}@ucsd.edu

Abstract—The rapid proliferation of the Internet of Things (IoT) has introduced substantial security vulnerabilities, highlighting the need for robust Intrusion Detection Systems (IDS). Machine learning-based intrusion detection systems (ML-IDS) have significantly improved threat detection capabilities; however, they remain highly susceptible to adversarial attacks. While numerous defense mechanisms have been proposed to enhance ML-IDS resilience, a systematic approach for selecting the most effective defense against a specific adversarial attack remains absent. To address this challenge, we propose *Dynamite*, a dynamic defense selection framework that enhances ML-IDS by intelligently identifying and deploying the most suitable defense using a machine learning-driven selection mechanism. Our results demonstrate that *Dynamite* achieves a 96.2% reduction in computational time compared to the Oracle, significantly decreasing computational overhead while preserving strong prediction performance. *Dynamite* also demonstrates an average F1-score improvement of 76.7% over random defense and 65.8% over the best static state-of-the-art defense.

I. INTRODUCTION

The Internet of Things (IoT) systems connect numerous devices that communicate and share data, enabling smart applications in sectors like healthcare, manufacturing, and transportation [1]. IoT systems are particularly susceptible to cyber threats due to their inter-connectivity, resource constraints, and diverse configurations [2]. Consequently, ensuring robust security measures is essential to safeguard these systems against potential attacks. Intrusion Detection Systems (IDS) play a crucial role in identifying and responding to malicious activities within IoT networks by monitoring network traffic and system behavior [1]. The integration of machine learning (ML) into IDS has significantly improved their effectiveness in detecting and mitigating cyber threats. ML-IDS possess the capability to analyze vast amounts of data, identify latent patterns, and detect cyberattacks that conventional methods may overlook [3]. Thus, ML-IDS serve as a robust approach for enhancing IoT security by addressing evolving threats. However, the rise of adversarial attacks poses a significant challenge to the effectiveness of ML-IDS [4]. These attacks allow malicious activities to go undetected and harm the

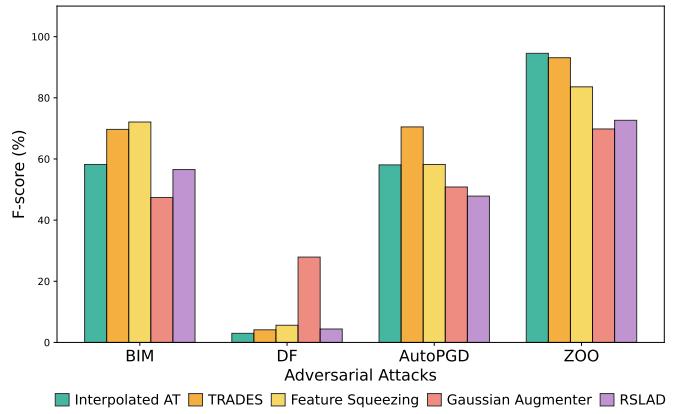


Fig. 1. SOTA Defense Performance Against Adversarial Attacks

security of IoT systems, leading to compromised operations, data breaches, and significant financial losses [5].

Developing effective defenses against adversarial attacks is crucial for maintaining the reliability and robustness of ML-IDS [6]. Several strategies, both general and specific to ML-IDS, have been proposed, including adversarial training [7], [8], modifications to the training process [9], [10], input transformation techniques [11], and methods for adversarial attack detection [12], [13]. However, the effectiveness of defense mechanisms varies depending on the specific type of attack they are intended to mitigate [14]. Given that adversarial attacks can differ in their techniques and objectives, tailored defense strategies are necessary to effectively address each distinct scenario. Fig. 1 demonstrates that no single defense model (represented by different colors) is universally effective against all adversarial attacks, with the optimal defense varying depending on the specific nature of the attack (as shown on the x-axis). This variability highlights the limitation of relying on a singular defense mechanism for comprehensive protection. It further emphasizes the importance of a dynamic defense selection mechanism that adaptively assigns the most appropriate defense for each attack scenario. Such an approach is crucial for achieving robust security, as it ensures the real-time deployment of the most effective defense in response to

*Both authors contributed equally to this research.

the evolving nature of adversarial attacks.

