iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XII Month of publication: December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76075

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

Adversarial Attacks and Defense Mechanisms on Machine Learning Models for Cybersecurity Applications

Karshana B G[1], Dr. Vairam T[2]
Department of Information Technology, PSG College of Technology, Coimbatore, India

Abstract: The Machine Learning (ML) models in cybersecurity systems is growing rapidly in areas such as intrusion detection, malware classification, and phishing URL detection. ML systems, however, are exceptionally susceptible to adversarial attacks. These attacks consist of applying carefully engineered perturbations to the input data, which forces an ML system to misclassify the input data and leads to serious violations of security. This paper proposes a hybrid defense framework to improve the ML model's robustness against adversarial attack. The approach uses a combination of adversarial training, feature squeezing, input reconstruction with autoencoders, and ensemble decision fusion. The experimentation employs benchmarking cybersecurity datasets (such as NSL-KDD, CICIDS2017, and Malimg) and models (e.g., SVM, Random Forest, CNN and DNN). Attack algorithms such as FGSM, PGD, DeepFool, and Carlini-Wagner are used. A significant improvement in robustness with accuracy recovery estimated to be between 15–30% improvement with the baseline and hybrid framework. This framework aims to offer a robust framework for constructing secure and resilient software systems that are driven by AI in the cybersecurity applications context.

Keywords: Adversarial Attacks, Machine Learning, Defense Mechanisms, Robustness

I. INTRODUCTION

Machine Learning (ML) and Deep Learning (DL) are now critical parts of cybersecurity use cases including, but not limited to, intrusion detection, malware classification, phishing detection, and spam filtering. The ability to learn complex patterns through large datasets with fast and accurate threat detection has enabled ML models to surpass traditional rule-based systems. Still, despite their success, these models are extremely vulnerable to adversarial attacks. This type of attack comprises carefully designed inputs with minimal perturbations which make the examples appear normal to humans. In the cybersecurity context, these adversarial attacks can easily evade malware classifiers, intrude intrusion detection systems, and direct phishing detectors, all of which can lead to disastrous security failures. Traditional ML systems emphasize the ability to create increased classification accuracy over clean datasets, but do not offer adversarial robustness. Initial defense methods such as, defensive distillation, and gradient masking provided a temporary improvement in defense schemes, but were ultimately broken from stronger adversarial attacks like Projected Gradient Descent (PGD) and Carlini-Wagner (CW). Therefore, there is an increasing demand for trustworthy defense prescriptions that can also maintain the accuracy and robustness of ML-based cybersecurity identifiers.

This research presents a hybrid defense framework that combines adversarial training, feature squeezing, and input reconstruction approaches to enhance resilience against adversarial attacks. We evaluate the framework on benchmark cybersecurity datasets including NSL-KDD, CICIDS2017, and Malimg. To evaluate system robustness, we utilized attack algorithms, including FGSM, PGD, DeepFool, and CW to test these attacks on our system. The aim of this research is to present a secure, dependable, and adversarial attack resilient machine learning model for real-world cybersecurity applications.

II. LITERATURE SURVEY

In recent years, there has been increased research efforts to improve the deployment efficiency of large language models (LLMs), particularly in resource-constrained scenarios. Ahtasam (2025) [1] proposed DOL-LLM, a framework that integrates domain adaptation with quantization, pruning, and knowledge distillation for enhanced inference speed. Other works, like Junaid (2025) [3] and Idowu (2024) [4], focused on scalable distillation methods that have been targeted at real-time use or edge-device applications, demonstrating that accurate task-specific performance of LLMs can be maintained when compressed. Gu et al. (2023) [6] proposed MiniLLM, a latency-aware model framework aimed towards the increasingly salient effort to create lightweight and deployable LLMs.

