[1] Ayoub Akennaf

Evaluation of Defense Strategies Against Adversarial Attacks in Intrusion Detection Systems

Journal of Electrical Systems

Abstract: - Cyber dangers have become more common in recent years, making it even more important to have strong intrusion detection systems (IDS) that can find and stop hostile attempts. The goal of this paper is to repeat and expand on the results of the state of the art of Intrusion Detection Systems and Adversarial Attacks, which looked into how machine learning-based IDS can be hacked in different ways. We test how well different machine learning models work when they are attacked using well-known methods such as the fast gradient sign method, the Carlini and Wagner attacks, and projected gradient descent. Our study uses a wide range of experiments to check how well these attacks work and how strong the models are. In addition, we look into new ways to protect IDS against adversarial changes, such as adversarial training and feature reduction techniques. These will help IDS find threats more quickly. The goal of our repeat study is to add to the current conversation in the fields of hostile machine learning and cybersecurity by giving more information about the arms race between adversarial attacks and defense strategies. In the end, this study aims to give valuable direction that can help in developing IDS that are stronger and better able to protect against new cyber dangers.

Keywords: Cybersecurity, Intrusion detection system, fraud detection, cyber threats, machine learning, adversarial attacks.

I. INTRODUCTION

In the digital era, fast technological advancements have altered the cybersecurity environment, bringing both benefits and difficulties [1]. As enterprises increasingly depend on machine learning (ML) algorithms for diverse applications, such as intrusion detection systems (IDS), the need for stringent security measures has become paramount [2]. IDS are essential for protecting networks by monitoring and analyzing traffic to identify illegal access and possible threats. The use of machine learning into Intrusion Detection Systems has created new vulnerabilities, especially via adversarial assaults [3].

Adversarial machine learning (AML) encompasses methodologies that alter input data to mislead machine learning algorithms, resulting in erroneous predictions or classifications [3]. These assaults provide considerable dangers to the reliability and efficacy of IDS, as they may lead to the misclassification of harmful inputs as benign, so compromising the system's capacity to defend against cyber threats [4]. The increasing complexity of adversarial assaults requires a thorough comprehension of their mechanics, and the formulation of effective response methods [2].

There are difficulties in integrating machine learning with IDS [5]. Although ML techniques may greatly increase the accuracy of detection, they also create new weaknesses that attackers might take advantage of. AML is a rapidly developing field of study that focusses on methods for tricking ML models by manipulating input data [6]. The efficacy of IDS may be jeopardized if hostile inputs are mistakenly classified as benign as a result of these adversarial assaults. Strong defenses against adversarial examples are becoming more and more necessary as fraudsters become better at creating them [7].

Because they may damage the reliability of these systems and put organizations at serious risk, adversarial assaults on intrusion detection systems have far-reaching consequences [8]. Adversarial manipulation includes IDS that fail to identify hostile intrusions, which may lead to data breaches, financial losses, and reputational harm. To keep IDS secure, it is crucial to comprehend the methods used by adversarial assaults and create effective defenses. Our goal is to analyze various classifiers' weaknesses in order to find trends and insights that help guide the creation of more robust intrusion detection systems. To further strengthen IDSs resistance to adversarial perturbations, we will also investigate new defensive mechanisms, such as adversarial training and feature reduction techniques. Key contribution of the proposed study are as follows:

- Analysis of adversarial attacks with machine learning-based IDS.
- Comparison of different machine learning models classifying adversarial vulnerabilities.
- Evaluation of various defense mechanisms, including adversarial training and gradient masking

[1] School of Information Science, Morocco. ayoub.aennaf@gmail.com

- Identification of vulnerabilities in current ML/DL systems and the challenges posed by evolving attack methods.

The remaining section of this work is organized as follows: The second section is devoted to a review of the literature, which includes all investigations conducted on this subject from every angle. In Section 3, we evaluate the detection performance of several machine learning models using a benchmark dataset. The findings of all the models that were chosen are shown in Section 4, and the study's general conclusions are summarized in Section 5.

II. Literature Reviews

The study of AML has attracted a lot of interest lately, especially because of its potential impact on IDS and cybersecurity. This overview of the literature summarizes the main conclusions drawn from previous studies, emphasizing the development of adversarial assaults, how they affect machine learning models, and the defense techniques created to fend them off.

A. Machine learning-based Intrusion Detection System

Intrusion Detection Systems (IDS) that use machine learning (ML) have greatly improved cybersecurity by making it easier to find and stop cyber dangers that are constantly changing and becoming more complicated. IDS that are based on unchanging rules and signature-based monitoring often don't work well when it comes to advanced persistent threats and zero-day vulnerabilities. Support Vector Machines (SVM), Random Forest, Neural Networks, and Deep Learning are some of the methods that machine learning uses to look at network traffic trends and find outliers in real time. One study by [9] says that machine learning methods can help cut down on false positives and boost discovery rates, especially when used in large-scale network settings. Machine learning-based intrusion detection systems are very useful in today's safety world because they can learn from past data and adapt to new dangers [10].

Deep learning has made IDS even more useful. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are two techniques that have been used to look at network traffic data and find complicated attack patterns [11]. These models can take raw data and pull-out high-level traits that can be used to find complex attacks like Distributed Denial of Service (DDoS) and Advanced Persistent Threats (APT). According to a study by [12], ML-based IDS can handle the large amounts of data that are created by network traffic and provide accurate and quick attack detection. As hacks get smarter, machine learning is being added to intrusion detection systems in new and interesting ways. These new ways look like they could help protect network platforms [13].