We propose an adaptive ML-IDS defense framework that ensures robust protection by dynamically selecting the most suitable defense for each adversarial attack. In contrast to traditional approaches that rely on static defenses, our framework adaptively mitigates the impact of these attacks. As depicted in Fig. 2, our framework, *Dynamite*, follows a comprehensive pipeline, starting with data preprocessing to clean, normalize, and encode features. The processed data is used to train both a baseline model and several defense models for robust evaluation. Adversarial samples are then generated using various attack strategies with different intensities to simulate real-world scenarios. Defense models are assessed based on their performance against these adversarial samples, and performance metrics are recorded to label the samples with their most effective defense. Finally, an ML classifier is trained on this labeled data to dynamically predict the most suitable defense model to unseen adversarial attacks. Our experiments on different intrusion datasets demonstrate that *Dynamite* outperforms both random defense and the best static defense, yielding an average F1-score improvement of 76.7% and 65.8%, respectively. Additionally, *Dynamite* significantly enhances computational efficiency, achieving a 96.2% computational time reduction over the Oracle with only a 1.7% F1-score gap. These results underscore the effectiveness of *Dynamite* as a scalable and efficient defense strategy for intrusion detection, successfully balancing high accuracy with reduced computational overhead.

II. RELATED WORK

The growing dependence on computer networks and the expansion of the Internet of Things (IoT) have introduced significant security challenges, driven by the increasing complexity and diversity of these interconnected systems. Intrusion Detection Systems (IDS) are designed to monitor network activity and detect malicious behavior. The integration of machine learning (ML) has enhanced their capability to identify complex and evolving attack patterns with greater accuracy. However, ML-based IDS are vulnerable to adversarial attacks, where carefully crafted input perturbations deceive the models into making incorrect predictions [15]. To enhance resilience against adversarial attacks, various defense strategies have been proposed, which can be broadly categorized into adversarial training [7], [8], [16], modifying the training process [9], [10], and using supplementary networks [11], [17]. Several efforts have been directed toward developing adversarial defense mechanisms specifically tailored to enhance the robustness of ML-IDS against adversarial attacks. Han et al. [18] address traffic-space attacks targeting ML-based NIDS, proposing a defense scheme that reduces evasion rates across multiple attack scenarios. Debicha et al. [12] introduce Adv-Bot, a framework for generating adversarial botnet traffic to test and strengthen IDS defenses. Additionally, Debicha et al. [13] present a transfer learning-based framework that employs multiple adversarial detectors to improve detection rates. Existing studies on ML-based IDS defenses against

adversarial attacks often focus on isolated mechanisms or manual selection, limiting their generalizability.

In contrast, our framework incorporates multiple SOTA defenses and dynamically selects the most effective one based on the performance across the dataset. Rather than requiring extensive manual tuning or being restricted to specific attack types, our approach generalizes to diverse adversarial scenarios by training a classifier to predict the most suitable defense. Once deployed, the algorithm processes each new sample individually and predicts the optimal defense in real-time based on learned patterns. This shift from static to performance-based dynamic defense selection enables our framework to offer robust protection across a broader range of adversarial threats, distinguishing it from existing methods [7]–[11], [16], [17].