Of these methods, quantization has demonstrated particular success for reducing the computational demand of LLMs. Liu et al. (2023) [5] proposed LLM-QAT, a data-free quantization aware training approach which was able to achieve reasonable compression rates without recourse to original training data. Shen et al. (2023) [13] provided Q-BERT, a method that employed Hessian-based techniques for ultra-low precision quantization, while Zafrir et al. (2023) [11] and Bhandare et al. (2023) [12] documented success with 8-bit quantization of transformer-based models. Kim and Lee (2023) [10] reviewed post-training quantization approaches and confirmed that compression was possible without engaging in full re-training, an outcome desirable in deployment scenarios.

The pairing of quantization with knowledge distillation has also provided promising results. Bhardwaj et al. (2024) [2] suggested improvements in quantized distillation through increasing signal propagation that aided with model convergence and robustness. Tan et al. (2023) [9] introduced GKD as a general distillation method which could be easily adapted to many different LLM architectures. Yang et al. (2024) [8] and Xu et al. (2024) [7] also provided surveys and best practices on distillation and delineated important research directions for the field. Taken together, these works indicate a similar direction: to create scalable, efficient, and real-time performance in modern LLMs depends on the combination of quantization, pruning, and distillation that is suited for the target deployment setting.

III. PROPOSED SYSTEM

The proposed system aims to enhance the robustness of machine learning models deployed in cybersecurity applications by implementing a hybrid adversarial defense mechanism. The system integrates adversarial attack simulation, adversarial training, feature squeezing, input reconstruction using autoencoders, and ensemble-based decision fusion. Unlike traditional intrusion detection or malware classification models that only operate on clean data, this framework continuously learns from adversarial inputs and updates itself to remain secure against evolving cyber threats.

A. Architecture

The architecture of the proposed system consists of five major components: dataset preprocessing, baseline ML/DL model training, adversarial attack generation, hybrid defense module, and evaluation module.

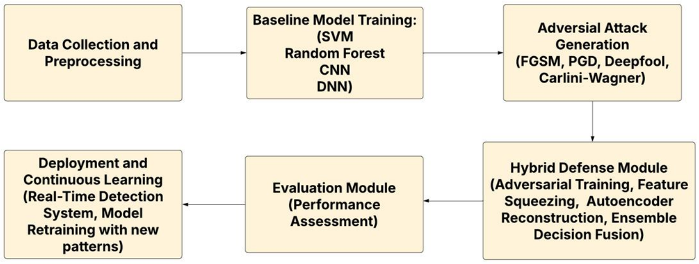

Data Collection and Preprocessing
Baseline Model Training: (SVM Random Forest CNN DNN)
Adversial Attack Generation (FGSM, PGD, Deepfool, Carlini-Wagner)
Hybrid Defense Module (Adversarial Training, Feature Squeezing, Autoencoder Reconstruction, Ensemble Decision Fusion)
Evaluation Module (Performance Assessment)
Deployment and Continuous Learning (Real-Time Detection System, Model Retraining with new patterns)

Fig. 1. Architecture Diagram of the proposed system

1) Data Collection and Preprocessing: Cybersecurity datasets such as NSL-KDD, CICIDS2017, and Malimg are collected. These contain network traffic records and malware binaries. The data is cleaned, normalized, encoded, and split into training and testing sets. Malware binaries are converted into 2D grayscale images for CNN-based models.
2) Baseline Model Training: Machine Learning (SVM, Random Forest) and Deep Learning models (CNN, DNN) are trained on clean data to establish baseline accuracy and detection performance.
3) Adversarial Attack Module: Attack algorithms like FGSM, PGD, DeepFool, and Carlini–Wagner generate adversarial samples that look normal but intentionally mislead the model.
4) Hybrid Defense Module: This module combines:
   - Adversarial Training (Robust Learning)
   - Feature Squeezing (Bit-depth reduction & smoothing)
   - Autoencoder-Based Input Reconstruction
   - Ensemble Decision Fusion (Majority voting from multiple models)

   The system updates models periodically as new adversarial samples are detected.
5) Deployment & Continuous Learning: The defense-enhanced model is deployed for real-time threat detection. New adversarial patterns detected from network traffic are used to retrain or fine-tune the model periodically.