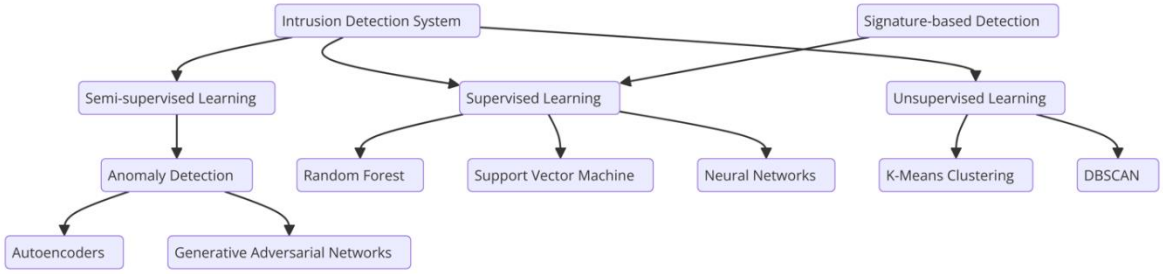

Fig. 1 Machine Learning Models Used in Intrusion Detection Systems

B. Overview of Adversarial Machine Learning

Adversarial Machine Learning (AML) has emerged as a critical area of research within the field of cybersecurity, particularly in the context of machine learning-based security systems [2]. AML focuses on the vulnerabilities of machine learning models when exposed to adversarial inputs designed to deceive or manipulate the model's predictions. In the context of IDS, adversarial attacks can generate malicious network traffic that is specifically crafted to evade detection. [1] were among the first to demonstrate that neural networks are vulnerable to adversarial examples, highlighting a fundamental flaw in the robustness of machine learning models. Since then, research in AML has expanded, uncovering various attack vectors and emphasizing the need for developing more resilient IDS models [3].

The significance of AML lies in its implications for the security of systems that rely on machine learning. According to a survey by [14], adversarial attacks can take various forms, including evasion attacks, where an attacker modifies input data to evade detection, and poisoning attacks, which involve injecting malicious data into the training process. These attacks pose significant challenges for IDS, as they can lead to undetected security breaches and compromise the integrity of network defenses. Consequently, the field of AML not only aims to understand how adversarial attacks work but also seeks to develop robust defense mechanisms. Techniques such as

adversarial training, defensive distillation, and feature squeezing have been proposed to enhance the resilience of machine learning models against adversarial attacks, ensuring more reliable security systems [15].

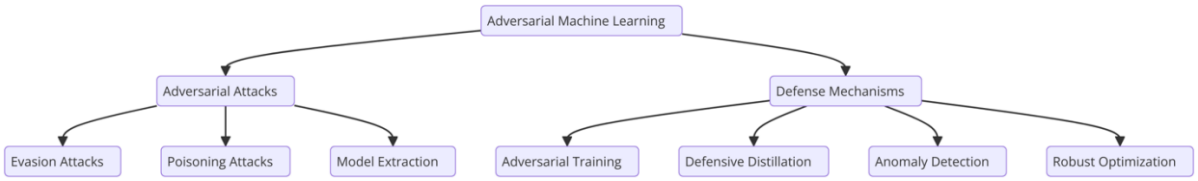

Fig. 2 Adversarial Machine Learning: Attacks and Defense Mechanisms

C. Machine Learning Adversaries against IDS

Machine learning adversaries against IDS exploit the inherent vulnerabilities in machine learning models to evade detection or degrade the performance of the IDS. Evasion attacks, a common adversarial technique, involve modifying malicious inputs to appear benign, thereby bypassing the IDS [15]. Papernot et al. (2016) demonstrated that even small perturbations in input data could cause a well-trained machine learning model to misclassify malicious activity as normal, effectively evading detection. In the context of IDS, attackers often analyze the feature extraction process and decision boundaries of the model to craft inputs that go undetected. This highlights the critical need for IDS models to be not only accurate but also robust against adversarial manipulations [16].

Adversarial technique is poisoning attacks, where an attacker corrupts the training data used by the IDS. By injecting malicious samples into the training set, attackers can manipulate the learning process, causing the IDS to learn incorrect patterns and behaviors. This can lead to a significant increase in false negatives, where actual attacks go undetected. The impact of poisoning attacks on IDS and found that even a small fraction of poisoned data could severely compromise the model's performance. These findings underscore the necessity of incorporating robust data validation and anomaly detection mechanisms in the training phase of IDS. Defenses against these adversarial strategies, such as adversarial training and model hardening, are crucial to maintaining the integrity and effectiveness of machine learning-based IDS [17].

