IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 October 2024, accepted 13 December 2024, date of publication 19 December 2024,
date of current version 31 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3520406

RESEARCH ARTICLE

Adversarial Attack Detection Approach for Intrusion Detection Systems

ELİF DEĞİRMENCİ[1,2], (Member, IEEE), İLKER ÖZÇELİK[3], (Member, IEEE), AND AHMET YAZıCı[1,2], (Member, IEEE)

[1]Department of Computer Engineering, Eskisehir Osmangazi University, 26040 Eskişehir, Türkiye
[2]Center for Intelligent Systems Applications Research (CISAR), Eskisehir Osmangazi University, 26040 Eskişehir, Türkiye
[3]Department of Software Engineering, Eskisehir Osmangazi University, 26040 Eskişehir, Türkiye

Corresponding author: Elif Değirmenci (edegirmenci@ogu.edu.tr)

ABSTRACT The adoption of deep learning has exposed significant vulnerabilities, especially to adversarial attacks that cause misclassifications through subtle small perturbations. Such attacks challenge security-critical applications. This study addresses these vulnerabilities by proposing a novel adversarial attack detection method leveraging data reconstruction errors. We evaluate this approach against three well-known adversarial attacks—Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and Basic Iterative Method (BIM)—on Intrusion Detection Systems. Our method combines reconstruction error alongside aleatoric, epistemic, and entropy metrics to distinguish between original and adversarial samples. Experimental results show that our approach achieves a detection success rate of 92% to 100%, outperforming existing methods, particularly at low perturbation levels. This research enhances the robustness and reliability of machine learning models against adversarial threats by using effective error metrics in adversarial detection.

INDEX TERMS Adversarial attacks, deep learning, intrusion detection systems, reconstruction error, security, machine learning resilience.

I. INTRODUCTION

In recent years, the widespread adoption of deep learning across various areas has highlighted specific inherent technological weaknesses. It has been observed that even small perturbations, indistinguishable from humans, can lead to different or incorrect classifications by deep learning models [1]. Adversarial attacks in the context of neural networks involve creating inputs similar to legitimate inputs but misclassified by the network, posing a significant challenge in deploying neural networks in security-critical areas [2]. These attacks can be formally defined as optimization problems aiming to find minimal perturbations to inputs that lead to misclassifications [3].

There is a growing interest in exploring the vulnerability of machine learning models in different domains, such as network intrusion detection systems, which are one of the prime targets of adversarial attacks [4]. Several motivations drive the exploration of adversarial attacks in the context of deep learning detection [5]. These subtle and often indiscernible attacks can drastically affect the model's predictions. The challenge lies in detecting and mitigating these attacks. Adversarial machine learning techniques are being developed to understand and mitigate these attacks [4]. These techniques involve studying attack strategies, developing detection systems, and exploring defense mechanisms [4]. The goal is to improve the security and resilience of machine learning models in the face of adversarial threats.

Various defense methods have been proposed to make deep learning networks more reliable and robust. Adversarial training is one of them. The model's resilience is strengthened by training with both adversarial and original examples [6], [7], [8], and [9]. Model training is also used for AdvGAN detection using generative adversarial networks or autoencoders [10]. Hybrid machine learning and network flow forensics methods have also been applied [11]. The defensive distillation method makes it more difficult for the attacker to mislead the model [12]. Uncertainty and closeness metrics are

The associate editor coordinating the review of this manuscript and approving it for publication was Yu Liu.

used to detect adversarial attacks and achieve high detection scores in deep learning (DL) models trained with network traffic datasets [13].

In this study, a new approach for detecting adversarial attacks is proposed. Besides the metrics used in the literature, we use the reconstruction error metric to detect adversarial attacks. Reconstruction error is the metric that measures the difference between the original data and the reconstructed data, which is the reconstruction of the compressed representation of the original data. This difference is significant in the case of adversarial attacks. Initially, adversarial samples and original samples were examined after adversarial attacks. Even with small perturbations from adversarial attacks, the best features represent the data change. It was observed that error rates in reconstructing data from the last layer of deep learning are a distinguishing feature between original and adversarial samples. During data reduction and reconstruction, small reconstruction errors occur between the original samples' initial and final states. Adversarial samples show more distortion between their initial and final states compared to the original samples. In the proposed method, we trained the reconstruction model with original samples. The weights learned by the model trained with original samples are determined according to the original sample, just like the best representation logic of the same model. When this model is used for the reconstruction of adversarial samples in the testing phase, it has been observed that the reconstruction error is higher than that of the original samples.

This paper evaluates three well-known adversarial attacks, which are the Fast Gradient Sign Method (FGSM), Projected Gradient Method (PGD), and Basic Iterative Method (BIM) on the Intrusion Detection System (IDS). This study proposes an adversarial detection approach, Reconstruction Error-based Adversarial Detection (READ), for feature-based network traffic. We used reconstruction error and three other metrics (aleatoric, epistemic, and entropy) to detect whether the data was original or adversarial. Our proposed method successfully detects these three attacks with a success rate ranging from 92% to 100%. The performance of our proposed method achieves good results as the level of perturbation of the adversarial attack increases. Notably, our proposed method achieves more successful detections at very small perturbation levels, such as 0.01, compared to the study by Tuna et al., which is recognized for achieving successful detection results in the literature.

Briefly, our contributions to this paper are as follows:

1. A method for detecting adversarial attacks based on data reconstruction errors was proposed.
2. Three adversarial attacks were evaluated and analyzed on three network traffic datasets.
3. Detailed research was conducted on which error metric will be more effective for adversarial attacks.

The remainder of this paper is structured as follows: Related Work on adversarial attack detection is reviewed in the second section. The background of the presented work is listed in the third section, and some known adversarial attacks used in this study are explained. In the fourth section, the hypothesis of the proposed detection method is given. The system design and experimental results are shown in the fifth section. Discussions and conclusions are provided in the sixth section.

II. LITERATURE REVIEW

Adversarial detection aims to identify adversarial examples before classification. Detection methods are categorized in the literature under four main headings: secondary classifier, protection-based attacks, statistics-based detection, and mutation-based detection [14]. The secondary classifier method uses an additional classification algorithm to classify adversarial examples. Adversarial and original examples are classified as binary and trained with a model that classifies them as either adversarial or binary [15]. In projection-based detection, attack detection is done by converting to a lower-dimensional space vector. In the study using PCA, adversarial images were found to have higher coefficients. Statistical-based detection methods use statistical tests to detect differences in distributions between adversarial and clean examples. Gao [16] used the Maximum Mean Discrepancy test, and Feinman et al. [17] modeled output layers as Gaussian Mixtures. These methods also face limitations against more advanced attacks. Mutation-based detection randomly alters decision boundaries and measures sensitivity to these changes. Adversarial examples, close to decision boundaries, show inconsistent classifications. Wang et al. [18] concentrated on understanding the nature of adversarial attacks. Wang et al.'s study shows that adversarial attacks tend to be close to both the original class and decision boundary. Wang et al. [18] study leveraged the proximity of adversarial examples (AEs) to their original manifold and decision boundary for accurate AE detection.

Another method for detecting adversarial attacks is using hybrid classification methods to determine if samples are adversarial or original. Pawlicki et al. [19] propose a detection approach based on adversarial machine learning with five pattern recognition algorithms. Uelwer et al. [20] analyze the detection performance of various attacks, including CW, BIM, FGSM, boundary attack, and their combinations using class scores.

A distinct approach, the EsPADA multi-model framework, uses feature extraction via N-gram, commonly used in natural language processing problems [21]. The features are stored in Counting Bloom Filters for further usage. Deep learning models' uncertainty metrics produce different results under adversarial attacks compared to original examples [17]. Tuna et al. [13] also investigated gradient-based attacks, employing five different metrics, including epistemic uncertainty, aleatoric uncertainty, scibilic uncertainty, entropy, and closeness score, to detect adversarial attack features. Another approach is the feature-squeezing method for detecting adversarial attacks on deep neural networks [22].

Ye and Liu [23] explore using feature autoencoders to detect adversarial examples. Peng et al. [24] leverage a bidirectional generative adversarial network to learn the distribution of normal data and identify adversarial samples, thereby improving the robustness and accuracy of network IDS in adversarial environments.

These studies above highlight the importance of developing effective adversarial detection methods for network intrusion detection systems. Despite pioneer studies in literature, detecting adversarial attacks is still an open research area. It requires investigation of various methods including data reconstruction, dimensionality reduction, uncertainty metrics and adversarial attack methods.