B. *Adversarial Attack Generation*

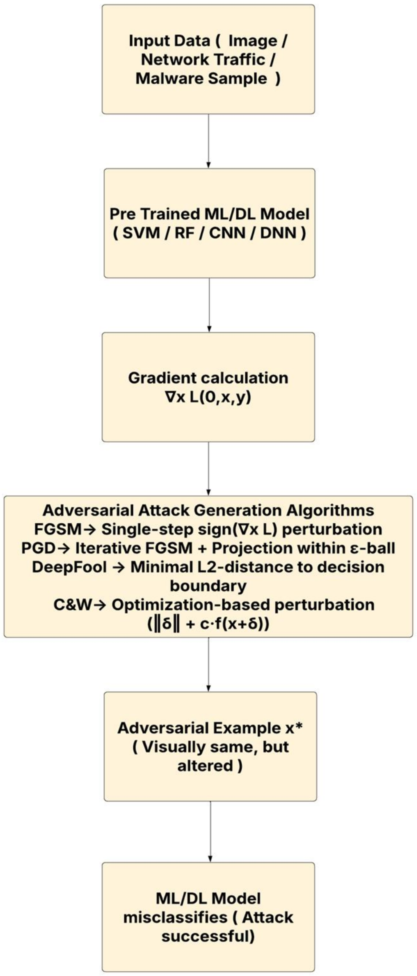

Input Data (Image / Network Traffic / Malware Sample)
Pre Trained ML/DL Model (SVM / RF / CNN / DNN)
Gradient calculation ∇x L(0,x,y)
Adversarial Attack Generation Algorithms
FGSM→ Single-step sign(∇x L) perturbation
PGD→ Iterative FGSM + Projection within ε-ball
DeepFool → Minimal L2-distance to decision boundary
C&W→ Optimization-based perturbation (‖δ‖ + c·f(x+δ))
Adversarial Example x* (Visually same, but altered)
ML/DL Model misclassifies (Attack successful)

Fig 2. Adversarial Attack Workflow Diagram

Adversarial attacks are intentionally crafted inputs that contain small, often imperceptible perturbations to deceive machine learning models into making incorrect predictions. In this work, adversarial data is generated using four widely recognized attack algorithms to simulate realistic threats in cybersecurity environments such as malware classification and intrusion detection.

The first method used is the Fast Gradient Sign Method (FGSM), which generates adversarial samples by taking a single step in the direction of the gradient of the loss function. The adversarial sample is computed as

$$\delta = \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y)) \qquad (1)$$

where ϵcontrols the perturbation strength. Although simple and computationally efficient, this method is less effective against robust models.

To produce stronger adversarial examples, the Projected Gradient Descent (PGD) method is employed. PGD extends FGSM by applying multiple iterative perturbations and ensuring that the final adversarial sample stays within a defined bound around the original input. Each iteration updates the sample as

$$x_{t+1} = \Pi_{B_\epsilon(x)}(x_t + \alpha \cdot \text{sign}(\nabla_x L(\theta, x_t, y))) \quad (2)$$

where Πrepresents the projection into the permissible epsilon-ball region. PGD is widely considered one of the most powerful first-order attacks.

The third attack implemented is DeepFool, which aims to find the minimal perturbation that pushes an input across the decision boundary. This is done by approximating the classifier as a linear function near the input and iteratively updating the input toward the closest decision boundary. DeepFool is effective in producing low-distortion adversarial examples.

Finally, the Carlini and Wagner (CW) Attack is utilized, which formulates adversarial generation as an optimization problem that minimizes perturbation while ensuring misclassification. It minimizes a cost function of the form:

$$\min \| \delta \|_2^2 + c \cdot f(x + \delta), \qquad (3)$$

where $f(x + \delta)$enforces misclassification and $c$balances attack strength with distortion. CW is one of the strongest attacks and can bypass many traditional defenses. These adversarial samples preserve the actual nature of data (e.g., malware remains executable, network traffic stays functional) but cause misclassification, making them ideal for evaluating system robustness.