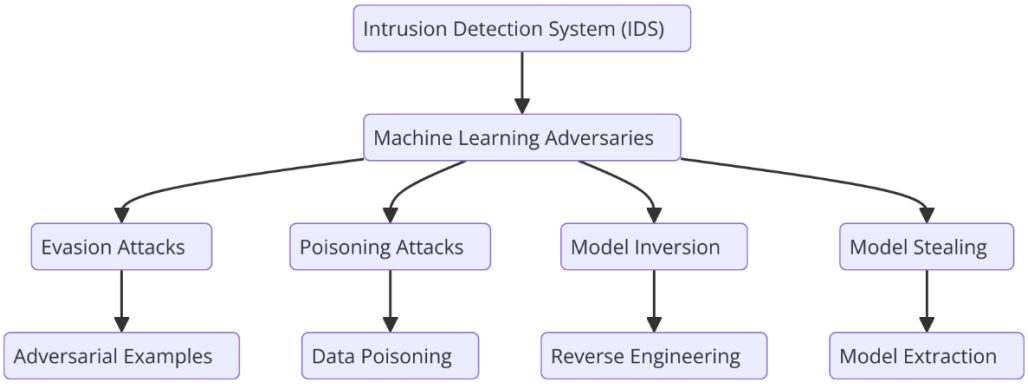

Fig. 3 Machine Learning Adversaries against IDS

D. White-Box attacks against IDS

White-box attacks represent a severe threat to machine learning-based IDS, as they assume the adversary has complete knowledge of the IDS model, including its architecture, parameters, and training data [19]. With this level of access, attackers can generate highly effective adversarial examples tailored to exploit the specific vulnerabilities of the IDS. The Fast Gradient Sign Method (FGSM), a white-box attack technique that uses the gradient of the loss function to create adversarial examples capable of misleading the model. In the context of IDS, white-box attacks can generate network traffic patterns that appear legitimate, successfully evading detection and potentially compromising network security [20].

The implications of white-box attacks on IDS are profound, as they expose the limitations of even the most sophisticated machine learning models. White-box attacks can be more potent than black-box attacks due to the attacker's ability to precisely craft adversarial inputs that target the model's weaknesses. This presents a significant challenge for IDS, as it necessitates the development of advanced defense mechanisms. Current strategies to counter white-box attacks include adversarial training, which involves training the IDS on adversarial examples to improve robustness, and gradient masking, which aims to obscure the model's gradient information to hinder the attacker's ability to craft adversarial inputs. Despite these efforts, white-box attacks continue to push the boundaries of IDS security, prompting ongoing research to develop more resilient defense techniques [21].

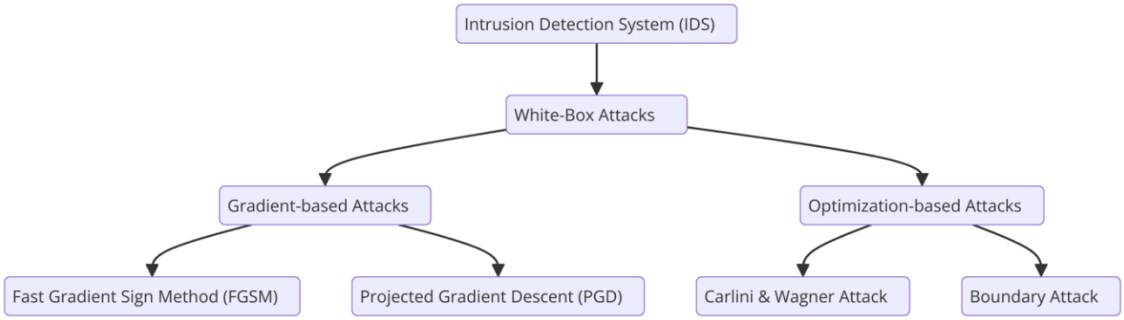

Fig. 4 Machine Learning Adversaries against IDS

E. Benchmark Datasets

Benchmark datasets are fundamental to the research and development of machine learning-based IDS. They provide standardized datasets that enable researchers to train, test, and evaluate the performance of various IDS models. One of the earliest and most widely used datasets is the KDD Cup 99 dataset [21], which contains a wide range of network traffic data labeled as normal or attack. Although it has been criticized for certain limitations, such as redundant records and outdated attack patterns, it paved the way for subsequent datasets. The NSL-KDD dataset, an improved version of KDD Cup 99, was introduced to address these shortcomings, offering a more balanced distribution of training and testing data for evaluating IDS models [22].

In recent years, more comprehensive and up-to-date datasets have been developed to reflect the evolving landscape of cyber threats. The CICIDS2017 dataset [23] and the UNSW-NB15 dataset (Moustafa and Slay, 2015) are examples of modern benchmark datasets that include a diverse range of attack types and realistic network traffic scenarios. These datasets have become essential tools for researchers to validate the effectiveness of ML-based IDS, providing insights into the detection capabilities of various algorithms. By using these benchmark datasets, researchers can perform a rigorous assessment of IDS performance, including metrics such as detection rate, false positive rate, and computational efficiency, ensuring that proposed models are robust and generalizable across different network environments.

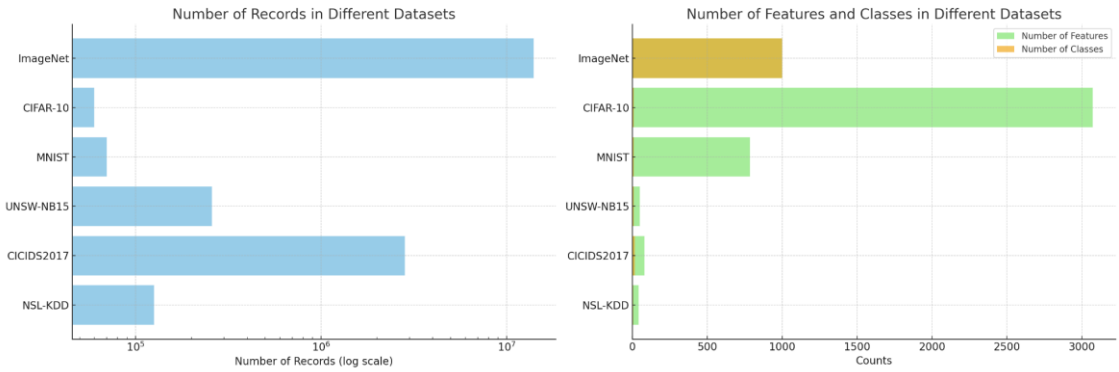

Fig. 5 Comprehensive Comparison of Dataset Size, Feature Count, and Class Diversity