III. BACKGROUND

Adversarial attack detection requires a variety of backgrounds. This study evaluates the effects of three white-box adversarial attacks on a network intrusion detection system. The details of these adversarial attacks are presented in this section. The dimensionality reduction and data reconstruction techniques used in machine learning and deep learning models for IDS, which also play a critical role in our adversarial attack detection approach, were explained. Uncertainty metrics, which are used to evaluate the reliability and robustness of deep learning models, were also discussed.

A. MACHINE LEARNING AND DEEP LEARNING

This section delves into two critical aspects of deep learning: dimensionality reduction and data reconstruction. Dimensionality reduction and data reconstruction are important for enhancing the effectiveness of machine learning models by focusing on essential information and discarding irrelevant or redundant data. According to our observations, these processes affect adversarial samples differently than original samples. Therefore, their role in detecting adversarial attacks will be explained in detail.

1) DIMENSIONALITY REDUCTION

In deep learning architectures used for classification problems, it is critical to reduce data dimension before the classification layer. Deep learning architectures apply complex transformations between layers to obtain efficient and useful data structures from input data. This process aims to reduce data dimensions while preserving its meaning. This would help to perform effective classification.

Let's assume x is an input vector going through k layers in the network, and the pre-activation function is given in (1).

$$z_k = W_k h_{k-1} + b_k \tag{1}$$

where W_k represents the weight matrix, b_k is the bias vector for the k^{th} layer, and h_1 is set x (the input data). The activation function (f) of this layer is then computed using a non-linear function such as sigmoid (σ), hyperbolic tangent (tanh), or Rectified Linear Unit (ReLU). This process effectively transforms the original data x into a new form with reduced dimension, as given in (2).

$$h_1 = f(z_l) \tag{2}$$

One of the powerful utilities of this reduced representation (h_K) is its capability to be utilized in transfer learning. Often, a neural network trained for a specific task (e.g., image classification) can be repurposed for another related task by stripping away its last hidden layers and retaining up to h_L. This truncated network can be treated as a feature extractor.

2) DATA RECONSTRUCTION

Data reconstruction is the process of re-generating data from its smaller representation to the original dimension. Given the dataset D with n high dimensional data points, D= $\{X_1, X_2, \ldots, X_n\}$ where each X_i is a vector in R^m. R^m is the original sample space in m-dimensional space. Each vector x_i represents a data point within this high-dimensional space, containing m features or attributes. Data reduction is a smaller representation of the data denoted as f; this function compresses the data from the high-dimensional space R^m to a lower-dimensional representation R^k. Data reconstruction is a function aimed at retrieving the original data by finding the inverse of the data reduction function (denoted as g). his function reconstructs the original data from its compressed form R^k back to the high-dimensional space $R^{m\prime}$ Such that:

$f : R^m \rightarrow R^k$ compresses the data to a smaller representation where k<m.

$g : R^k \rightarrow R^{m\prime}$ reconstructs the original data space from the compressed form.

The reconstruction error measures the difference between the original data and its reconstruction. There are several methods to calculate reconstruction error, like mean square error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and median absolute error (MedAE) [25], [26], [27]. The reconstruction success can be calculated using the error calculation formulas. A low error rate indicates the correctness of the selected reconstruction method. The reconstruction error formula of MSE is defined as:

$$MSE = \sum_{i=1}^{n} ||X_i - g(f(X_i))||^2 \tag{3}$$

where:

- n is the number of data points,
- X_i is the original data point,
- $f(d_i)$ is the reduced representation of the data point,
- $g(f(X_i))$ is the reconstructed data point,
- $||X_i - g(f(X_i))||^2$ is the squared error between the original and reconstructed data points.

Another reconstruction error metric MedAE mathematical formula is defined as:

$$MedAE = median(||g(f(X_i) - X_i)||)_{i=1}^{n} \tag{4}$$

where:

- $||g(f(X_i) - X_i)||$ is the absolute error between the original and reconstructed data points,
- The median is taken over all n data points.

These metrics provide insights into how closely the reconstructed data approximates the original data. In practical applications, choosing the appropriate metric depends on the specific requirements and characteristics of the data and the reconstruction method. For instance, MSE is sensitive to outliers, whereas MedAE provides a robust measure that is less affected by extreme values.

3) UNCERTAINTY METRICS

Various methods are used to detect adversarial attacks, such as defensive distillation, statistical methods, GAN-based detection, uncertainty, and closeness metrics. Our study evaluated aleatoric and epistemic uncertainty metrics and entropy metrics for detecting adversarial samples. Many metrics have been used in studies. Based on our tests and literature research, we found that aleatoric, epistemic, and entropy metrics are more successful in perception. Therefore, we decided to use them as references.

In machine learning, uncertainties have been evaluated using statistical methods [28], [29]. Aleatoric uncertainty is based on the randomness of the training data [30]. Tossing a coin can be given as an example. There is a possibility that the coin will land heads or tails, but the certainty is not sure. Or it is the unpredictability of natural events such as the arrival of a hurricane [31]. Aleatoric uncertainty is a measurable metric that cannot be manipulated [32]. High aleatoric uncertainty means that the model cannot learn stably, and the noise or variability in the data. Therefore, an adversarial sample might purposefully change features in the testing. This results to high aleatoric uncertainty. Addressing aleatoric uncertainty is crucial for improving the robustness and reliability of machine learning models, particularly in tasks where the data exhibits significant variability and unpredictability [33].

On the other hand, epistemic uncertainty is based on the model's lack of knowledge [30]. Compared to other uncertainty metrics, this metric can be overcome with more information. In the context of machine learning and predictive modeling, epistemic uncertainty can arise when the training data is not representative of the true distribution of the data, leading to a lack of generalization in the model's predictions [33]. In adversarial attacks, it has been observed that small perturbations in the data cause differences in both of these metrics [13], [31]. Epistemic uncertainty is particularly relevant because adversarial samples often fall outside the model's learned distribution. High epistemic uncertainty suggests the model is less certain about its predictions due to a lack of training on similar data, making it an indicator of potentially adversarial samples.

Another approach for detecting adversarial attacks is entropy as a metric. It measures how decisively data is used when making decisions during the testing phase. A high entropy value indicates a higher level of uncertainty and randomness; this suggests that the data may not provide clear distinctions for decision-making. Conversely, a low entropy value implies a more stable and informative dataset that can contribute to more robust and reliable decision-making processes. Understanding the entropy of data is necessary to evaluate the quality and reliability of information used in decision-making processes, especially in machine learning and statistical modeling [34]. High entropy in predictions indicates low confidence and uncertainty, which may be due to an adversarial sample attempt. One pixel adversarial attack can be detected using the entropy metric by analyzing the entropy level as low or high [35]. Adversarial attacks like PGD, CW, Spatial, and Hopskip for images can be detected by analyzing entropy-changing image processing methods [36]. In both image and attack detection systems, the entropy value is used for detection along with many metrics.

B. ADVERSARIAL ATTACKS

1) FAST GRADIENT SIGN METHOD (FGSM)

Goodfellow et al. [1] proposed the FGSM in 2014 to deceive deep learning methods. FGSM was initially conceptualized to deceive deep learning techniques, predominantly in image processing. However, its utility has been expanded to encompass areas such as network traffic detection [37] and NLP [34]. FGSM offers a singular adversarial attack mechanism applicable to targeted and untargeted attacks. The method integrates a user-determined proportion (e) of the utmost loss to the primary data, as denoted in (5). Initially, it discerns the prediction, original, and either targeted or nearest decision boundaries then supplements the genuine data with amplified loss through a multiplier termed epsilon. FGSM attack sample generation function is:

$$X_{adv} = X + e^{*}\text{sign}(\nabla_x J(X, y)) \tag{5}$$

where;

- X_{adv} : The modified input is designed to fool a machine-learning model.
- X: The original input that is normally classified correctly.
- e: A small number that limits the amount of modification applied.
- sign(·): Function returning the sign of the gradient.
- ∇_x J(X, y): Gradient of the loss function with respect to the input data.
- *J(X,y)*: The loss function indicates model performance.
- *y*: True label of the data.

2) BASIC ITERATIVE METHOD (BIM)

Kurakin et al. [38] proposed an iterative adversarial attack method called Basic Iterative Method (BIM) in 2016. BIM was proposed as an iterative approach for targeted or untargeted use since targeted attacks require the attack to attain more perturbation to achieve the targeted class.