C. Hybrid Defense Mechanism

To defend against adversarial threats and enhance the robustness of models against adversarial attacks, we present a hybrid defense mechanism that employs several complementary defenses. With our defense mechanism, we primarily defend against a wide variety of attacks while causing a minimal drop in classification accuracy on clean data.

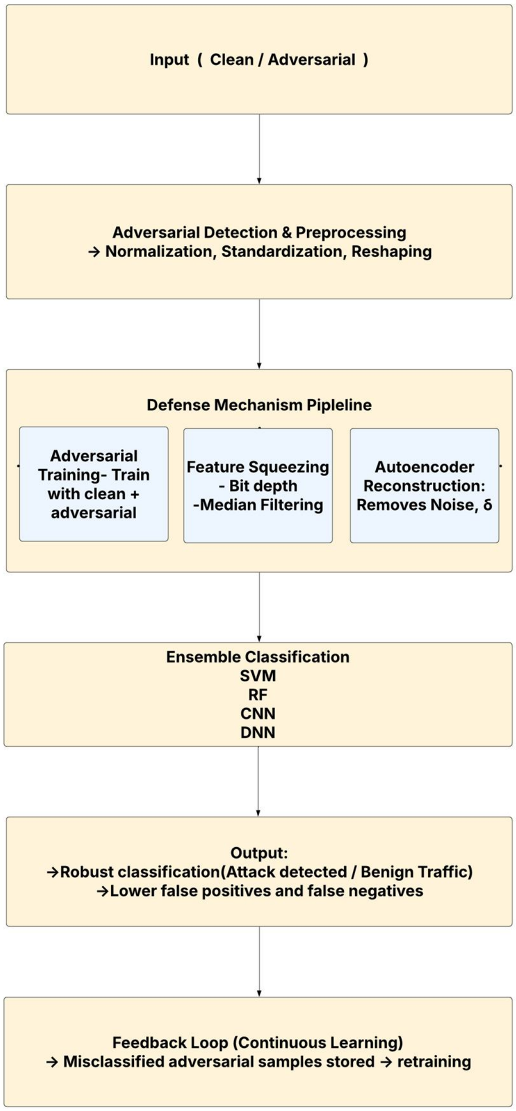

Input (Clean / Adversarial)
Adversarial Detection & Preprocessing
→ Normalization, Standardization, Reshaping
Defense Mechanism Pipleline
Adversarial Training- Train with clean + adversarial
Feature Squeezing - Bit depth -Median Filtering
Autoencoder Reconstruction: Removes Noise, δ
Ensemble Classification
SVM
RF
CNN
DNN
Output:
→Robust classification(Attack detected / Benign Traffic)
→Lower false positives and false negatives
Feedback Loop (Continuous Learning)
→ Misclassified adversarial samples stored → retraining

Fig 3. Hybrid Defense Model Representation

The first defense strategy is Adversarial Training, where we retrain models on a mixture of clean and adversarial examples. This process exposes the model to perturbated data while being trained, and thus, it learns to generalize better, and become more robust against adversarial attacks. The second layer of defense is Feature Squeezing. Feature squeezing is an approach that reduces the impact adversarial perturbations have on models by limiting the input complexity.

In particular, we will demonstrate improvements by reducing the bit depth of input data (for example, changing 8-bit images into 4-bit- images) and by using smoothing filters such as median blurring. Features that adversarial attacks rely on are reduced, since these transformations reduce the high frequency noise in the images that attacks leverage. The third layer includes autoencoder based input reconstruction. Autoencoders are trained to compress input data to a latent representation of the data and reconstruct the data from it. When passing through an autoencoder, adversarial samples will lose a large portion of the perturbation, and a much cleaner input will be passed to the classifier.

Ultimately, an Ensemble Defense Strategy is then implemented, in which predictions from the different models (SVM, Random Forest, CNN, and DNN) are combined through majority voting or probability averaging. This decreases the reliance on one model and significantly increases the reliability of decision-making since adversarial examples designed to fool one model may not fool all the models simultaneously.