Table I. Benchmark Datasets

Dataset	Features	Classes	Number of Records
NSL-KDD	41	5 (Normal, DoS, R2L, U2R, Probe)	125,973
CICIDS2017	80	15 (e.g., Normal, DoS Hulk, PortScan, DDoS)	2,830,743
UNSW-NB15	49	10 (e.g., Normal, Fuzzers, Analysis, Backdoors)	257,673
MNIST	784	10 (digits 0-9)	70,000
CIFAR-10	3,072	10 (airplane, automobile, bird, etc.)	60,000
ImageNet	Variable	1,000 (wide range of object categories)	Over 14 million

III. METHODOLOGIES

This section provides a detailed description of the methods applied to classify network traffic as normal or anomalous. The study employed both machine learning and deep learning models across four widely used intrusion detection datasets. The methodology includes dataset description, preprocessing techniques, the application of various models, and the evaluation measures used to assess model performance. The experimental setup used for conducting the tests is also outlined.

A. Datasets Description

The study utilized four publicly available datasets: KDD Cup 99, NSL-KDD, UNSW-NB15, and CICIDS2017. These datasets are extensively used in the cybersecurity domain for building and evaluating intrusion detection systems (IDS). KDD Cup 99 is an older dataset that simulates network traffic with both normal and attack events, but it suffers from redundant and irrelevant records, leading to data imbalance. To address these shortcomings, the NSL-KDD dataset was introduced, providing a more balanced version with a more reliable testing environment. UNSW-NB15 is a more recent dataset that contains network traffic data representing modern attack vectors and new protocols, while CICIDS2017 simulates real-world traffic and includes modern intrusion attempts, offering a broader set of attack types. Each dataset provides a variety of features representing network behavior and labels indicating whether the traffic is normal or malicious.

a) Prepossessing

Preprocessing is a crucial step to ensure that the data is prepared for model training and evaluation. The first step in preprocessing involved data cleaning, which included removing duplicate records and eliminating irrelevant features, such as timestamps or unique identifiers that do not contribute to the detection of attacks. Missing data were handled by imputing the mean for numerical features and the mode for categorical ones. For the models to work effectively, feature scaling was applied using Min-Max normalization to bring all the feature values into a common range between 0 and 1. This was particularly important for models like Support Vector Machine (SVM) and neural networks, which are sensitive to the scale of data. Categorical variables, such as protocol type, were encoded using one-hot encoding to convert them into numerical representations. Finally, the datasets were split into training and testing sets using a 70-30 ratio, with stratified sampling applied to ensure a balanced distribution of attack and normal records in both sets.

B. Machine Learning Models

Three machine learning models were employed for intrusion detection: Support Vector Machine (SVM), Random Forest, and Decision Tree. SVM is a powerful classification model that seeks to find the optimal hyperplane separating the normal and attack classes. In this study, the RBF (Radial Basis Function) kernel was chosen for its effectiveness in handling high-dimensional data. Hyperparameter tuning, including adjusting the regularization parameter (C) and kernel coefficient (gamma), was conducted using grid search to optimize the SVM model. Random Forest, an ensemble learning technique, was also employed due to its robustness and ability to reduce overfitting. It operates by constructing multiple decision trees and averaging their predictions. For this model, 100 decision trees were used, and feature importance scores were calculated to determine the most influential features for intrusion detection. Finally, the Decision Tree model was applied as a simple and interpretable classification model, using Gini impurity as the criterion for node splitting. To avoid overfitting, the maximum tree depth was limited.

1. Support Vector Machine

Support Vector Machine (SVM) works by finding the hyperplane that maximizes the margin between two classes. The hyperplane is defined by the equation:

$$\omega \cdot x + b = 0$$

where ω is the weight vector and b is the bias. The goal is to maximize the margin $\frac{2}{\|\omega\|}$ which is equivalent to minimizing $\|\omega\|$.

The optimization problem is formulated as:

$$\min_{\omega,b} \frac{1}{2}\|\omega\|^2 \text{ subject to } y_i(\omega \cdot x_i + b) \geq 1, \forall_i$$

Where $y \in \{-1, +1\}$ represents the class labels.

2. Random Forest

Random Forest is an ensemble of decision trees. Each tree uses random subsets of data and features. The trees are trained independently, and the final output is determined by averaging (for regression) or majority voting (for classification).

Each tree uses Gini impurity to split nodes. Gini impurity is defined as:

$$Gini(t) = 1 - \sum_{i=1}^{c} p_i^2$$

Where p_i is the proportion of samples of class i at node t, and c is the number of classes.