The BIM formula is given in (6). BIM is also known as I-FGSM. First, the attacker calculates the loss for the targeted or untargeted classes. Then, the alpha value represents the step size or the amount of perturbation added to the data in

each iteration. The alpha value (a) and maximum iteration are one of the hyperparameters of BIM. BIM attack generation procedure is:

$$X_{adv,0} = X,$$
$$X_{adv,n+1} = X_{adv,n} + a^{*}\text{sign}(\nabla_x J(X_{adv,n}, y) \quad (6)$$

where;

- $X_{adv,n}$: The adversarial data at the n^{th} iteration.
- $X_{adv,n+1}$: The adversarial data at the $(n+1)^{th}$ iteration, which is the updated adversarial data.
- a: The step size or learning rate that scales the update.
- sign(·): The sign of the gradient of the loss function with respect to the adversarial data at the nth iteration. This indicates the direction of the update.
- $\nabla_x J(X_{adv,n}, y)$: Gradient of the loss function with respect to the adversarial data at the n^{th} iteration.
- $J(X_{adv,n},y)$: Loss function, indicating model performance.
- y: True label of the data.

3) PROJECTED GRADIENT DESCENT (PGD) METHOD

Madry et al. [39] proposed the Projected Gradient Descent (PGD) in 2017. This methodology, adept at crafting adversarial examples, resolves the intrinsic maximization problem. The maximization is achieved by iteratively adjusting the input to maximize the loss function, which leads to the generation of adversarial examples. It stands compatible with both FGSM and PGD methods, whether targeted or untargeted. Unlike BIM, where iterations invariably commence from the same point, PGD starts arbitrarily from varied initial positions within the neighboring vicinity but identifies the optimal direction via the BIM approach. The PGD equation is outlined in (7).

$$X_{adv,0} = X,$$
$$X_{adv,n+1} = P_e(X_{adv,n} + a * sign(D_x J(X_{adv,n}, y)) \quad (7)$$

where;

- $X_{adv,0}$: The initial adversarial data equals the original data.
- $X_{adv,n+1}$: The adversarial data after the $(n+1)^{th}$ update.
- P : A projection function that ensures the adversarial data stays within a valid range (often within a certain distance from the original data).
- a : The learning rate or step size for the update.
- sign(·): The sign of the gradient of the loss function with respect to the adversarial data at the n^{th} iteration, which directs the adversarial update.
- $\nabla_x J(X_{adv,n}, y)$: Gradient of the loss function with respect to the adversarial data at the n^{th} iteration.
- $J(X_{adv,n},y)$: Loss function, indicating model performance.
- y: True label of the data.

IV. SYSTEM DESIGN

A. HYPOTHESİS

Given that adversarial attacks are known to perturb the most critical features of data, leading to significant changes in data representation, we hypothesize that a reconstruction-based methodology could be employed for efficient and accurate detection of these attacks. We anticipate that while these adversarial attacks can subtly alter data to mislead classification methods, they could concurrently lead to considerable shifts in the data's representation in lower dimensions. This significant shift can be exploited to detect such attacks.

In our study, we initially consider the normal scenario of data processing where X represents the original data. The process begins with a dimensionality reduction step, denoted as $F(X)$ in (8), which transforms the original data X into a reduced form Z. Following this, a reconstruction operation, $R(Z)$, is applied, attempting to revert Z back to its original form, resulting in $\hat{X}$ in (9). The reconstruction error in this case, Δ, is typically negligible, as shown in (10).

$$Z = F(X) \quad (8)$$
$$\hat{X} = R(Z) \quad (9)$$
$$\Delta = \left| X - \hat{X} \right| \quad (10)$$

Then, an adversarial attack $A(X)$ is applied to the original data, yielding a modified data X_{adv} in (11). This modified data undergoes the same dimensionality reduction and reconstruction process, which is given in (12), yielding a significantly larger reconstruction error, denoted as $\bar{\Delta}$, which is significantly larger than in the standard scenario in (13).

$$A(X) \rightarrow X_{adv} \quad (11)$$
$$R(F(X_{adv})) \rightarrow X + \bar{\Delta} \quad (12)$$
$$\bar{\Delta} \gg \Delta \quad (13)$$

For our reconstruction model, the reconstructed version of original data is $R(F(X))$ and of the adversarial attacked data is $R(F(X_{adv}))$. The error rates for original and adversarially attacked data are denoted $\epsilon_{RE}(R(F(X)))$ in (14) and $\epsilon_{RE}(R(F(X_{adv})))$ is given (15) respectively. Based on proposed hypothesis:

$$R(F(X)) \approx X$$
$$\epsilon_{\text{RE}}(\text{R}(\text{F}(X))) \rightarrow minimal \quad (14)$$
$$R(F(X_{adv})) \neq X_{adv}$$
$$\epsilon_{RE}(R(F(X_{adv}))) \rightarrow high \quad (15)$$

Our model architecture includes the last hidden layers, H and a reconstruction architecture, $R(X)$, which would provide a robust mechanism to capture and model the nuanced relationships and patterns in the transformed data representation, potentially introduced due to adversarial attacks. If $A(X) \rightarrow X_{adv}$ and $F(X) \rightarrow F(X_{adv}) + \delta$ then difference between X_{adv} and $R(F(X_{adv}))$ gauged by reconstruction error calculation ϵ_{RE}, can serve as a mechanism to detect adversarial attacks.

B. ARCHITECTURE

Our methodology is divided into two distinct segments, as illustrated in Fig. 1. The first segment involves the learning process up to the final layer of a deep learning method (Fig. 1a). The second segment focuses on reconstructing the data back to its original size from this last layer, employing a specific reconstruction model (Fig. 1b).

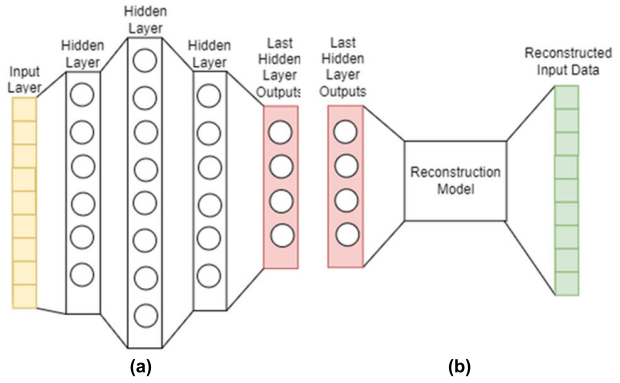


FIGURE 1. The system architectures used in the proposed approach: (a) Deep learning architecture includes the last hidden layers and output layer, (b) Reconstruction architecture takes input from the last hidden layer outputs and the output is reconstructed input data.

The first segment is represented by a DNN model, shown in Fig. 1a. This model encompasses the last hidden and output layers. It is important to note that outputs from the last hidden layer of any deep learning architecture can be extracted, not just from the DNN employed in this research.

The second segment depicted in Fig. 1b, the Reconstruction Model is a pivotal part of our approach. The hypothesis mentioned in Section IV-A is based on original samples and adversarial samples. The best feature representation is diversified, as shown in Fig. 3. We used this hypothesis and constructed a reconstruction model that reconstructs input data using the last hidden layers of input data. This model is trained using original samples, so the model's weight is optimized for original samples. Therefore, if an adversarial sample is reconstructed using this model, it is reconstructed using the original sample reconstruction weights. This led to higher divergences between input and output data (reconstructed data).

To assess the quality of the reconstructed data, we employ two metrics. The MSE, outlined in (16), quantifies the squared differences between the original and the reconstructed data. The MedAE, detailed in (17), measures their absolute differences.

- Input data vector: $X = \{x_1, x_2, \ldots, x_n\}$,
- Reconstructed data vector: $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$

$$MSE_i = \frac{1}{n}\sum_{k=0}^{n}(X_i - \hat{X}_i)^2, for\ all\ i\ \in [1, n] \tag{16}$$

$$MedAE_i = median(|\hat{X}_1 - X_1\Big|, |\hat{X}_2 - X_2\Big|, \ldots, |\hat{X}_n - X_n|) \tag{17}$$

where X_i refers to each element of the input data vector X. $\hat{X}_i$ refers to each element of the reconstructed data vector $\hat{X}$. n refers to the total number of elements in the vectors X and X." $|\hat{X}_i - X_i|$ indicates the absolute difference between corresponding elements of the input and reconstructed data. $(X_i - \hat{X}_i)^2$ indicates the squared difference between corresponding elements of the input and reconstructed data.

Algorithms 1 and 2 define the reconstruction of the input data process and reconstruction error metric functions, respectively. Algorithm 1 applies a reconstruction model trained previously with the original last hidden layer outputs and original input data. Algorithm 1 derives the reconstructed data from the hidden layer output for any input sample.

Algorithm 1 Reconstruct Input Data

function: Reconstruct_Input_Data ($\boldsymbol{R}$, $\boldsymbol{H_i}$):
input: A regressor model ($\boldsymbol{R}$), last hidden layer outputs from IDSs model ($\boldsymbol{H_i}$), input data ($\boldsymbol{X_i}$))
output: Reconstructed data ($\hat{\boldsymbol{X}}_{\boldsymbol{i}}$)
$\hat{\boldsymbol{X}}_{\boldsymbol{i}} \leftarrow \boldsymbol{R}$.**predict**($\boldsymbol{H_i}$)
return $\boldsymbol{RD_i}$
end function

Algorithm 2 then computes the reconstruction error, giving us an essential metric to evaluate our model's performance. The error computation caters to both MSE and MedAE, offering flexibility in model evaluation depending on the research question or analytical requirements.

Algorithm 2 Calculate Reconstruction Error

function: Calculate_Rreconstruction_Error ($\boldsymbol{X_i}$, $\hat{\boldsymbol{X}}_{\boldsymbol{i}}$, ***Str***)
input: input data ($\boldsymbol{X_i}$), reconstructud data ($\hat{\boldsymbol{X}}_{\boldsymbol{i}}$), error metric (***Str***) (either 'mse' or 'medae')
output: Average Mean Squared Error ($\boldsymbol{MSE_i}$), Average Median Absolute Error ($\boldsymbol{MedAE_i}$)), Error ($\boldsymbol{E}$)
if size($\boldsymbol{X_i}$)$\neq$ **size**($\boldsymbol{X_i''}$) then
raise an error "InputData and ReconstructedData must have the same shape"
else if ***Str*** = 'mse' then
$\boldsymbol{MSE_i} \leftarrow$ **mean**$((\hat{\boldsymbol{X}}_{\boldsymbol{i}} - \boldsymbol{X_i})^2)$
$\boldsymbol{E} \leftarrow \boldsymbol{MSE_i}$
else if Metric = 'medae' then
$\boldsymbol{MedAE_i} \leftarrow$ median $(|\hat{\boldsymbol{X}}_{\boldsymbol{i}} - \boldsymbol{X_i}|)$
$\boldsymbol{E} \leftarrow \boldsymbol{MedAE_i}$
Else * raise an error* "Invalid metric. Supported metrics are 'mse' and 'medae'."
return $\boldsymbol{E}$
end function

V. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

Our experimental system design, like other machine learning lifecycles, includes training and testing, and our experimental setup comprises five steps. In Fig. 2, all processes are

comprehensively shown. The first step includes two training stages. In the initial step of this research, two models were constructed: the IDS model and the reconstruction model. The IDS model was trained using a CNN-based deep learning architecture with original data samples. The second training of the first step aims to train the last hidden layer model reconstruction to be as close as possible to the original data. In this study, we employed a Linear Regression model to establish a relationship between the outputs of the last hidden layer and the input data. We initialized the Linear Regression model using the LinearRegression() function from the scikit-learn library.

The second step involved the creation of an adversarial dataset. FGSM, BIM, and PGD were employed to generate the adversarial dataset, applying various perturbation values. The third step was focused on calculating reconstruction error, entropy, and aleatoric and epistemic metrics for original and adversarial samples. As detailed in the background section, reconstruction errors were computed using two distinct algorithms, Algorithm 1 and Algorithm 2. We compare two reconstruction error metrics (MSE and MedAE), and for the experimental results, we specifically chose MedAE as the reconstruction error. In the fourth stage, an adversarial attack detection model is trained using the mentioned metrics (reconstruction error, entropy, aleatoric and epistemic).

The last step involves testing the performance of the adversarial attack detection model. Adversarial attack detection distinguishes adversarial samples from benign samples. Finally, the fifth step involves testing the adversarial detection classification. The four metrics from the test data were classified, and the prediction results were subsequently analyzed to determine whether the samples were original or adversarial.

Fig. 3 shows the testing steps of the READ model, which is implemented to enhance the IDS model. ''Original Samples,'' which may undergo adversarial attacks such as FGSM, BIM, and PGD, leading to the creation of ''Adversarial Samples.''. These samples, along with the original ones, are input into the ''READ model.'' The READ model then assesses whether a sample is adversarial. If the sample is identified as adversarial, it is classified accordingly and marked as an ''Adversarial Sample.'' If not, it is passed on to the IDS model for further analysis. This figure illustrates the critical role of the READ model in distinguishing and filtering out adversarial samples before they impact the IDS model.

B. EFFECTS OF ADVERSARIAL ATTACKS ON DIFFERENT DATA FEATURES

The proposed approach for detecting adversarial attacks fundamentally relies on the notion that insignificant modifications, which are subtle enough to go unnoticed by classification methods, can result in significant changes in the best data representation.

Our main idea first finding is that the most critical features change after adversarial attacks. This is depicted in Table 1, which shows two perspectives between original and adversarial samples. As observed in Table 1, when an FGSM attack is added with an intensity of 0.01, the five most important features of the data change slightly.

TABLE 1. The top five important features of the cse-cic-ids2018 dataset are original samples and adversarial samples.

Rank	Original Samples	Score	Adversarial Sample	Score
1	Feature: 30	3.07548	Feature: 30	8.36310
2	Feature: 8	2.80700	Feature: 8	4.94248
3	Feature: 24	1.42428	Feature: 28	4.48721
4	Feature: 6	1.02446	Feature: 24	1.46055
5	Feature: 28	0.38068	Feature: 22	1.35351

Following this initial exploration, we delve deeper into the data by studying the weight of feature importance. This analysis is graphically represented in Fig. 4. This step validates the first part's findings and provides a comprehensive understanding of how the data distribution and characteristics change after an adversarial attack, even with small perturbations).

C. EXPERIMENTAL RESULTS OF STANDARD ADVERSARIAL ATTACKS AND THEIR EFFECTS ON IDSs MODEL

In this study, experiments systematically evaluated the performance of IDSs, which aim to detect unauthorized access and malicious activities. The IDSs were trained using the CSE-CIC-IDS2018 [40], KDDCup99 [41], and ROSIDS23 [42], [43], and their performance was evaluated under standard conditions and adversarial attack scenarios.

The CSE-CIC-IDS2018 [40] dataset contains over 80 features and includes various attack types such as Brute Force, DDoS, Botnet, and Web Attacks. The KDDCup99 [41] dataset has 41 features and covers four main attack categories: DoS, Probe, U2R, and R2L. The ROSIDS23 [42], [43] is the dataset aimed at enhancing IDS capability within the Robot Operating System (ROS) environment. This dataset contributes to the advancement of cybersecurity measures tailored to the unique requirements of robotic networks, addressing vulnerabilities and potential attacks specific to the ROS ecosystem. The dataset consists of 84 features. The three datasets—KDDCup99 (145,586 samples), CSE-CIC-IDS2018 (508,248 samples), and ROSIDS23 (136,681 samples)—are evaluated using the same preprocessing steps. This process included normalization using a min-max scaler and the removal of any null or infinity values. For each dataset, 80% of the data was used for training, while the remaining 20% was used for testing.

This study focused on the three known adversarial attacks: FGSM, BIM, and PGD at varying intensity levels (0.01 to 010). Under normal conditions, both IDSs showed high classification accuracy (see Table 2). Specifically, KDDCup99 achieved the highest accuracy at 99.66%, outperforming both ROSIDS23 with 96.64% and CSE-CIC-IDS2018 with 93.71%. However, all systems' accuracy was significantly compromised under adversarial attack conditions.

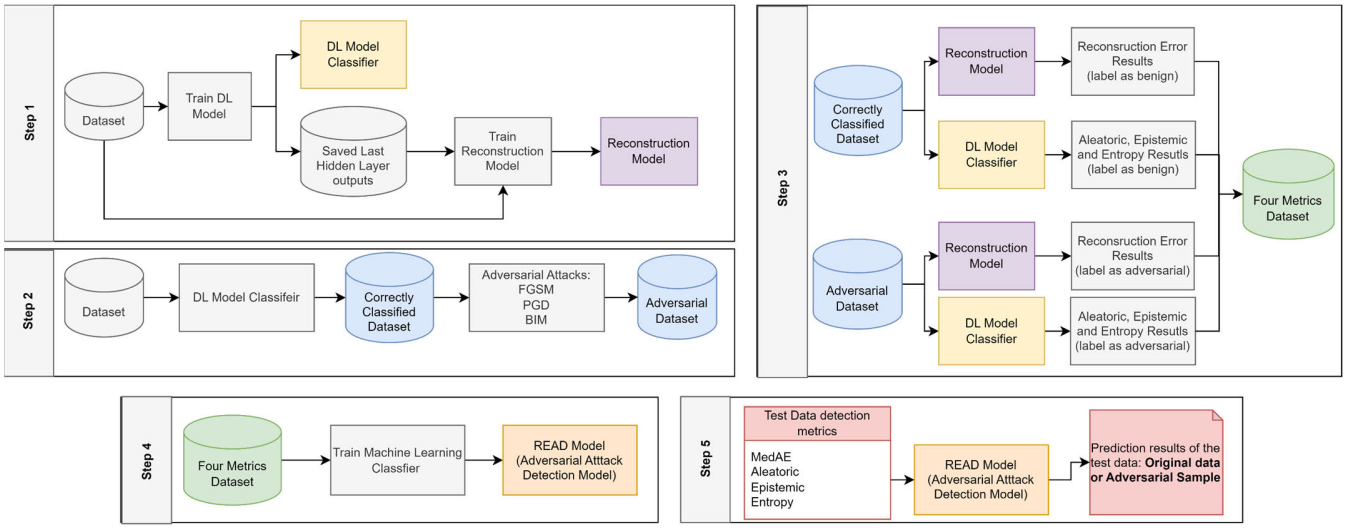


FIGURE 2. Experimental setup for the adversarial detection system.

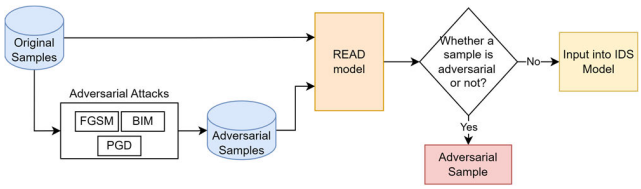
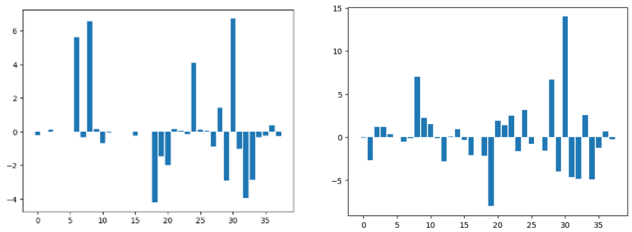


FIGURE 3. Experiment design for the test of the READ.



(a) Original Samples (b) Adversarial Samples

FIGURE 4. Feature importance weight result's graph of CSE-CIC-IDS2018 dataset.

TABLE 2. CSE-CIC-IDS2018, KDDCup99, and ROSIDS23 datasets accuracy result under normal and adversarial attacks with various perturbation amounts.

		CSE-CIC-IDS2018	KDDCup99	ROSIDS 23
	Standard IDSs Accuracy	93,71	99,66	96,64
Attack Type	**Perturbation Amount**			
FGSM	0.01	76,27	99,13	58,98
	0.04	28,01	57,91	35,64
	0.10	1,74	39,08	12,61
PGD	0.01	65,53	99,11	59,96
	0.04	4,06	51,24	29,94
	0.10	0,3	36,68	2,89
BIM	0.01	54,65	99,03	58,98
	0.04	3,25	49,95	24,21
	0.10	0,23	36,96	3,63

D. DATA RECONSTRUCTION ERROR ON THE BENIGN AND MALICIOUS SAMPLE

Our proposed methodology for detecting adversarial samples uses several metrics, and one of them relies on calculating reconstruction errors using the MSE and MedAE calculations. The reconstruction error results of the original and adversarial samples from the CSE-CIC-IDS2018 dataset are shown in Fig. 5. This indicates that the reconstruction error rates between the adversarial and original samples differ, even with a perturbation of 0.04 under the FGSM attack.

Table 3 shows the MSE and MedAE of reconstruction errors for the CSE-CIC-IDS2018, KDDCup99 and ROSIDS23 datasets under normal conditions and various adversarial attacks (FGSM, PGD, and BIM) with different perturbation amounts.

For original samples, the CSE-CIC-IDS2018 dataset has an MSE of 0.00251 and a MedAE of 0.00411, while the KDDCup99 dataset has an MSE of 0.00293 and a MedAE of 0.00505. The ROSIDS23 dataset, on the other hand, shows an MSE of 0.00657 and a MedAE of 0.01351. These values represent the baseline reconstruction errors without any adversarial perturbations.

Upon analysis, it was observed that the MSE and MedAE scores increased under the adversarial attack conditions across all datasets. This trend underscores the substantial influence of these adversarial attacks. This finding suggests more significant average deviations from the true values, indicating a more pronounced impact of adversarial attacks on this dataset.

TABLE 3. MSE and MedAE of reconstruction errors for different datasets under FGSM, PGD, and BIM with various perturbation amounts and normal samples.

		CSE-CIC-IDS2018		KDDCup99		ROSIDS23	
Attack Type	Epsilon	MSE	MedAE	MSE	MedAE	MSE	MedAE
None	0.00	0.00251	0.00411	0.00293	0.00505	0.00657	0.01351
FGSM	0.01	0.03291	0.03355	0.00644	0.01223	0.01596	0.03157
	0.04	0.03773	0.04524	0.02748	0.04484	0.12169	0.10562
	0.10	0.37858	0.12729	0.06046	0.09771	0.16085	0.11215
PGD	0.01	0.03701	0.03392	0.00643	0.01223	0.01841	0.03086
	0.04	0.29255	0.06455	0.01894	0.04484	0.0997	0.08672
	0.10	7,97365	0.16187	0.03792	0.09771	0.07323	0.13217
BIM	0.01	0.03543	0.03207	0.00648	0.01229	0.01738	0.03072
	0.04	0.48575	0.06191	0.01815	0.03802	0.14822	0.10492
	0.10	9,06683	0.15634	0.0394	0.07434	0.08148	0.16986

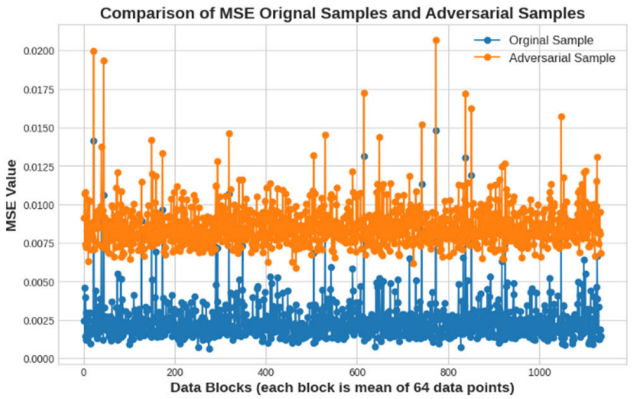


FIGURE 5. CSE-CIC-IDS2018 dataset reconstruction error results in original samples and adversarial samples.

MSE and MedAE metrics effectively highlight the differentiation between original and adversarial data. While most studies in the literature rely on MSE for adversarial attack detection, MedAE tends to perform better in the values between 0-1 due to its calculation based on absolute errors rather than squared differences.

In our study, we found that MedAE more effectively highlights reconstruction errors in the presence of small perturbations, whereas MSE becomes more effective as the intensity of the attack increases. However, MedAE introduces a greater computational burden on the detection system compared to MSE. Our approach focuses on using the MedAE metric at the initial stage to detect small perturbations accurately.

Fig. 6 presents a graphical representation of the reconstruction error outcomes for the CSE-CIC-IDS2018 dataset subjected to FGSM, PGD, and BIM with multiple perturbation degrees. It's apparent that attacks using perturbation rates differing from 0.1 are more conspicuously separable from the original dataset. Furthermore, there's a noticeable trend that with the rise of the perturbation rate, the reconstruction error of each attack starts showing a higher degree of divergence.

E. ADVERSARIAL ATTACK DETECTION USING RECONSTRUCTION ERROR

The effects of adversarial attacks have shown different impacts on various datasets for the same epsilon values (see Table 4). For this reason, when comparing our proposed approach with studies in the literature, we focused on how much performance loss the system experiences under adversarial attack. We evaluated the success of detection methods based on this performance loss. The experimental results first demonstrated that the effects on different datasets yield different outcomes depending on the epsilon values (see Table 4. Therefore, in our literature comparisons, the experiments were designed based on the performance loss caused by the attacks on the system (see Table 5).

Fig. 7 shows the ROC curves for the FGSM attack on three datasets: CSE-CIC-IDS2018, KDDCup99, and ROSIDS23, with the epsilon set to 0.01. The ROC curves compare different detection methods: Aleatoric, Epistemic, Entropy, MedAE, and a combined method (All). These results illustrate the performance of different detection methods under FGSM attacks. For the CSE-CIC-IDS2018 dataset, the combined method (All) achieved the highest AUC of 0.98, indicating strong performance in distinguishing between true and false positives. The MedAE method also performed well, with an AUC of 0.95. For the KDDCup99 dataset, the MedAE method achieved a high AUC of 0.92, while the combined method (All) had an AUC of 0.93. The other methods (Aleatoric, Epistemic, and Entropy) showed lower performance on this dataset, with AUC values of 0.61. For the ROSIDS23 dataset, the MedAE method achieved a high AUC of 0.84, while the combined method (All) also had an AUC of 0.84. The other methods (Aleatoric, Epistemic, and Entropy) showed lower performance, with AUC values of 0.59 for both Aleatoric and Entropy and 0.62 for Epistemic.

These experiments demonstrate that the combined method and MedAE effectively detect adversarial samples, especially for the CSE-CIC-IDS2018 dataset. The KDDCup99 dataset results indicate that the MedAE and combined methods are effective results indicate that the MedAE and combined

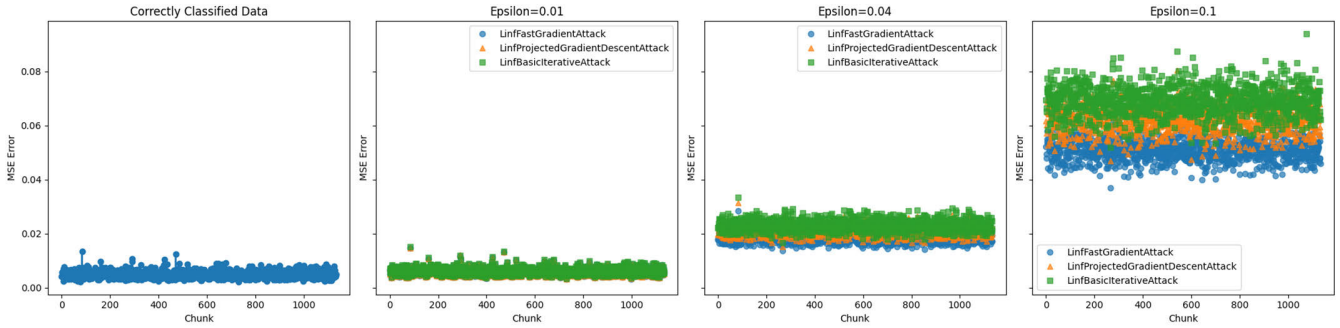
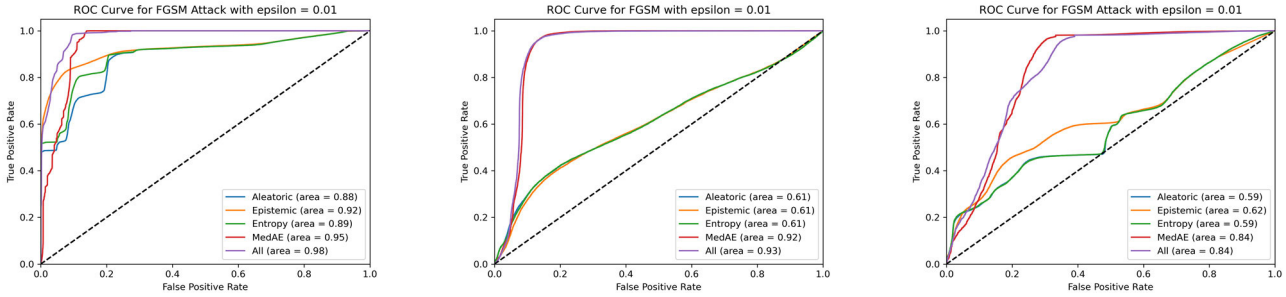


FIGURE 6. CSE-CIC-IDS2018 dataset reconstruction error results in graphical visualization under FGSM, PGD, and BIM with various perturbation amounts.



(a) CSE-CIC-IDS2018 (b) KDDCup99 (c) ROSIDS23

FIGURE 7. Comparative evaluation of ROC curves for FGSM attack.

TABLE 4. Proposed detection system ROC-AUC scores across CSE-CIC-IDS-18 and KDDCup99 datasets under FGSM, BIM, and PGD attacks at varying perturbation levels.

		CSE-CIC-IDS2018		KDDCup99		ROSIDS23	
Epsilon	Attack Type	Attack success rate	Roc-Auc detection score	Attack success rate	Roc-Auc detection score	Attack success rate	Roc-Auc detection score
0.01	FGSM	24%	96%	1%	93%	41%	84%
	BIM	35%	96%	1%	92%	40%	84%
	PGD	45%	96%	1%	93%	41%	84%
0.04	FGSM	72%	100%	42%	100%	64%	97%
	BIM	95%	99%	49%	100%	70%	97%
	PGD	96%	100%	50%	99%	76%	97%
0.1	FGSM	98%	100%	61%	100%	87%	100%
	BIM	99%	100%	63%	100%	97%	99%
	PGD	99%	100%	63%	100%	96%	99%

methods are effective. Similarly, for the ROSIDS23 dataset, the MedAE and combined methods also show strong performance in detecting adversarial samples, highlighting their robustness across different datasets.

Fig. 8 shows ROC curve performance for different detection methods on the CSE-CIC-IDS2018 dataset under FGSM attack with 0.04 perturbation. The ROC curves compare different detection methods: Aleatoric, Epistemic, Entropy, MedAE, and a combined method (All). These results indicate that the combined method (All) achieved the highest AUC of 1.00, demonstrating its effectiveness in distinguishing between true positives and false positives. The MedAE method also performed well, with an AUC of 0.99. The Epistemic and Entropy methods followed with AUC values of 0.96 and 0.94, respectively. The Aleatoric method had the lowest AUC among the methods compared, with a value of 0.93.

Even if the epsilon of the attack is the same, the impact of the attack can vary; the effect of the attack changes depending on whether the classes are close or distant, as analyzed in the literature by Wang et al. [18] study. Therefore, when comparing the effects of attacks, we evaluated the attack's success as the baseline. The decrease in the success of the IDS after an adversarial attack is evaluated as the attack success rate.

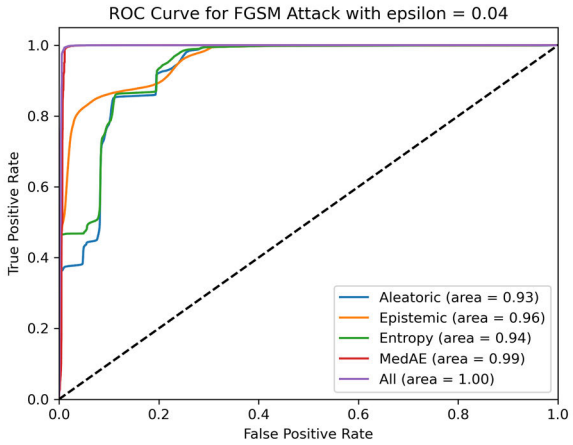


FIGURE 8. ROC curve of CSE-CIC-IDS2018 dataset under FGSM attack with 0.04 perturbation amount.

As the attack's success increases, the attack's detection also increases. At an epsilon value of 0.04, the detection of the FGSM attack, although the results for the detection of BIM and PGD attacks are very close, it is observed that FGSM's success is slightly better. This situation is because other attacks are iterative, resulting in higher impact with smaller perturbations, while FGSM works in a single iteration, adding all perturbations at once. This might slightly increase its detection.

Table 4 presents the ROC-AUC scores of a proposed detection system tested on the CSE-CIC-IDS2018, KDDCup99, and ROSIDS23 datasets under FGSM, BIM, and PGD attacks at varying perturbation levels. For the CSE-CIC-IDS2018 dataset, the attack success rate ranges from 24% to 99%, while the detection scores remain high, ranging from 96% to 100%. In contrast, for the model trained with the KDDCup99 dataset, the attack success rate is generally low at lower perturbation levels, ranging from 1% to 50%, but it increases to 63% at higher levels, with detection scores consistently high between 92% and 100%. The ROSIDS23 dataset exhibits a wider range of attack success rates from 40% to 97%, with detection scores ranging from 84% to 100%. Increasing perturbation levels lead to higher attack success rates. For example, in the CSE-CIC-IDS2018 dataset, FGSM success rates rise from 24% at 0.01 perturbation to 98% at 0.1 perturbation. In the ROSIDS23 dataset, FGSM success rates increase from 41% to 87% across the same range. The detection scores remain robust across all datasets, even as the attack success rate increases. However, the KDDCup99 dataset shows significantly lower attack success rates at lower perturbations, ranging from 1% to 50%, but it aligns more closely with the other datasets at higher perturbations, reaching up to 63%. Despite these increased attack success rates with higher perturbation levels, the detection system maintains high ROC-AUC scores across all datasets, indicating strong resilience and detection capability against varying levels of attack intensity.

These points highlight the robustness of the detection system and its effectiveness in maintaining high detection scores even as the attack success rates increase with higher perturbation levels.

Fig. 9 and Fig. 10 present the outcomes of three adversarial attacks, showing detection scores versus attack success rates. Both figures compare the proposed method's attack accuracy and ROC-AUC scores with the method by Tuna et al. [13]. For both FGSM and PGD attacks, our proposed method consistently achieves higher ROC-AUC scores across various epsilon levels, indicating better detection performance.

Fig. 9 shows the successful detection of adversarial attacks against the effect of adversarial attacks on the IDS. Fig. 9 depicts the x-axis as the success rate of adversarial attacks and the y-axis representing the response rate of the models to the attacks. Our proposed method starts above 90% detection performance for all three adversarial attacks and quickly approaches near 100%, even at lower attack success rates, demonstrating successful detection. While Tuna et al.'s method also shows high performance, closely matching ours, our proposed method achieves higher scores, particularly at lower attack accuracies.

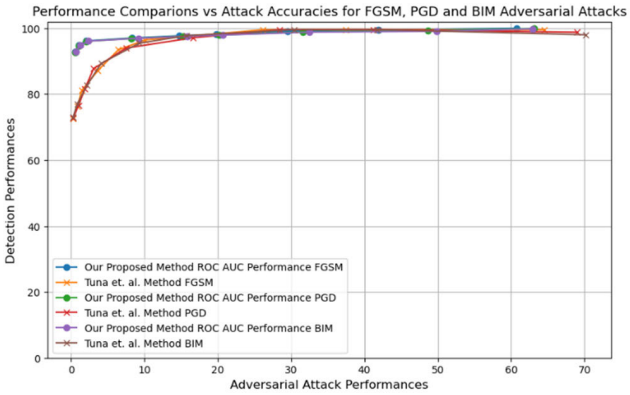


FIGURE 9. Comparative evaluation of adversarial attack detection performance on FGSM, PGD, and BIM for the proposed method and Tuna et al. [13].

In Fig. 10, the success of adversarial attacks in terms of the epsilon values and the success of detection methods are separated on a per-attack basis. The success rates of deceiving IDSs in both models follow a similar trend. The graph also shows that as the perturbation amount increases, the impact of the attacks becomes more pronounced. Similarly, even with very small perturbations of attack effect, the success of our proposed detection methods achieves better performance than Tuna et al. [13], with detection scores over 90%. As the attack's impact increases, the success of both detection methods approaches nearly 100%.

The READ performance is compared with another study that detected adversarial samples in the literature Peng *et. al.* study as ASD [24] (see in Table 5). The evaluation metrics are Accuracy and F1-Score.

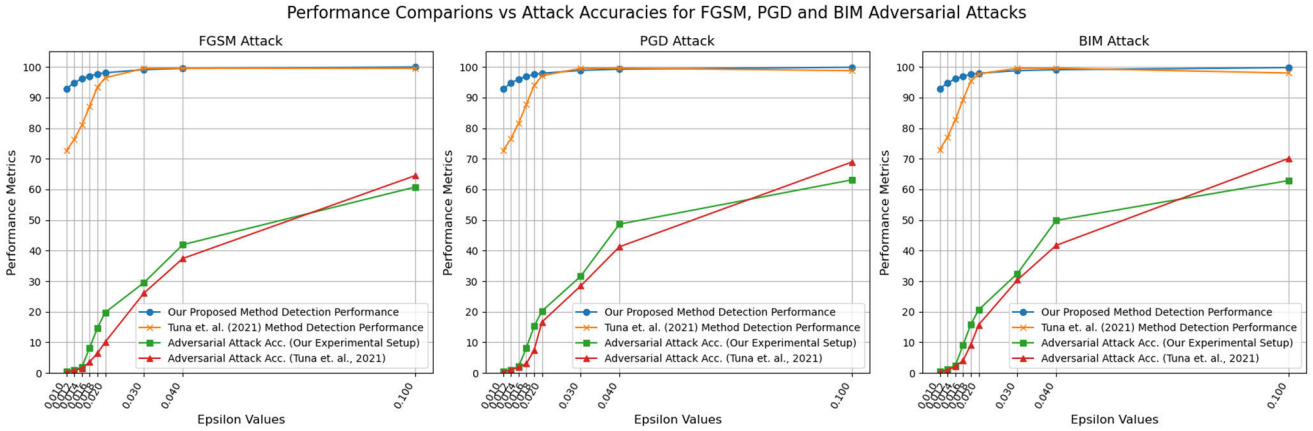


FIGURE 10. Performance comparison of proposed method and Tuna et al. [13] at adversarial attack detection ROC-AUC scores.

TABLE 5. Proposed detection system comparison for FGSM and PGD attacks.

Attack Type	Adversarial Detection Model	Accuracy	F1-Score
FGSM	None (Peng et al. [24])	61,56%	59,10%
	ASD (Peng et al. [24])	73,41%	84,46%
	None	57,84%	25,06 %
	READ	**99,69%**	**99,66%**
PGD	None (Peng et al. [24])	55,81%	55,69%
	ASD (Peng et al. [24])	82,27%	89,52%
	None	51,17%	22,81%
	READ	**98,49%**	**98,71%**

Under FGSM attacks, the ASD improves the system performance, with an accuracy of 61,56% to 73,41%. In our study, we improve the system performance with the READ, with an accuracy improvement of 57,84% to 99,69%. READ achieves higher performance than ASD on the system under FGSM attacks.

For PGD attacks, ASD improves the system performance, with an accuracy of 55,81% to 82,2%. In our study, we improve the system performance with the READ, with an accuracy improvement of 51,17% to 98,49%. READ achieves higher performance than ASD in the system under PGD attacks.

Overall, the READ system outperforms the ASD in both accuracy and F1-Score across both types of attacks, demonstrating its effectiveness in detecting adversarial attacks.

Adversarial attacks involving low-level perturbations pose challenges for ASD detection. The ASD approach, which relies on analyzing reconstruction errors and matching losses, is less effective in identifying such attacks. Small perturbations, particularly in the early stages of adversarial attacks, introduce minimal modifications to mislead the IDS's decision, resulting in decreased differences between input and reconstruction data. This makes it difficult for ASD to detect adversarial samples accurately. Our proposed method (READ), utilizing MedAE for reconstruction error calculation, offers a more effective solution, achieving high accuracy even with small perturbations. However, this approach incurs a computational overhead on the detection system.

VI. CONCLUSION

In this study, four metrics were used for attack detection. Three metrics previously used in the literature for adversarial attack detection were combined with the reconstruction error rate metric and the proposed adversarial detection approach named READ. Although high success rates starting from 0.04 were achieved in the literature, success rates exceeding 92% were observed in detecting attacks in low-impact ranges. Adding the three metrics from the literature to this proposed metric increased the success rate to 100% with computational complexity.

The impact of adversarial attacks generally increased as the epsilon perturbation rate increased. As found in previous studies, FGSM attacks were less successful compared to PGD and BIM attacks. In the proposed detection method, as the success of the adversarial attack increased, the success of the proposed attack detection also increased. Specifically, when evaluating the results at an epsilon value of 0.04, FGSM attacks were detected slightly better than other adversarial attacks. This is because FGSM is a single-step attack and is usually performed with a simple computation. Therefore, the attack manipulates the target using only the first gradient information. BIM and PGD attacks are iterative versions of FGSM. They are performed in multiple steps, making small changes at each step to increase the attack's impact. The main reason FGSM attacks are detected better is that the attack is simpler and single-step. This simplicity makes the attack more superficial and easier for detection systems to notice.

By integrating READ in front of the IDS, the robustness and effectiveness of the IDS have been significantly enhanced. In scenarios where adversarial attacks occur, the

accuracy of the IDS can decrease, falling to as low as 51%. However, with the implementation of READ, the accuracy of the IDS is substantially improved, reaching levels as high as 98%. This demonstrates that READ effectively mitigates the negative impact of adversarial attacks, ensuring a much higher detection accuracy and overall reliability of the IDS.

Our experimental results highlight the importance of incorporating multiple metrics to enhance the detection capabilities of IDSs against adversarial threats. High detection success rates are achieved with READ at high perturbation levels. Furthermore, the novelty of our method of detecting low-level perturbations is a significant advancement over existing methods in the literature.

The same adversarial attack with the same perturbation may affect various datasets differently. Wang et al. [18] study examined the differences and detection of adversarial examples for cases where decision boundaries are completely disjoint, close, or intertwined. In our study, the reason for the varying success of adversarial attacks detected on different datasets could be similar.

The current detection system's dependency on the last hidden layer of the data model necessitates access to the data, which might not always be feasible or secure. Future studies, therefore, focus on creating detection systems that operate independently of the data layer, ensuring broader applicability and enhanced security. In our approach, we used MedAE to detect smaller perturbations successfully. While MSE is computationally efficient to calculate, MedAE is more effective at capturing small perturbations. Therefore, future studies could explore the combined use of MSE and MedAE, which may enhance detection performance while optimizing computational efficiency. As a future study, the relationship between the effects of attacks on datasets and data distribution can be examined, and detection system updates can be suggested.

REFERENCES

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

[2] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[3] C. Finlay, A.-A. Pooladian, and A. Oberman, "The logbarrier adversarial attack: Making effective use of decision boundary information," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4861–4869.

[4] E. Alshahrani, D. Alghazzawi, R. Alotaibi, and O. Rabie, "Adversarial attacks against supervised machine learning based network intrusion detection systems," *PLoS ONE*, vol. 17, no. 10, Oct. 2022, Art. no. e0275971.

[5] E. Degirmenci, I. Ozcelik, and A. Yazici, "Effects of un targeted adversarial attacks on deep learning methods," in *Proc. 15th Int. Conf. Inf. Secur. Cryptogr. (ISCTURKEY)*, Oct. 2022, pp. 8–12, doi: 10.1109/ISCTURKEY56345.2022.9931786.

[6] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," 2017, *arXiv:1705.07204*.

[7] Y. Zhou, X. Zheng, C.-J. Hsieh, K.-W. Chang, and X. Huang, "Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 1–13.

[8] S. Henrique Silva and P. Najafirad, "Opportunities and challenges in deep learning adversarial robustness: A survey," 2020, *arXiv:2007.00753*.

[9] F. Liu, W. Zhang, and H. Liu, "Robust spatiotemporal traffic forecasting with reinforced dynamic adversarial training," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2023, pp. 1417–1428.

[10] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2020, pp. 1–10, doi: 10.1109/PERCOM45495.2020.9127389.

[11] K. Demertzis, N. Tziritas, P. Kikiras, S. L. Sanchez, and L. Iliadis, "The next generation cognitive security operations center: Adaptive analytic lambda architecture for efficient defense against adversarial attacks," *Big Data Cognit. Comput.*, vol. 3, no. 1, p. 6, Jan. 2019.

[12] K. T. Y. Mahima, M. Ayoob, and G. Poravi, "Adversarial attacks and defense technologies on autonomous vehicles: A review," *Appl. Comput. Syst.*, vol. 26, no. 2, pp. 96–106, Dec. 2021.

[13] O. F. Tuna, F. O. Catak, and M. T. Eskil, "Closeness and uncertainty aware adversarial examples detection in adversarial machine learning," *Comput. Electr. Eng.*, vol. 101, Jul. 2022, Art. no. 107986.

[14] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 538–566, 1st Quart., 2023, doi: 10.1109/COMST.2022.3233793.

[15] Z. Gong and W. Wang, "Adversarial and clean data are not twins," in *Proc. 6th Int. Workshop Exploiting Artif. Intell. Techn. Data Manage.*, Jun. 2023, pp. 1–5.

[16] R. Gao, F. Liu, J. Zhang, B. Han, T. Liu, G. Niu, and M. Sugiyama, "Maximum mean discrepancy test is aware of adversarial attacks," 2020, *arXiv:2010.11415*.

[17] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017, *arXiv:1703.00410*.

[18] N. Wang, Y. Chen, Y. Xiao, Y. Hu, W. Lou, and Y. T. Hou, "MANDA: On adversarial example detection for network intrusion detection system," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1139–1153, Mar. 2023.

[19] M. Pawlicki, M. Choraś, and R. Kozik, "Defending network intrusion detection systems against adversarial evasion attacks," *Future Gener. Comput. Syst.*, vol. 110, pp. 148–154, Sep. 2020.

[20] T. Uelwer, F. Michels, and O. D. Candido, "Learning to detect adversarial examples based on class scores," in *Proc. KI Adv. Artif. Intell., 44th German Conf. AI, Virtual Event*. Cham, Switzerland: Springer, Oct. 2021, pp. 233–240.

[21] J. M. Vidal, M. A. S. Monge, and S. M. M. Monterrubio, "EsPADA: Enhanced payload analyzer for malware detection robust against adversarial threats," *Future Gener. Comput. Syst.*, vol. 104, pp. 159–173, Mar. 2020.

[22] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," 2017, *arXiv:1704.01155*.

[23] H. Ye and X. Liu, "Feature autoencoder for detecting adversarial examples," *Int. J. Intell. Syst.*, vol. 37, no. 10, pp. 7459–7477, 2022.

[24] Y. Peng, G. Fu, Y. Luo, J. Hu, B. Li, and Q. Yan, "Detecting adversarial examples for network intrusion detection system with GAN," in *Proc. IEEE 11th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Oct. 2020, pp. 6–10.

[25] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, pp. 79–82, Jun. 2005.

[26] W. Wang and Y. Lu, "Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model," *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, vol. 2018, Aug. 2018, Art. no. 012049.

[27] J.-M. Sánchez-González, C. Rocha-De-Lossada, and D. Flikier, "Median absolute error and interquartile range as criteria of success against the percentage of eyes within a refractive target in IOL surgery," *J. Cataract Refractive Surgery*, vol. 46, no. 10, p. 1441, 2020.

[28] S. C. Hora, "Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management," *Rel. Eng. Syst. Saf.*, vol. 54, nos. 2–3, pp. 217–223, Nov. 1996.

[29] A. D. Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?" *Structural Saf.*, vol. 31, no. 2, pp. 105–112, Mar. 2009.

[30] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Mach. Learn.*, vol. 110, no. 3, pp. 457–506, Mar. 2021.
[31] S. Bjarnadottir, Y. Li, and M. G. Stewart, "Climate adaptation for housing in hurricane regions," in *Climate Adaptation Engineering*, E. Bastidas-Arteaga and M. G. Stewar, Eds., London, U.K.: Butterworth, 2019, doi: 10.1016/B978-0-12-816782-3.00009-7.
[32] W. Zhang, "One step closer to unbiased aleatoric uncertainty estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 15, pp. 16857–16864, doi: 10.1609/aaai.v38i15.29627.
[33] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
[34] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
[35] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
[36] A. Pedraza, O. Deniz, and G. Bueno, "Approaching adversarial example classification with chaos theory," *Entropy*, vol. 22, no. 11, p. 1201, Oct. 2020.
[37] X. Fu, N. Zhou, L. Jiao, H. Li, and J. Zhang, "The robust deep learning–based schemes for intrusion detection in Internet of Things environments," *Ann. Telecommun.*, vol. 76, nos. 5–6, pp. 273–285, Jun. 2021.
[38] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*.
[39] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.
[40] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, vol. 1, 2018, pp. 108–116.
[41] (1999). *KDD Cup 1999*. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[42] E. Değirmenci, Y. S. Kırca, İ. Özçelik, and A. Yazıcı, "ROSIDS23: Network intrusion detection dataset for robot operating system," *Data Brief*, vol. 51, Dec. 2023, Art. no. 109739.
[43] E. Değirmenci. *ROSIDS23 Dataset*. Accessed: Dec. 23, 2024. [Online]. Available: https://zenodo.org/records/10014434

ELİF DEĞİRMENCİ (Member, IEEE) received the B.S. degree from the Department of Computer Engineering, Faculty of Engineering, Anadolu University, Türkiye, in 2016, and the M.S. degree from the Institute of Science, Department of Computer Engineering, Eskisehir Osmangazi University, Türkiye, in 2019, where she is currently pursuing the Ph.D. degree. Her research interests include computer engineering fields, focusing on data science, machine learning, and artificial intelligence applications.

İLKER ÖZÇELİK (Member, IEEE) received the Ph.D. degree in electrical engineering from the Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, USA. He is currently an Assistant Professor with the Department of Software Engineering, Eskisehir Osmangazi University, Merkez/Eskişehir, Türkiye. His research interests include network traffic analysis, network security, software-defined networking, blockchain, and security and privacy in intelligent systems.

AHMET YAZıCı (Member, IEEE) received the M.S. and Ph.D. degrees in control systems from Eskisehir Osmangazi University (ESOGU), in 2000 and 2005, respectively. He was a member of the OSU-ACT Team, Darpa Urban Challenge, in 2007. He has been with the Department of Computer Engineering, ESOGU, since 2005. He is currently the Founder of the Intelligent Factory and Robotic Laboratory. He is also the Founder and the Director of the Center for Intelligent Systems Research (CISAR), ESOGU. His current research interests include verification and validation and autonomous systems.