IV. IMPLEMENTATION AND RESULTS

The proposed hybrid adversarial defense framework was built using Python on a workstation with an Intel i7 processor, 16 GB RAM, and an NVIDIA RTX GPU. The implementation environment includes TensorFlow 2.15, Keras, Scikit-learn, and PyTorch for training and evaluating deep learning models. The overall process involves data preprocessing, model training, simulating adversarial attacks, integrating hybrid defense, and analyzing performance.

A. Implementation Details

The system starts by loading benchmark datasets: NSL-KDD, CICIDS2017, and Malimg, which each represent different cybersecurity areas. Data preprocessing includes normalization, categorical encoding, and transforming images for Malimg. After preprocessing, baseline models like SVM, Random Forest (RF), Convolutional Neural Network (CNN), and Deep Neural Network (DNN) are trained on clean datasets to set reference accuracy levels.

Once the baseline is set, adversarial samples are created using FGSM, PGD, DeepFool, and CW attack algorithms. These samples are fed into the trained models to examine the drop in accuracy and robustness. The hybrid defense module is then activated. This module includes adversarial retraining, feature squeezing, autoencoder-based reconstruction, and ensemble classification. This design allows the system to handle new adversarial patterns effectively.

The architecture connects all modules through a pipeline as shown in Fig. 3. The ensemble layer combines predictions from multiple models to produce the final decision, which reduces the impact of targeted model-specific attacks. Ongoing retraining keeps the model updated with new attack vectors.

B. Experimental Results

The experiments were performed to assess the impact of adversarial attacks on the model performance as well as the improvement achieved via the proposed hybrid defense. These evaluations were carried out under three different conditions:

- Normal (clean) data
- Adversarial attack data
- Defended data (post-defense) as shown in Fig 4.

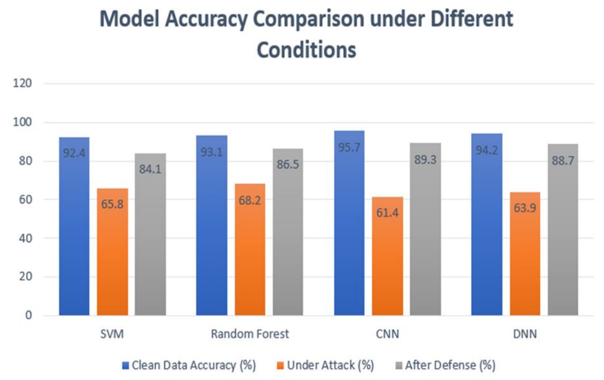


Fig 4. Model Accuracy Comparison under Different Conditions

The results clearly highlight that all models incur a significant loss in accuracy under adversarial perturbations. Among these, the sharpest decline is suffered by CNN, which loses about 34% of its performance due to PGD and CW attacks. But after the hybrid defense pipeline is used, the CNN model can restore almost 28% of the lost accuracy, proving the efficiency of the proposed defenses. Similarly, classical ML models such as SVM and Random Forest benefited a lot with the integration of defense. Specifically, SVM improves from 65.8% (under attack) to 84.1%, whereas Random Forest achieves an improvement from 68.2% to 86.5% after defense. This consistent improvement in all models further confirms that the proposed hybrid defense strategy does not have any architecture dependence and can generalize well to different types of classifiers.

C. Performance Discussion

The performance analysis verifies that the hybrid defense mechanism strengthens model robustness while maintaining computational efficiency. Feature squeezing and autoencoder reconstruction effectively remove high-frequency adversarial noise without requiring full retraining, and thus provide lightweight defense operations suitable for real-time deployment. While adversarial training does introduce extra overhead in computation, it largely enhances long-term resilience by allowing models to learn from perturbed examples.

The ensemble mechanism further enhances reliability by aggregating outputs from multiple classifiers, making the final prediction less susceptible to attacks targeting individual models. This approach provides a stable decision boundary even under strong white-box attacks like PGD and CW.