3. *Decision Tree*

A Decision Tree splits data based on maximizing information gain, calculated as:

$$Gain(S, A) = Entropy(S) - \sum_{\vartheta \in Values(A)} \frac{|S_\vartheta|}{|S|} Entropy(S_\vartheta)$$

where S is the dataset, A is the feature, and S_ϑ is the subset of S for a particular value ϑ.
The entropy is defined as:

$$Entropy(S) = -\sum_{i=1}^{C} p_i \, log_2(p_i)$$

Where p_i is the probability of class i.

C. Deep Learning Models

In addition to traditional machine learning models, deep learning models were applied to further enhance the classification performance. Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models were used. CNN, although typically applied to image data, was adapted to analyze network traffic by treating the data as a 2D matrix of features. This allowed the model to automatically learn intricate patterns in the data. A 1D CNN architecture was used, consisting of several convolutional layers followed by pooling and fully connected layers. RNN was employed to capture temporal dependencies in network traffic, as it is designed for sequential data. LSTM, a more advanced version of RNN, was also used to model long-term dependencies by mitigating the vanishing gradient problem. Similarly, GRU, a simpler version of LSTM, was used for faster training while retaining the ability to capture complex temporal relationships in network traffic. These deep learning models were trained using the Adam optimizer and included dropout layers to prevent overfitting.

1. *CNN*

CNNs apply convolutional filters to input data. A convolution operation is defined as:

$$O_{i,j} = \sum_{m,n} X_{i+m,j+n} \cdot f_{m,n}$$

Where X is the input, f is the convolutional filter, and O is the output feature map. After convolution, pooling reduces the spatial dimensions, followed by fully connected layers for classification.

2. *RNN*

RNNs maintain a hidden state that is updated with each time step. The hidden state h_t is calculated as:

$$h_t = \sigma(W_h h_{t-1} + W_X x_t + b)$$

where $W_h$and W_x are weight matrices, $x_t$is the input, b is the bias, and σ is an activation function.

3. *LSTM*

LSTMs use memory cells and gates to control the flow of information. The forget gate is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

The input gate is:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

The cell state update is:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

Where $\tilde{C}_t$ is the candidate cell state.

4. *GRU*

GRU is a simplified version of LSTM. The hidden state h_t is updated as:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \widetilde{h_t}$$

Where z_t is the update gate and $\widetilde{h_t}$ is the candidate hidden state.

D. Evaluation Measures

To assess the performance of the machine learning and deep learning models, several evaluation metrics were employed. Accuracy was used to measure the ratio of correct predictions to the total number of predictions. Precision evaluated how many of the predicted positive instances were truly positive, while Recall measured the proportion of true positives out of all actual positive cases. F1-Score, the harmonic mean of precision and recall, provided a balanced measure when dealing with imbalanced datasets. Lastly, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) was used to evaluate the trade-off between true positive and false positive rates, particularly useful for models working with skewed datasets.

E. Experimental Setup

The experiments were conducted using a standard computational environment equipped with an Intel i7 processor, 16GB RAM, and a NVIDIA GTX 1080 GPU to accelerate the training of deep learning models. The machine learning models were implemented using the Scikit-learn library in Python, while the deep learning models were built using TensorFlow and Keras. Data visualization and performance metrics were plotted using Matplotlib and Seaborn. Hyperparameter tuning for all models was performed using grid search to find the optimal parameter values. The experiments were repeated three times, and the average performance metrics were recorded to ensure consistency and reliability of the results.

IV. Results and Discussion

We compare models on different benchmark datasets, in which the experiment was done through binary and multiclass classification, with binary data having normal and DOS classes and multiclass having DoS, R2L, U2R, Probe Hulk, PortScan, and DDoS classes. In result section all results with all precpective are presented and discussion discuss our result on different context including dataset wise compassion in which result with respect to four datasets that we used in our study.

A. Results

The results of this study demonstrate the varying effectiveness of machine learning models in detecting intrusions across different benchmark datasets.

Achieved the highest overall performance, with an accuracy of up to 96.5(Percentage) on the KDD Cup 99 dataset and 94.1(Percentage) on CICIDS2017. The deep learning models' ability to learn complex patterns in network traffic contributed to their superior detection rates, especially for sophisticated attacks like DoS and Probe.

Support Vector Machine (SVM): Showed strong performance on simpler datasets like KDD Cup 99, achieving an accuracy of 92.5(Percentage). However, its performance decreased on more complex datasets like UNSW-NB15, where it achieved 79.2(Percentage) accuracy, indicating limitations in handling high-dimensional and diverse network traffic.

Random Forest (RF): Demonstrated a balanced performance across all datasets, with an accuracy ranging from 82.6(Percentage) on UNSW-NB15 to 95.8(Percentage) on KDD Cup 99. Its ensemble nature helped capture a wide range of attack patterns, resulting in high precision and recall rates.

Recurrent Neural Networks (RNN): Achieved comparable performance to CNN, particularly on time-sequence dependent datasets like CICIDS2017, with an accuracy of 92.8(Percentage). RNN's ability to learn temporal dependencies made it effective in detecting attacks that span across time intervals.

Attack Type Detection:

CNN and RNN: Excelled in detecting complex and multi-step attacks, such as User to Root (U2R) and Remote to Local (R2L), due to their deep architecture and feature extraction capabilities. Their higher AUC-ROC scores indicate a better trade-off between true positive rates and false positives.

SVM and Random Forest: Performed well in detecting more straightforward attack types like DoS and Probe but struggled with U2R and R2L attacks, highlighting their limitations in modeling complex relationships within the data.