III. PROPOSED FRAMEWORK

We propose *Dynamite*, a dynamic defense selection framework designed to strengthen ML-based IDS against adversarial attacks. By addressing the challenges posed by diverse attack types and varying intensities, *Dynamite* provides a robust defense solution. Our framework integrates key components, including adversarial sample generation, defense model training, and dynamic defense selection, forming a comprehensive pipeline to evaluate and mitigate adversarial threats effectively. As shown in Figure 2, the process begins with data preprocessing, where raw data is cleaned, normalized, and encoded. This preprocessed data is then used to train both a baseline DNN model and various defense models. To simulate real-world scenarios, adversarial datasets are generated using multiple attack models, providing a comprehensive benchmark for testing the framework's effectiveness. Next, we evaluate and record the performance of several state-of-the-art defense models to identify the most effective strategies for different adversarial scenarios. To enable dynamic defense selection, an XGBoost-based classifier analyzes patterns in the dataset to predict the most suitable defense model for "new" (unseen) adversarial samples. By integrating optimal defense labels derived from the performance matrix—which evaluates the effectiveness of each defense model against various adversarial attacks—the *Dynamite* dynamically selects the most suitable defense for each attack scenario. Finally, the *Dynamite*'s performance is compared to that of Oracle, the best static defense models, and random defense selection.

A. Data Preprocessing

This module involves data cleaning to remove redundant or irrelevant features, feature standardization to normalize numerical features for consistent scaling, and categorical encoding to convert classification features into numerical representations compatible with ML models. After preprocessing the data, it is divided into training and test sets. The training set is used for baseline model and defense model training, while the test set is reserved for adversarial attack generation and final evaluation.

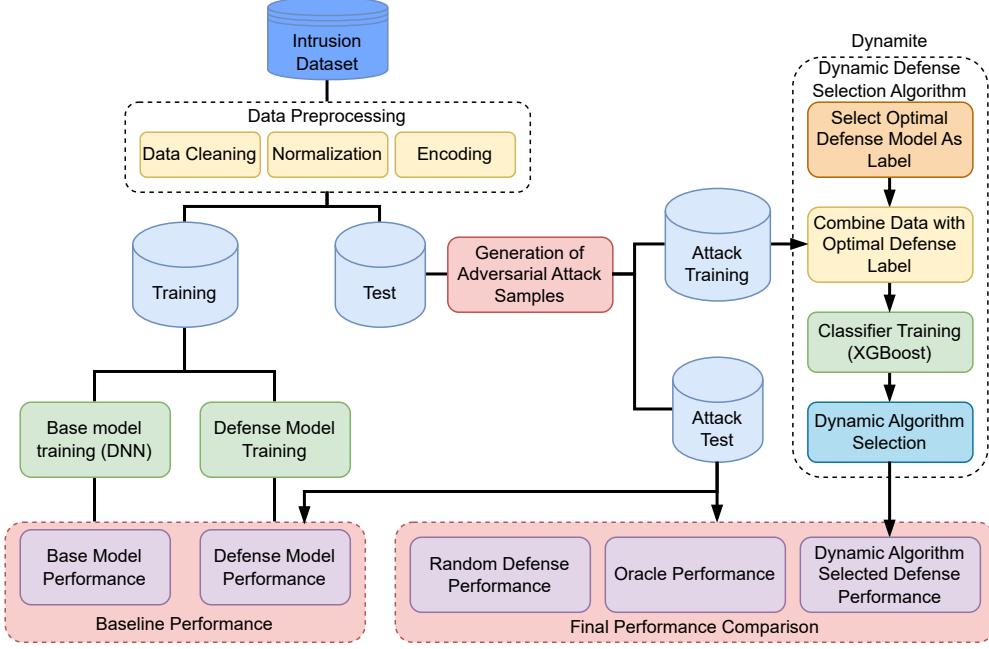


Fig. 2. *Dynamite* is a dynamic defense selection framework designed to enhance ML-based intrusion detection against adversarial attacks. It integrates baseline model training, adversarial sample generation, defense model training, and dynamic defense assignment to effectively address a wide range of attack scenarios.

B. Generation of Adversarial Attack Samples

1) *Selected Adversarial Attacks*: We employ six widely used adversarial attacks: BIM [19], FGSM [20], PGD [7], DF [21], AutoPGD [22], and ZOO [23]. These attacks use gradient-based and query-based methods to generate adversarial samples, introducing input perturbations to manipulate model predictions. The perturbation amount, controlled by the epsilon (ϵ) value, determines the intensity of the attacks, ranging from subtle to more pronounced alterations. This setup enables comprehensive testing under diverse and realistic conditions, offering valuable insights into the framework's performance against various adversarial scenarios.

2) *Adversarial Dataset Generation*: The generation process involves applying each attack model to the dataset, with epsilon (ϵ) values adjusted to simulate varying levels of adversarial intensity. A unique adversarial dataset is generated for each combination of six attack methods (BIM, FGSM, PGD, DF, AutoPGD, and ZOO) and four epsilon values (0.01, 0.1, 0.2, 0.3), resulting in a total of 24 distinct datasets. Each attack is applied to the test dataset, maintaining the same sample size as the original. This ensures consistent evaluation while introducing adversarial perturbations based on attack type and intensity. After generating adversarial attack samples, we split them into two sets: attack training and attack test. The training portion is used to train our dynamic defense selection model, while the test portion is used for final evaluation.

C. Baseline Model Training

To establish a performance baseline, a Deep Neural Network (DNN) [24] is trained on the original, unperturbed dataset. The model is then evaluated under different adversarial attack configurations, providing a reference for assessing the effectiveness of defense strategies. This baseline serves as a crucial benchmark, illustrating the impact of adversarial attacks on model performance and emphasizing the importance of robust defense mechanisms and dynamic selection approaches.

D. Defense Model Training

We evaluate the effectiveness of nine state-of-the-art defenses against adversarial attacks: Projected Gradient Descent Adversarial Training [7], Interpolated Adversarial Training [25], Tradeoff-inspired Adversarial Defense via Surrogate-loss Minimization (TRADES) [8], Free Adversarial Training [16], Gaussian Augmenter [26], Defensive Distillation [9], Robust Soft Label Adversarial Distillation (RSLAD) [10], Feature Squeezing [11], and Gaussian Noise [17]. These defenses were selected for their diverse mechanisms, including adversarial training, data augmentation, loss optimization, and input preprocessing. This selection ensures a comprehensive and diverse evaluation of defense strategies across multiple methodologies. To address varying defense requirements, we introduce multiple parameter configurations for certain models. For RSLAD, configurations like RSLAD10 and RSLAD100 adjust optimization strength to evaluate robustness tradeoffs. This approach systematically assesses adaptability to different adversarial perturbation levels. Applying these defense meth-

ods to diverse adversarial datasets enables the framework to evaluate model adaptability and performance across attack scenarios. These defenses form the basis of the dynamic selection mechanism, allowing the framework to deploy the most effective strategy for each adversarial sample, ensuring robust performance under varying attack types and intensities.

E. Optimal Defense Identification

1) Constructing Attack Training and Attack Test Data:

The attack training and attack test data are created using a subset of the 24 adversarial datasets—generated using different attack methods and epsilon values from the test set—ensuring a distinction between known and unknown data during model evaluation. Specifically, the datasets with an epsilon value of 0.1 (8 datasets) are used as attack training data, representing the known data. The remaining datasets, with other epsilon values (16 datasets), serve as attack test data, representing the unknown data. This setup allows the framework to assess its ability to generalize beyond the perturbation strengths encountered during training.

2) Optimal Defense Selection:

To assess the defense models, we process attack training data through all nine defenses and record key metrics, such as the macro F1-score. This generates a performance matrix, where each entry represents a defense model's effectiveness against a specific adversarial dataset. The matrix serves as a basis for comparing defenses and identifying best strategies, offering insights into how each model addresses adversarial perturbations. To determine the most effective defense for each adversarial sample, we analyze the performance metrics of all nine defense models and select the highest-performing defense for each sample. This selected model is then used as the label, which forms the ground truth for training our dynamic defense selection mechanism.

F. Dynamic Defense Selection Algorithm

Our dynamic defense selection algorithm is designed to adaptively assign the most suitable defense model to each adversarial sample, helping to maintain strong performance across varying attack conditions. To achieve this, we utilize XGBoost (XGB) [27] as a classifier. During training, we combine attack training data with their corresponding optimal defense labels and feed them into XGB. This enables the model to learn the relationships between network features and the most effective defense models. Through this process, XGB identifies key features and establishes a mapping between attack patterns and defense strategies. In the classification phase, XGB can efficiently process the attack test data and assign the most appropriate defense model to each adversarial sample. Furthermore, it generalizes to new adversarial data, dynamically adapting to different attack types and intensities in real-time. This dynamic adaptation mechanism eliminates the reliance on static defense, allowing the framework to adjust its defense strategies based on the unique characteristics of each attack. By dynamically selecting defenses, the framework enhances its resilience against unseen adversarial threats, en-

suring robust and consistent performance across diverse attack scenarios.

G. Final Performance Comparison

During the final evaluation phase, each sample is assigned to a corresponding defense model (e.g., TRADES, RSLAD), ensuring that the selected strategy effectively mitigates the adversarial attack. To comprehensively assess *Dynamite*'s effectiveness, we compute the Macro F1-Score for each selected defense model, offering a holistic measure of overall performance. This final metric is then compared with other baseline approaches, such as Oracle, random defense, and the best static defense, highlighting *Dynamite*'s robustness and adaptability across diverse adversarial scenarios.

IV. EXPERIMENTAL ANALYSIS

A. Baselines

No Defense: This baseline evaluates the performance of a standard DNN model under adversarial attacks without defenses, establishing the lower performance bound.

Random Defense: The random defense performance is assessed by randomly selecting a defense model for each adversarial dataset 100 times. The average performance is then computed, providing a benchmark for evaluating a non-deterministic, uninformed defense selection strategy.

Best Static Defense: The best static defense evaluates defense models on attack training data to identify the most effective defense model. The model with the highest average performance across all adversarial attack types is selected and tested on the attack test data to assess its effectiveness.

Oracle Defense: The Oracle represents the theoretical upper bound of defense performance, derived by selecting the best-performing defense for each of the 24 adversarial datasets across all defense models. Comparing our framework's performance to the Oracle shows how closely the dynamic defense mechanism approximates this ideal.

B. Selected Datasets

WUSTL-IIoT [28]: The WUSTL-IIoT dataset, designed for cybersecurity research in Industrial Internet of Things (IIoT) environments, replicates real-world industrial operations for realistic cyber-attack simulations. It includes network traffic data from IIoT testbeds across various attack scenarios, with 41 features and 1M samples, supporting the development of ML-driven security solutions for industrial settings.

UNSW-NB15 [29]: The UNSW-NB15 dataset is designed for network intrusion detection, combining real-world normal network activities with synthetically generated attack behaviors. It includes nine attack types and features extracted using both traditional and novel techniques. With 43 features and 278K samples, it serves as a key benchmark for developing and evaluating intrusion detection systems.

TABLE I
FINAL PERFORMANCE (MACRO F1-SCORE) COMPARISON

(%)	UNSW-NB15					WUSTL-IIoT				
	No Defense	Dynamite	Oracle	Random	Best-Static [7]	No Defense	Dynamite	Oracle	Random	Best-Static [11]
BIM	30.78	81.17	81.78	64.64	68.73	28.44	77.77	87.16	53.06	72.08
FGSM	43.20	81.22	83.23	66.64	81.99	29.51	71.75	77.73	54.28	59.57
PGD	30.78	81.17	81.86	64.64	68.73	28.44	77.78	87.16	53.06	72.08
DF	10.81	50.64	55.19	20.56	39.69	1.99	25.37	27.91	6.15	5.61
AutoPGD	29.52	80.15	82.39	65.97	71.03	26.03	56.52	76.83	50.91	58.20
ZOO	83.49	90.57	90.77	87.15	87.36	79.24	91.18	94.61	81.86	83.59
Average	38.10	77.49	79.20	61.60	69.59	32.28	66.73	75.23	49.89	58.52

C. Experimental Setup

Hardware: We conduct our experiments on a Linux virtual machine server equipped with a 16-core CPU, 32 GB of RAM, and an NVIDIA A100 GPU with 80 GB of memory.

Evaluation Metric: We select the Macro F1 score as our evaluation metric because it offers a balanced assessment of model performance across all classes, independent of class distribution. This metric is especially pertinent for datasets with imbalanced attack types, as it ensures that minority classes are appropriately represented in the evaluation.

Dynamite Performance Scoring: The final defense performance is evaluated using a weighted scoring formula, which combines the number of samples handled by each defense model and its performance:

$$\text{Score} = \sum_{i=1}^N \left(\frac{\text{Sample Count}_i}{\text{Total Samples}} \times \text{Model Performance}_i \right) \quad (1)$$

where Sample Count_i represents the number of adversarial samples assigned to the i -th defense model, and $\text{Model Performance}_i$ denotes the Macro F1 score of the i -th defense model. This formula calculates a weighted average of the defense models' performances, with the weight determined by the proportion of samples managed by each model. By doing so, it provides a holistic view of how well the dynamic algorithm selection performs across all assigned samples. A higher score reflects both the framework's ability to assign the most suitable defense models and the overall effectiveness of those models in mitigating adversarial impacts.

V. RESULTS

A. Dynamite Defense Performance

Table I provides a comparative analysis of *Dynamite* against selected baselines (no defense, oracle, random, and best static), emphasizing key performance variations across different attacks and datasets. The table reports the macro F1 score for each adversarial attack, as well as the average scores across all attack scenarios. It is evident that *Dynamite* substantially outperforms the best static, random, and no defense baselines, achieving performance nearly equivalent to that of the Oracle. These results underscore *Dynamite*'s capability in improving model robustness while reducing adversarial impact.

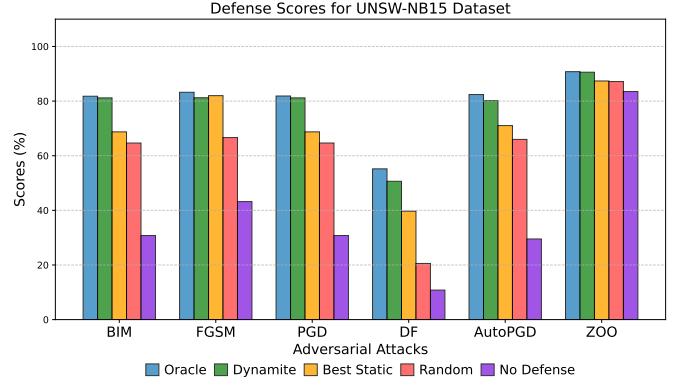


Fig. 3. Prediction Performance Comparison (UNSW-NB15)

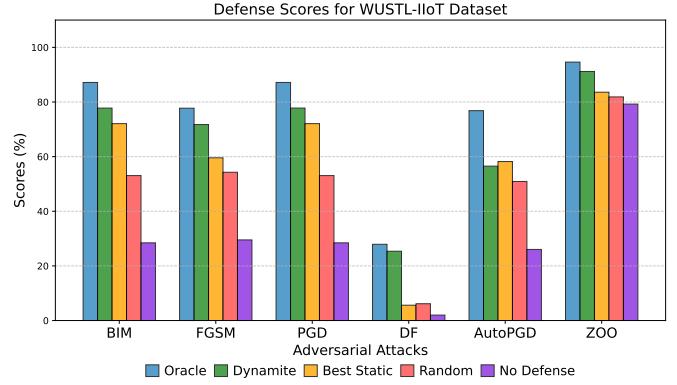


Fig. 4. Prediction Performance Comparison (WUSTL-IIoT)

1) Comparison with Random and Best Static Defenses:

Dynamite exhibits the most significant improvement in performance for the DF attack among all considered adversarial attacks. *Dynamite* achieves substantial improvements over both random and best static defenses, with performance gains of 146.3% and 27.6% on the UNSW-NB15 dataset, and 312.5% and 352.2% on the WUSTL-IIoT dataset, respectively, for the DF attack. This demonstrates that, especially in the case of stronger attacks, *Dynamite* substantially outperforms random and best static, highlighting its effectiveness and reliability in optimizing defense model allocation. As shown in Fig. 3 and Fig. 4, static defenses are more vulnerable to white-box attacks, as attackers can exploit their weaknesses, whereas

TABLE II
Dynamite F1-SCORE IMPROVEMENT RATE

Improvement Rate (%)	UNSW-NB15		WUSTL-IIoT	
	Random	Best-Static	Random	Best-Static
Max	146.3	27.6	312.5	352.2
Average	40.8	13.2	76.7	65.8

in black-box attacks like ZOO, static defenses maintain relatively stable performance. However, *Dynamite* still achieves a notable performance gain, improving up to 14% over the best static defense. Table II presents both the maximum and average performance improvements of *Dynamite* compared to the random and best static defenses. Our method also shows average improvements over both baselines, highlighting its effectiveness and adaptability in dynamically assigning optimal defense strategies.

2) *Comparison with Oracle*: *Dynamite* demonstrates a remarkably small performance gap with the Oracle, underscoring its effectiveness in dynamically selecting defenses across diverse adversarial conditions. In the UNSW dataset, *Dynamite* achieves 90.57% on the ZOO attack, with only a 0.2% gap from the Oracle's 90.77% in a less challenging scenario. Even in the most challenging case, the DF attack, the gap remains limited to just 4.55%. Similarly, in the WUSTL dataset, *Dynamite* reduces the gap to only 2.54% for the DF attack. These results highlight *Dynamite*'s ability to allocate defenses effectively without requiring exhaustive model evaluations, making it both efficient and overhead. On average, the performance difference remains minimal, with a 1.71% gap in UNSW and 8.5% in WUSTL, reinforcing *Dynamite*'s adaptability even in more complex adversarial scenarios.

Insights: The results reflect *Dynamite*'s ability to approach the theoretical upper bound established by the Oracle while demonstrating its flexibility in handling diverse adversarial scenarios. Since the Oracle determines the best performance by testing every dataset across all models, achieving this level of optimality in practice would require significant effort and resources, making it impractical for real-world applications. In contrast, the *Dynamite* dynamically allocates defense strategies using a machine learning-based approach. This method eliminates the need for manually selecting the optimal defense for each attack, showcasing the *Dynamite*'s adaptability in addressing complex adversarial scenarios. Moreover, the *Dynamite* offers enhanced scalability in practical applications, achieving performance levels close to the Oracle without requiring explicit identification of each attack type.

B. Overhead Analysis

The overhead analysis examines the computational efficiency of *Dynamite* by comparing the time required for defense selection per sample with that of the Oracle and Best-static. Table III presents the processing time in ms/sample, highlighting the substantial difference between the two algorithms. For the UNSW-NB15 dataset, the Oracle requires 22.08 ms/sample, whereas the *Dynamite* reduces this

TABLE III
PROCESSING TIME PER SAMPLE COMPARISON

ms/Sample	UNSW-NB15	WUSTL-IIoT
Dynamite	0.8396	0.6670
Oracle	22.077	24.276
Best-static	2.198	2.211

to 0.84 ms/sample, achieving a time reduction of 96.2%. Similarly, in the WUSTL-IIoT dataset, the Oracle requires 24.28 ms/sample, while the *Dynamite* reduces this to 0.67 ms/sample, resulting in a 97.3% computational time reduction. These results demonstrate that *Dynamite* substantially reduces computational overhead in comparison to the Oracle. When evaluated alongside Best-static, which requires 2.20 ms/sample in UNSW-NB15 and 2.21 ms/sample in WUSTL-IIoT, *Dynamite* further reduces processing time by 61.8% and 69.8%, respectively. While Best-static is more efficient than the Oracle, it still lacks the adaptability of *Dynamite*. This contrast underscores *Dynamite*'s efficiency in accelerating defense selection while maintaining strong performance, making it a viable adversarial defense solution.

VI. CONCLUSION

Ensuring robust cybersecurity in machine learning-based intrusion detection remains a critical challenge due to its susceptibility to adversarial attacks. Although various defense mechanisms have been proposed for resilient ML-IDS, a systematic methodology for selecting the most effective defense tailored to specific adversarial attacks is still lacking. This paper proposes a dynamic framework that overcomes the limitations of static defenses by integrating multiple defense models and selecting the most effective one for each attack scenario. Unlike traditional approaches that rely on fixed or manually chosen defenses, *Dynamite* continuously adapts to evolving threats, achieving superior performance over both random selection and the best static defense. *Dynamite* also reduces computational overhead by 96.2% compared to the Oracle, significantly decreasing computational time while maintaining strong defensive capabilities, with only a 1.7% average F1-score loss.

ACKNOWLEDGMENTS

This work has been funded in part by NSF, with award numbers #1826967, #1911095, #2003279, #2052809, #2100237, #2112167, #2112665, and in part by PRISM and CoCoSys, centers in JUMP 2.0, an SRC program sponsored by DARPA.

REFERENCES

- [1] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. De Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.
- [2] O. I. Abiodun, E. O. Abiodun, M. Alawida, R. S. Alkhawaldeh, and H. Arshad, "A review on the security of the internet of things: Challenges and solutions," *Wireless Personal Communications*, vol. 119, pp. 2603–2637, 2021.

- [3] K. A. Da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019.
- [4] O. Gungor, E. Li, Z. Shang, Y. Guo, J. Chen, J. Davis, and T. Rosing, "Rigorous evaluation of machine learning-based intrusion detection against adversarial attacks," in *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2024, pp. 152–158.
- [5] N. Mishra and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," *IEEE Access*, vol. 9, pp. 59 353–59 377, 2021.
- [6] A. Alotaibi and M. A. Rassam, "Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense," *Future Internet*, vol. 15, no. 2, p. 62, 2023.
- [7] A. Madry, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.
- [8] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*. PMLR, 2019, pp. 7472–7482.
- [9] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.
- [10] B. Zi, S. Zhao, X. Ma, and Y.-G. Jiang, "Revisiting adversarial robustness distillation: Robust soft labels make student better," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 443–16 452.
- [11] W. Xu, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.
- [12] I. Debicha, B. Cochez, T. Kenaza, T. Debatty, J.-M. Dricot, and W. Mees, "Adv-bot: Realistic adversarial botnet attacks against network intrusion detection systems," *Computers & Security*, vol. 129, p. 103176, 2023.
- [13] I. Debicha, R. Bauwens, T. Debatty, J.-M. Dricot, T. Kenaza, and W. Mees, "Tad: Transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems," *Future Generation Computer Systems*, vol. 138, pp. 185–197, 2023.
- [14] C. Wang, J. Wang, and Q. Lin, "Adversarial attacks and defenses in deep learning: A survey," in *Intelligent Computing Theories and Application: 17th International Conference, ICIC 2021, Shenzhen, China, August 12–15, 2021, Proceedings, Part I 17*. Springer, 2021, pp. 450–461.
- [15] O. Gungor, T. Rosing, and B. Aksanli, "Roledf: Robust layered defense for intrusion detection against adversarial attacks," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–6.
- [16] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in neural information processing systems*, vol. 32, 2019.
- [17] S. A. Kassam, *Signal detection in non-Gaussian noise*. Springer Science & Business Media, 2012.
- [18] 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.
- [19] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.
- [20] I. J. Goodfellow, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
- [21] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.
- [22] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International conference on machine learning*. PMLR, 2020, pp. 2206–2216.
- [23] 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.
- [24] M. Al-Hawawreh, E. Sitnikova, and N. Abutorab, "X-iiotid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3962–3977, 2022.
- [25] A. Lamb, V. Verma, J. Kannala, and Y. Bengio, "Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 95–103.
- [26] V. Zantedeschi, M.-I. Nicolae, and A. Rawat, "Efficient defenses against adversarial attacks," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 39–49.
- [27] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [28] M. Zolanvari *et al.*, "Wustl-iiot-2021 dataset for iiot cybersecurity research," *Washington University in St. Louis, USA*, 2021.
- [29] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.