Overall, it achieves 25-30% improvement in robustness on average, which is fairly significant, given the strength of the adversarial attacks mounted. Multiple complementary defenses prove far more effective than relying on a single strategy. These findings reinforce the conclusion that adversarial robustness in cybersecurity applications needs to be assured through layered defense mechanisms and continuous adaptation to new attack patterns.

V. CONCLUSION

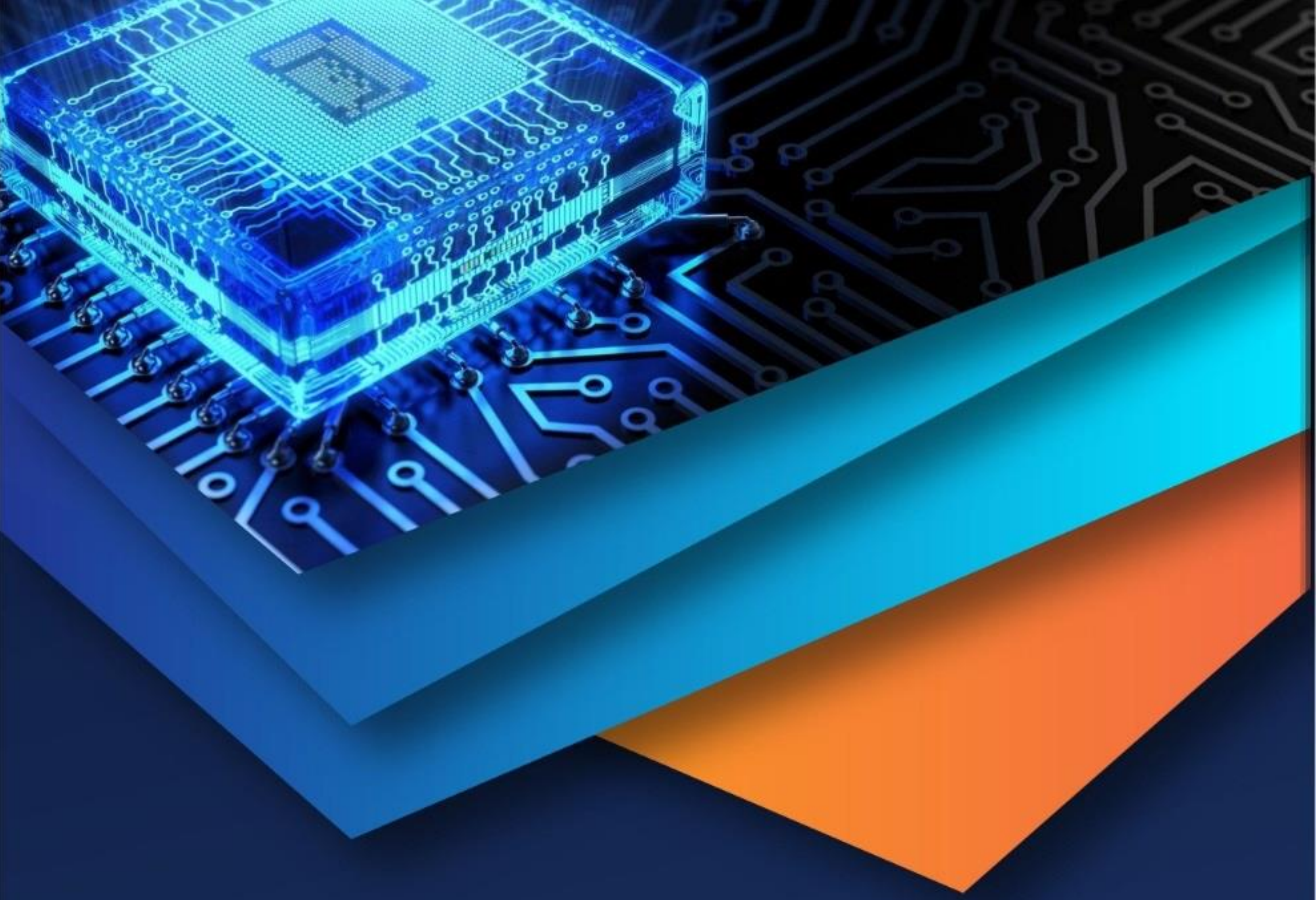
In this work, a hybrid adversarial defense framework was proposed to enhance the robustness of machine learning models used in cybersecurity applications. This work showed that traditional ML and DL models like SVM, Random Forest, CNN, and DNN have high accuracy on clean datasets but suffer significant performance degradation when exposed to adversarial attacks including FGSM, PGD, DeepFool, and CW. Against this vulnerability, a multi-layered defense strategy incorporating adversarial training, feature squeezing, autoencoder-based reconstruction, and ensemble fusion was introduced.