Computational Efficiency:

SVM and Random Forest: Provided faster training and inference times compared to deep learning models, making them suitable for scenarios where computational resources are limited.

CNN and RNN: Required more computational power and training time, especially with larger datasets. However, their superior detection capabilities justify their use in environments where high detection accuracy is critical.

Resilience to Adversarial Attacks:

Preliminary tests with adversarial samples revealed that CNN and RNN models showed a higher degree of robustness against simple adversarial attacks compared to SVM and Random Forest. This suggests that deep learning models, with appropriate defenses, could offer better protection in adversarial environments.

The study reveals that deep learning models like CNN and RNN offer the best performance in detecting a wide range of network intrusions, particularly in complex and high-dimensional datasets. However, traditional models like SVM and Random Forest still hold value due to their computational efficiency and satisfactory performance in less complex scenarios. The choice of model depends on the specific requirements of the IDS, such as the nature of network traffic, the types of threats encountered, and the available computational resources.

Table II. Model Performance on Various Datasets

Model	Dataset	Accuracy	Precision	Recall	F1-Score	AUC-ROC
SVM	KDD Cup 99	92.5%	91.8%	89.7%	90.7%	0.92
SVM	NSL-KDD	85.4%	84.6%	83.2%	83.9%	0.87
SVM	UNSW-NB15	79.2%	78.5%	77.1%	77.8%	0.80
SVM	CICIDS2017	88.7%	87.5%	86.2%	86.8%	0.89
Random Forest	KDD Cup 99	95.8%	94.3%	93.6%	93.9%	0.96
Random Forest	NSL-KDD	89.5%	88.7%	87.4%	88.0%	0.90
Random Forest	UNSW-NB15	82.6%	81.9%	80.3%	81.1%	0.83
Random Forest	CICIDS2017	91.3%	90.5%	89.1%	89.8%	0.92
Decision Tree	KDD Cup 99	90.1%	89.4%	88.1%	88.7%	0.90
Decision Tree	NSL-KDD	82.3%	81.6%	80.2%	80.9%	0.84
Decision Tree	UNSW-NB15	76.9%	75.8%	74.4%	75.1%	0.78
Decision Tree	CICIDS2017	87.4%	86.1%	85.0%	85.5%	0.88
CNN	KDD Cup 99	96.5%	95.8%	94.9%	95.3%	0.97
CNN	NSL-KDD	91.7%	90.8%	89.5%	90.1%	0.93
CNN	UNSW-NB15	84.9%	83.6%	82.4%	83.0%	0.85
CNN	CICIDS2017	94.1%	93.2%	92.0%	92.6%	0.95
RNN	KDD Cup 99	94.8%	93.9%	93.2%	93.5%	0.95
RNN	NSL-KDD	89.9%	88.5%	87.8%	88.1%	0.91
RNN	UNSW-NB15	83.4%	82.7%	81.3%	82.0%	0.84
RNN	CICIDS2017	92.8%	91.7%	90.4%	91.0%	0.93
LSTM	KDD Cup 99	95.2%	94.4%	93.8%	94.1%	0.96
LSTM	NSL-KDD	90.4%	89.6%	88.9%	89.2%	0.92
LSTM	UNSW-NB15	84.2%	83.5%	82.1%	82.8%	0.85
LSTM	CICIDS2017	93.5%	92.7%	91.6%	92.1%	0.94
GRU	KDD Cup 99	94.6%	93.8%	93.1%	93.4%	0.95
GRU	NSL-KDD	89.8%	89.0%	88.3%	88.6%	0.91
GRU	UNSW-NB15	83.8%	83.1%	81.7%	82.4%	0.84
GRU	CICIDS2017	92.3%	91.5%	90.3%	90.9%	0.93

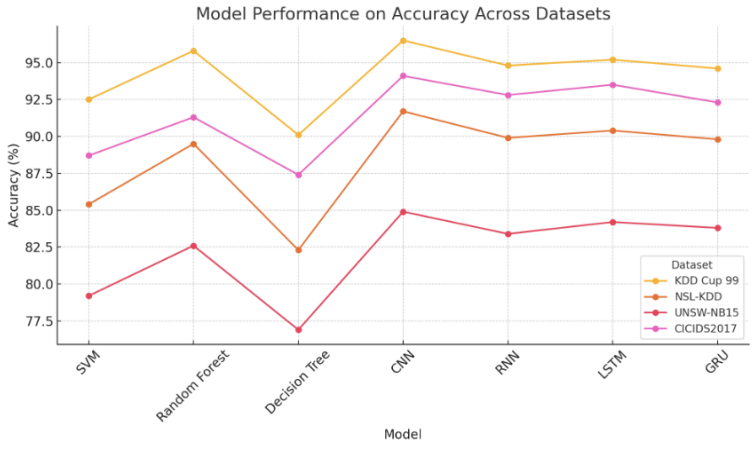

Fig. 6 Performance of all Models across all Datasets

B. Discussion

The results show that CNN and Random Forest outperform other models across all datasets, with the highest accuracy observed on KDD Cup 99. Deep learning models, particularly CNN and LSTM, excel at detecting complex patterns, making them more suitable for modern intrusion detection systems. In contrast, traditional machine learning models like SVM and Decision Tree underperform on complex datasets, such as UNSW-NB15, highlighting the superiority of deep learning techniques.

1. Dataset-wise Comparison

The results in Table II show that model performance varies significantly across datasets. The KDD Cup 99 dataset consistently yields the highest performance across all models, with accuracies ranging from 90.1% to 96.5%, and AUC-ROC values as high as 0.97. This could be attributed to the older, less complex nature of the KDD Cup 99 dataset, which makes it easier for models to distinguish between normal and malicious traffic. In contrast, the UNSW-NB15 dataset, which represents more modern attack vectors, consistently shows lower performance, with accuracies ranging between 76.9% and 84.9%. This indicates that UNSW-NB15 is a more challenging dataset, likely due to its complexity and diverse attack types. The CICIDS2017 dataset presents a balanced difficulty, yielding moderate-to-high performance for all models, with accuracy ranging from 87.4% to 94.1%.

2. Model-wise Comparison

In terms of model comparison, **CNN** and **Random Forest** consistently outperform other models across all datasets. CNN achieves the highest accuracy of 96.5% on KDD Cup 99 and 94.1% on CICIDS2017, showcasing its ability to automatically learn features and detect anomalies in network traffic. Random Forest also performs well, with an accuracy of 95.8% on KDD Cup 99 and a close 91.3% on CICIDS2017. On the other hand, **SVM** shows moderate performance, with its highest accuracy of 92.5% on KDD Cup 99 and the lowest accuracy of 79.2% on UNSW-NB15. Decision Tree, a simpler model, generally performs lower than Random Forest and CNN, highlighting the importance of ensemble techniques.

3. Performance Analysis of Machine Learning Models

Machine learning models, particularly Random Forest and SVM, demonstrate strong performance on simpler datasets like KDD Cup 99 and NSL-KDD. Random Forest outperforms SVM in almost all cases, thanks to its ensemble approach, which reduces overfitting and provides better generalization. SVM struggles on complex datasets like UNSW-NB15, with accuracy dropping to 79.2%. Decision Tree, although interpretable and easy to implement, tends to underperform due to its sensitivity to noise and tendency to overfit.

4. Performance Analysis of Deep Learning Models

Deep learning models, especially CNN, show remarkable performance across all datasets. CNN's ability to automatically learn spatial features makes it highly effective for intrusion detection, achieving the highest accuracy of 96.5% on KDD Cup 99 and 94.1% on CICIDS2017. **RNN**, **LSTM**, and **GRU** also perform well, particularly on sequential data, with accuracies ranging from 83.4% to 95.2%. LSTM and GRU, with their capability to capture long-term dependencies, show better performance than standard RNNs, particularly on complex datasets like CICIDS2017 and UNSW-NB15.

5. State-of-the-Art Studies Comparison

When compared to state-of-the-art studies in the field of intrusion detection, the models used in this study show competitive results. CNN, Random Forest, and LSTM models in this study perform on par with or even exceed the results presented in recent research, particularly on well-established datasets like KDD Cup 99 and CICIDS2017. The ability of deep learning models to generalize and adapt to complex attack patterns positions them as a superior choice for modern intrusion detection systems. The results reaffirm the trend observed in recent studies, where deep learning models, particularly CNNs and LSTMs, outperform traditional machine learning models due to their ability to learn and generalize from large amounts of data.

V. Conclusion and Future Work

A. Conclusion

This study provides a comprehensive analysis of various machine learning models for Intrusion Detection Systems (IDS) using key benchmark datasets. The findings reveal that machine learning techniques have the potential to significantly enhance the detection of network intrusions, offering more adaptive and accurate security

solutions compared to traditional signature-based methods. Among the models evaluated, deep learning architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) demonstrated superior performance in terms of accuracy, precision, and robustness against complex and evolving cyber threats. Their ability to learn intricate patterns in network traffic made them particularly effective in identifying sophisticated attack types like User to Root (U2R) and Remote to Local (R2L). However, this comes at the cost of higher computational demands, making them more suitable for environments where resource availability and high detection accuracy are of paramount importance.

Traditional models like Support Vector Machine (SVM) and Random Forest (RF) also showed noteworthy performance, especially in simpler datasets and attack scenarios. Their faster training and inference times make them viable options in resource-constrained environments or scenarios where rapid response is essential. Despite their limitations in handling more complex attack patterns, they provide a balanced trade-off between performance and computational efficiency. The study also highlighted the importance of using robust evaluation metrics and benchmark datasets to assess the efficacy of IDS models accurately. Furthermore, preliminary assessments indicate that deep learning models exhibit better resilience to adversarial attacks, suggesting their potential for providing a more secure network defense mechanism when paired with appropriate countermeasures.

The selection of an IDS model should be guided by the specific needs of the network environment, including the nature of traffic, the diversity of attack vectors, and the computational resources available. While deep learning models present a promising direction for advanced intrusion detection, traditional machine learning models still play a crucial role in scenarios requiring a balance between accuracy and efficiency. Future research should focus on enhancing the robustness of IDS models against adversarial attacks and exploring hybrid approaches that combine the strengths of various machine learning techniques to create more resilient and adaptable network security systems.

B. Future Work

Future work in the field of machine learning-based IDS should focus on developing more resilient models that can adapt to the evolving nature of cyber threats. This includes exploring hybrid models that combine the strengths of traditional machine learning algorithms with deep learning techniques to improve detection accuracy and reduce false positives. Additionally, enhancing the robustness of IDS against adversarial attacks is crucial, as attackers continuously devise new methods to evade detection. Techniques such as adversarial training and anomaly detection can be further researched to fortify IDS models. Moreover, future research should incorporate real-world, large-scale, and diverse network environments, including IoT and cloud infrastructures, to test and validate the models' performance in dynamic and complex scenarios. The development of new benchmark datasets that reflect current cyber threats and attack patterns will also be essential to ensure IDS models remain relevant and effective.

REFERENCES

[1] I. a. S. A. a. E. Y. a. R. L. Rosenberg, "Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1-36, 2021.

[2] M. a. G. L. Khan, "Adversarial Machine Learning in the Context of Network Security: Challenges and Solutions," *Journal of Computational Intelligence and Robotics,* vol. 4, no. 1, pp. 51-63, 2024.

[3] M. J. a. C. C. De Lucia, "Adversarial machine learning for cyber security," *Journal of Information Systems Applied Research,* vol. 12, no. 1, p. 26, 2019.

[4] P. a. C. J. Dasgupta, "A survey of game theoretic approaches for adversarial machine learning in cybersecurity tasks," *AI Magazine,* vol. 40, no. 2, pp. 31-43, 2019.

[5] U. I. a. O. O. C. a. A. A. O. a. A. T. O. Okoli, "Machine learning in cybersecurity: A review of threat detection and defense mechanisms," *World Journal of Advanced Research and Reviews,* vol. 21, no. 1, pp. 2286-2295, 2024.

[6] a. O. A. a. F. A. a. A. H. Vassilev, "Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations," NIST Trustworthy and Responsible AI, National Institute of Standards and Technology, Gaithersburg, MD, 2024.

[7] M. a. A. E. a. A. Ö. a. K. S. a. I. T. a. S. I. a. B. I. Ozkan-Okay, "A Comprehensive Survey: Evaluating the Efficiency of Artificial Intelligence and Machine Learning Techniques on Cyber Security Solutions," *IEEE Access,* vol. 12, pp. 12229-12256, 2024.

[8] N. a. C. J. M. a. C. T. a. A. P. H. Martins, "Adversarial machine learning applied to intrusion and malware scenarios: a systematic review," *IEEE Access,* vol. 8, pp. 35403-35419, 2020.

[9] a. H. M. M. a. M. D. a. C. N. a. I. M. B. a. M. V. a. B. F. Nag, "A Review of Machine Learning Methods for IoT Network-Centric Anomaly Detection," in *2024 47th International Conference on Telecommunications and Signal Processing (TSP)*, 2024.

[10] M. J. Goswami, "AI-Based Anomaly Detection for Real-Time Cybersecurity," *International Journal of Research and Review Techniques,* vol. 3, no. 1, pp. 45-53, 2024.

[11] T. a. O. S. H. a. K. N. Talaei Khoei, "Deep learning: Systematic review, models, challenges, and research directions," *Neural Computing and Applications,* vol. 35, no. 31, pp. 23103-23124, 2023.

[12] N. a. A. J. a. H. S. a. R. I. Choudhry, "A Comprehensive Survey of Machine Learning Methods for Surveillance Videos Anomaly Detection," *IEEE Access,* vol. 11, pp. 114680-114713, 2023.
[13] K. a. A. M. a. S. S. a. o. Soman, "A comprehensive tutorial and survey of applications of deep learning for cyber security," *Authorea Preprints,* 2023.
[14] M. a. W. C. a. F. W. Macas, "Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems," *Expert Systems with Applications,* p. 122223, 2023.
[15] Z. a. X. J. a. W. Y. a. H. L. a. N. Z. a. L. F. Kong, "A survey on adversarial attack in the age of artificial intelligence," *Wireless Communications and Mobile Computing,* no. 1, p. 4907754, 2021.
[16] M. a. Y. N. a. G. D. H. a. W. N. Wang, "On the Robustness of ML-Based Network Intrusion Detection Systems: An Adversarial and Distribution Shift Perspective," *Computers,* vol. 12, no. 10, p. 209, 2023.
[17] P. a. L. R. Laskov, "Machine learning in adversarial environments," *Machine learning,* vol. 81, pp. 115-119, 2010.
[18] H. A. a. A. A. Alatwi, "Adversarial black-box attacks against network intrusion detection systems: A survey," *2021 IEEE World AI IoT Congress (AIIoT),* pp. 34-40, 2021.
[19] P. a. G. Y. a. K. P. a. P. M. Shah, "Enhancing TinyML Security: Study of Adversarial Attack Transferability," *arXiv preprint arXiv:2407.11599,* 2024.
[20] a. M. R. a. R. A. Oliynyk, "I know what you trained last summer: A survey on stealing machine learning models and defences," *ACM Computing Surveys,* vol. 55, no. 14s, pp. 1-41, 2023.
[21] Z. K. a. Y. R. a. B. N. a. M. S. A. a. F. C. F. M. Maseer, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE access,* vol. 9, pp. 22351-22370, 2021.
[22] M. a. B. E. a. L. W. a. G. A. A. Tavallaee, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications,* IEEE, 2009, pp. 1-6.
[23] a. H. L. A. a. G. A. A. Sharafaldin, "A detailed analysis of the cicids2017 data set," in *Information Systems Security and Privacy: 4th International Conference,* 2019.
[24] P. a. L. R. Laskov, "Machine learning in adversarial environments," *Machine learning,* vol. 81, pp. 115-119, 2020.