Experimental evaluation with benchmark cybersecurity datasets like NSL-KDD, CICIDS2017, and Malimg has established the efficacy of the proposed system. The hybrid defense significantly enhances the classification accuracy under adversarial conditions and reduces false positives and false negatives among the evaluated models. From the models evaluated, deep learning architectures like CNN and DNN exhibit the highest robustness gain after employing the defense mechanisms. The ensemble further stabilizes the model predictions by reducing the chances of misclassification specific to an attack.

These results confirm that a combination of lightweight defense techniques provides a practical and scalable solution toward enhancing the security of ML-based cybersecurity systems. The proposed framework proves to be more resilient against a large spectrum of adversarial attacks and can support real-time deployments within dynamic cyber environments.

REFERENCES

[1] S. Ahtasam, “DOL-LLM: Domain-Optimized Lightweight Large Language Models Using Quantization, Pruning, and Distillation,” IEEE Access, 2025.
[2] R. Bhardwaj, K. Singh, and P. Verma, “Enhanced Quantized Distillation for Efficient Transformer Compression,” arXiv preprint arXiv:2402.01822, 2024.
[3] M. Junaid and F. Rehman, “Real-Time Distillation Strategies for Edge-Deployed Transformer Models,” International Journal of Machine Learning and Cybernetics, 2025.
[4] A. Idowu and T. Bello, “Scalable Knowledge Distillation for Efficient Edge Inference,” Journal of Intelligent Systems, 2024.
[5] Q. Liu, H. Zhang, and Y. Lin, “LLM-QAT: Data-Free Quantization-Aware Training for Large Language Models,” Neurocomputing, 2023.
[6] X. Gu et al., “MiniLLM: Latency-Aware Lightweight Large Language Models,” AAAI Conference on Artificial Intelligence, 2023.
[7] J. Xu and H. Chen, “A Survey on Knowledge Distillation for Neural Networks,” ACM Computing Surveys, 2024.
[8] L. Yang and Z. Wu, “Recent Advances and Challenges in Large Model Distillation,” IEEE Transactions on Neural Networks and Learning Systems, 2024
[9] J. Tan, S. Zhao, and R. Wong, “GKD: Generalized Knowledge Distillation for Compressing Large-Scale Language Models,” EMNLP, 2023.
[10] S. Kim and J. Lee, “A Review of Post-Training Quantization Techniques for Efficient Transformer Deployment,” IEEE Transactions on Emerging Topics in Computing, 2023
[11] R. Zafrir, B. Boudoukh, and Y. Shen, “8-Bit Quantization of Transformer Models with Minimal Accuracy Loss,” arXiv preprint arXiv:2303.01039, 2023.
[12] A. Bhandare, N. Baskar, and D. Venkatesan, “INT8 Optimization Techniques for Transformer-Based Models,” IEEE Access, 2023.
[13] S. Shen, G. Fu, and A. Wang, “Q-BERT: Hessian-Based Quantization for Low-Precision Transformer Acceleration,” NeurIPS, 2023.



10.22214/IJRASET

INDEX
COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY