electronics

Article

TXAI-ADV: Trustworthy XAI for Defending AI Models against Adversarial Attacks in Realistic CIoT

Stephen Ojo [1], Moez Krichen [2,*], Meznah A. Alamro [3] and Alaeddine Mihoub [4]

[1] Department of Electrical and Computer Engineering, College of Engineering Anderson, Anderson University, Anderson, SC 29621, USA; sojo@andersonuniversity.edu
[2] ReDCAD Laboratory, University of Sfax, Sfax 3038, Tunisia
[3] Department of Information Technology, College of Computer and Information Science, Princess Nourah Bint Abdul Rahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; meaalamro@pnu.edu.sa
[4] Department of Management Information Systems, College of Business and Economics, Qassim University, P.O. Box 6640, Buraidah 51452, Saudi Arabia; a.mihoub@qu.edu.sa
* Correspondence: moez.krichen@ieee.org

Abstract: Adversarial attacks are more prevalent in Consumer Internet of Things (CIoT) devices (i.e., smart home devices, cameras, actuators, sensors, and micro-controllers) because of their growing integration into daily activities, which brings attention to their possible shortcomings and usefulness. Keeping protection in the CIoT and countering emerging risks require constant updates and monitoring of these devices. Machine learning (ML), in combination with Explainable Artificial Intelligence (XAI), has become an essential component of the CIoT ecosystem due to its rapid advancement and impressive results across several application domains for attack detection, prevention, mitigation, and providing explanations of such decisions. These attacks exploit and steal sensitive data, disrupt the devices' functionality, or gain unauthorized access to connected networks. This research generates a novel dataset by injecting adversarial attacks into the CICIoT2023 dataset. It presents an adversarial attack detection approach named `TXAI-ADV` that utilizes deep learning (Mutli-Layer Perceptron (MLP) and Deep Neural Network (DNN)) and machine learning classifiers (K-Nearest Neighbor (KNN), Support Vector Classifier (SVC), Gaussian Naive Bayes (GNB), ensemble voting, and Meta Classifier) to detect attacks and avert such situations rapidly in a CIoT. This study utilized Shapley Additive Explanations (SHAP) techniques, an XAI technique, to analyze the average impact of each class feature on the proposed models and select optimal features for the adversarial attacks dataset. The results revealed that, with a 96% accuracy rate, the proposed approach effectively detects adversarial attacks in a CIoT.

Keywords: adversarial attacks; Consumer Internet of Things (CIoT); Explainable Artificial Intelligence (XAI); Shapley Additive Explanation (SHAP); artificial intelligence (AI); machine learning (ML)

check for updates

Citation: Ojo, S.; Krichen, M.; Alamro, M.A.; Mihoub, A. TXAI-ADV: Trustworthy XAI for Defending AI Models against Adversarial Attacks in Realistic CIoT. *Electronics* **2024**, *13*, 1769. https://doi.org/10.3390/electronics13091769

Academic Editor: Dimitris Apostolou

Received: 19 March 2024
Revised: 22 April 2024
Accepted: 1 May 2024
Published: 3 May 2024

1. Introduction

The Consumer Internet of Things (CIoT) is an interconnected network that uses sensors, actuators, and smart devices to collect data, communicate, and perform complicated activities [1,2]. With the progression of technology from web2 (social networking web) to web3 (ubiquitous computing web), the CIoT is becoming the fundamental technology for linking actuators and sensor devices into an integrated network since it extends the reach of the Internet into the physical realm [3,4]. Smart homes, warehouses, automobile networks, environmental monitoring, and perimeter security are examples of CIoT applications [1]. Wireless Sensor Networks (WSNs) are frequently conceived as the basis of CIoT. However, CIoT sensing devices can malfunction [5]. While data transmission between diverse sensing devices/actuators and data servers is necessary, numerous CIoT tools have strict timeliness, security, and stability requirements. Thus, the security and long-term viability of CIoT depend on the hub or servers' being able to sufficiently gather the real data observed by sensors and actuators in a wireless environment.

Since the complexity of CIoT grows rapidly with the integration of new devices, ML has become more crucial in interpreting and learning massive amounts of data provided by CIoT devices [6]. The application of ML to wireless security has seen an increase in research. These include attacks that target spectrum detection [7] and signal categorization [8] activities, as well as spoofing [9], jamming [10], and other attacks on data transfer [11]. Specifically, ML helps CIoT security by identifying devices [12], authenticating signals [13], and detecting anomalies [14]. Given the expanding range of uses for ML, it is critical to comprehend the security risks that affect the technology. Although ML improves the efficiency of CIoT, it also provides attackers with innovative ways of launching potent attacks against CIoT. ML analysis with adversaries is known as adversarial ML [14].

The study of ML's security consequences in the face of adversaries has developed adversarial ML [14,15]. Adversarial ML has traditionally examined a variety of attacks on data domains distinct from wireless communications. An adversary attempting to understand the internal functions of the target ML framework, such as a classifier, conducts an exploration (or inference) attack [10,16]. Evasion attacks are intended to deceive ML systems towards making wrong inferences [10,17]. An adversary can execute a causative (or poisoning) attack to provide erroneous training data so that an ML system can (re)train voluntarily [18]. These attacks can be performed individually or in tandem; for instance, causal and evasion assaults can be carried out by expanding on the inferences obtained from an exploratory attack [19].

To ensure ML systems are trustworthy and secure, researchers continue investigating various tactics for avoiding adversarial attacks, including input preprocessing, model hardening, and robust training techniques. Because ML models depend on pattern recognition and learning by nature, attackers can manipulate or circumvent these systems by taking advantage of this characteristic. Adversarial machine learning attacks must be avoided. When added to ML-based privacy solutions, malicious input data have the potential to alter the ML model itself or produce biased conclusions. Numerous research investigations have examined the influence of adversarial attacks on the performance of ML systems in several domains, including speech recognition [20], natural language processing [21,22], and image processing [23,24]. Most currently available surveys deal with adversarial attacks against ML in conventional network security [25,26] and image recognition [23,27]. However, CIoT protection has yet to give much thought to these attacks. To solve these issues and identify attacks in CIoT devices, the research generated various ML, ensemble voting, and DL models using the injected adversarial attacks CICIoT2023 dataset.

Research Contribution: The primary contributions of the research are listed below.

- This research generates a novel dataset by injecting adversarial attacks into the CICIoT2023 dataset. It presents an adversarial attack detection approach named `TXAI-ADV` that utilizes deep learning and machine learning classifiers to detect attacks and avert such situations rapidly in CIoT.
- This study utilized Shapley Additive Explanations techniques (SHAP), an Explainable Artificial Intelligence (XAI) technique, to analyze the average impact of each class feature on the proposed model and select optimal features for the adversarial attacks dataset.
- The findings show that the proposed method predicts adversarial attacks in the CIoT with 96% accuracy, suggesting the proposed approach for interpretable and accurate attack detection.

Organization: This paper is organized as follows. In Section 2, relevant works and background information are provided. The proposed ML technique and the DL method against adversarial attacks in IoT smart devices are provided in Section 3. The efficacy of the proposed strategy is evaluated and compared to the baseline methods in Section 4. Following the conclusion of this paper, recommendations are provided in Section 5.

2. Literature Review

This section extensively explains IoT technology and how ML and DL techniques can counter adversarial attacks on IoT devices.

Among the various applications of IoT are the Industrial Internet of Things (IIoT), smart cities and homes, medicine, and mobility. The positive aspects of IoT are made possible by the following essential components: (i) middle-ware (data aggregation/fusion hubs and storage systems); (ii) hardware (heterogeneous instruments and controllers); (iii) presentation (representation and additional analysis tools that allow access to a variety of platforms) [2,28]. For many IoT systems, radio frequency identification (RFID) is a significant data source [29]. This technology transfers data and detects objects automatically by using electromagnetic fields. The data can be identified by scanning their labels using the RFID tag. Data transfer for processing and archiving is the next step after sensor data collection. Cloud computing, upon which data analytics are built, is the foundation of IoT storage and computation. Users who have access to cloud computing can visualize the gathered data [30]. Digital replicas of real systems and IoT gadget life cycle management are provided via cloud services [31]. According to [32], the IoT and smart devices have been around for more than ten years, helping to make ubiquitous computing popular and popularize the idea of intelligence, smartness, and the ability to identify, monitor, and control caliber for building and designing smart home devices. Developing smart home automation necessitates considering the IoT. Authors in [19] defined an information network of physical items facilitating interaction and cooperation among these objects to deliver smart home automated services.

Over the past decade, integrating intrusion detection systems (IDSs) and artificial intelligence (AI) in IoT connections has brought an additional layer to technological progress. Based on DL and ML, adversarial perturbations can impact IDS. However, anomaly detection approaches need to be improved by skewed and missing data points, which makes IDS training challenging. The study suggests that by addressing imbalanced data and lacking specific class instances, conditional generative adversarial networks (cGANs) can improve training and potentially avoid our IDS model based on Convolutional Neural Network-Long Short-Term Memory (CNNLSTM). Researchers used the Bot-IoT dataset to assess the proposed IDS model before and after the adversarial training. Positive findings indicated a 40% increase in the detection accuracy of takeover attacks [33]. The study uses data collected by residential intelligent meters to build adversarial examples, demonstrating that their statistical properties match the original data points. Researchers provide this by creating an adversarial white-box attack method. The attack technique focuses on DL-based models identifying appliances in smart home environments. The statistical consistency among the adversarial and actual data points suggests that the task provided by adversarial instances can be beyond the capabilities of non-ML-based solutions [34]. The research introduces IoT Sentinel. The system is capable of automatically identifying the types of devices connected to an IoT network and enforcing regulations to restrict the connectivity of susceptible devices to mitigate the potential harm caused by their incursion [12].

To ensure the dependability of IoHT monitoring software, the research suggests a hybrid ConvLSTM approach that identifies abnormalities and adversarial material in the training set used to create DL models. Additionally, preventative measures are proposed to defend the DL models from adversarial attacks of this kind in the training stage. An empirical assessment using the public PhysioNet dataset shows that the proposed model can detect aberrant values in the frequency of adversarial attacks throughout the training and testing phases. Despite introducing adversarial assaults, the outcomes showed that the model earned a mean F1 score of 97% and an accuracy of 98% [35]. The researchers suggest an adversarial ML-based partial model attack that targets a limited portion of the sensing devices to compromise the IoT's data fusion/aggregation process. The assault is feasible to disrupt data fusion decision making even with limited control over IoT devices, as our numerical results show. For example, the attack success rate increases to 83% when the adversary interferes with eight connected devices [36].

In addition to offering a methodology for a reliable adversarial robustness assessment with a genuine adversarial evasion attack vector, the paper in [37] explains the types of limitations needed for a real adversarial cyber-attack instance. The three supervised algorithms, RF, XGB, and LGBM, as well as the one unsupervised algorithm, isolation forest (IFOR), were assessed using the suggested technique. The adaptive perturbation pattern approach (A2PM) was used to produce constrained adversarial instances, and evasion attacks were conducted on models trained both adversarially and regularly. The acquired results show the positive effects of adversarial training and a security-by-design strategy for a more reliable IoT network intrusion detection and cyber-attack categorization, as well as the inherent vulnerability of tree-based algorithms and ensembles to adversarial evasion attacks. Two new methods for choosing adversarial samples to retrain a classifier are presented in [38]. One is based on the distance from the core of the malware cluster, and the other is based on a probability measure that is obtained via kernel-based learning (KBL). According to our tests, the KBL selection approach increases detection accuracy by 6%, and both sample selection techniques perform better than the random selection method. Furthermore, the study evaluates the influence of such adversarial samples on other classifiers, and the suggested selective adversarial retraining approaches demonstrate comparable enhancements in efficiency for these classifiers as well. In [39], the author analyzes eight distinct adversarial attack approaches that can be acquired as a standard to drive the model to misclassify. The goal of the GEA technique is to carefully embed a benign sample within a malicious one while maintaining the functionality and usefulness of the resulting adversarial sample. Extensive tests are carried out to assess the effectiveness of the suggested approach, demonstrating that commercial adversarial attack techniques can obtain a 100% misclassification rate.

The authors use a recent IoT dataset to examine how adversarial attacks affect DL and basic models. Researchers also suggest an adversarial retraining technique that can greatly enhance IDS effectiveness in the context of adversarial attacks. Simulation results show that while the proposed model can achieve detection accuracy of over 99% across all attacks, including adversarial attacks, including hostile samples dramatically reduces recognition accuracy by over 70% [40]. The study provides significant initiatives toward rendering compressed models robust by thoroughly examining the flaws of compressed audio DNNs. The author suggests a stochastic compression method that produces compressed models that are more resilient to hostile attacks. The study examines two widely used attack algorithms, FGSM and PGD. It presents a comprehensive set of assessments on the adversarial susceptibility and resilience of DNNs in two different audio recognition tasks. According to the proposed study, attack-prone, traditionally trained audio DNNs might have up to 100% mistake rates [41].

Adversarial machine learning has the potential to improve system security of this research [42–46] in a number of ways. Firstly, it strengthens adversarial robustness by protecting models from attacks that tamper with input data in order to misclassify data. Second, recognizing and reducing the hazards connected to re-ID systems and masking identifying characteristics to enhance privacy contributes to privacy protection. Adversarial training also enhances generalization, making models more effective in a variety of scenarios. Finally, it makes adversarial attack detection easier, enabling systems to recognize possibly malicious inputs and take appropriate action. These observations underscore the need for adversarial ML to strengthen the security of person re-identification systems.

The authors of [47] investigate how hostile agents can take advantage of these flaws to compromise deep learning-based network intrusion detection systems (NIDSs). The study found that an attacker may successfully deceive an intended victim of deep learning-based NIDS by changing, on average, just 1.38 of the input attributes of each observed packet. Consequently, the adversarial ML field and its performance from a standard network security perspective need to be carefully considered when constructing such systems.

3. Material and Methods

The proposed methodology offers a systematic approach to problem solving or research, guaranteeing validity, accuracy, and reliability. Figure 1 graphically visualizes the entire process of the proposed methodology. The first step is to collect the "CICIoT2023" original dataset. Following that, before adversarial attacks are introduced into the original dataset, the data undergo preprocessing steps, such as extracting benign traffic from it and splitting it into four data frames. Adversarial instances in machine learning are data points that have been purposefully changed to introduce errors into a model. Data normalization and label encoding have been implemented after the introduction of adversarial attacks. Then, data are split into two sets, training and testing sets, in the data splitting process. The SHAP method determines the mean value of each class's features after the data have been split. The model's efficacy is evaluated on the testing set after it has been trained on the training set. We predict the adversarial attacks in the final phase using various ML, ensemble voting, and deep learning models.

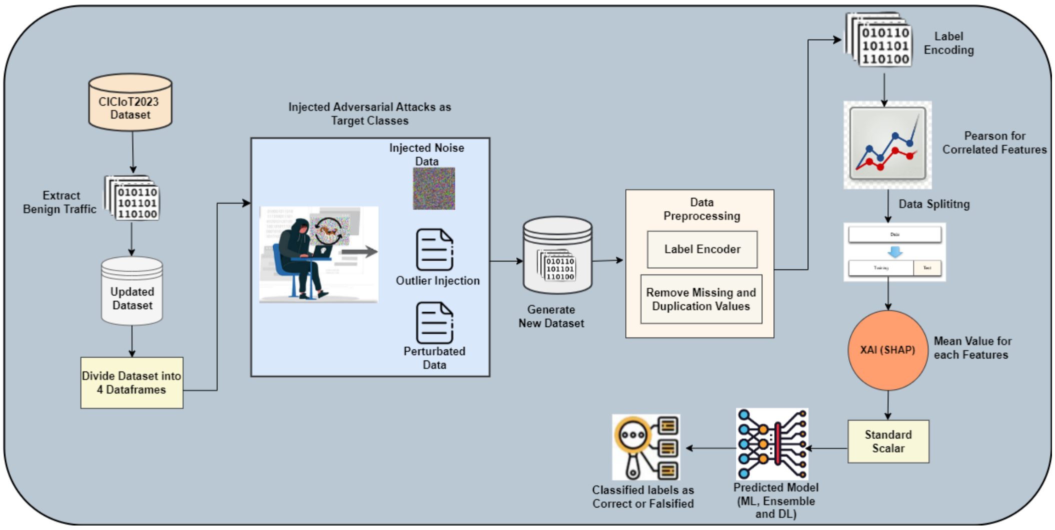


Figure 1. Graphical visualization of proposed methodology.

3.1. *Experimental Dataset Creation*

This study employs the CICIoT2023 dataset, an industry standard for extensive attacks in an IoT context and genuine data collection. Both pcap and CVS file formats are available for the CICIoT2023 dataset. The first data created and gathered within the CIC IoT network under different circumstances are provided by pcap files. These files can build new features and contain all sent packets. Additionally, using CSV files simplifies storing and utilizing the data. The attached fixed-size packet window summarizes the features taken from the original pcap files. In summary, packets that transfer data between two hosts collect the features [48]. Forty-seven features are used in this study's assessment. We combine the values reported in time frames of 100 and 10 packages to minimize varying data sizes. Consequently, the dataset in CSV file format is used in this study. The features of each data block are presented in the resulting CSV datasets. Every attack employed in this study also has distinct characteristics. Malicious IoT devices initiate attacks on susceptible IoT devices. DDoS attacks, for instance, target all devices, but web-based attacks are limited to those that enable web applications [48].

Several adversarial attacks are available in the literature. We use one benign and three adversarial attacks that are explained one by one below.

Benign data: The benign data show that the IoT has a genuine application. Consequently, the primary goal of the data-gathering process hinges on acquiring IoT traffic during idle moments and human interactions (i.e., sensor data, echo dot queries, and smart webcam footage). In this study, we designated the additional columns "label" and assigned them the default value "attack". Next, we execute changes to the 'Label' column for each row in which 'BenignTraffic' is classified as 'Benign'.

Adversarial Noise Injection: This type of attack involves incorporating random noise into the initial data. The purpose of noise injection is to introduce disruptions into the data, increasing the difficulty for methods to classify or forecast accurately. The extra noise could obscure the patterns in the data, leading to erroneous model outputs.

Adversarial Outlier Injection: The original data are enhanced with values through outlier injection. The data point or subset that significantly deviates from the rest of the dataset is called an outlier. Outliers might distort machine learning models and statistical studies because of their unpredictable results. Outlier injection attacks attempt to take advantage of deficiencies in models' sensitivity to abnormalities.

Adversarial Perturbation: Perturbation injection imparts purposeful modifications or disturbances to the original data. One can alter the sign of a feature value, multiply a random factor, and add or remove a minor constant. Attacks such as perturbation injection aim to trick machine learning models by gradually altering the input data in ways that might not be evident initially but can lead to incorrect predictions or classifications.

3.2. Data Preprocessing

Data preprocessing is necessary to prepare raw data for evaluation or modeling. It entails handling anomalies and missing values while cleaning data, combining data from several sources, appropriately formatting data, choosing relevant characteristics, and minimizing dimensionality. Methods, including label encoders and standard scalar techniques, are used to improve data quality while preparing it for further analysis. To train and evaluate the model, the preprocessed data are finally divided into training and test sets. Numerical label input is made possible in an ML model using label encoding. Label encoder uses numbers to assign a value to each label, replacing the values of each label in the dataset. When they have divergent priorities, labels can be employed. This step is crucial in the data preparation process for supervised learning methods [49]. Usually, this technique replaces each value in a category column with a number between 0 and N − 1. In this study, a label encoder assigns a value of 0 to 1 or 2 to each categorical variable. One method for scaling characteristics in ML is called StandardScaler. Through transformation, the data are made to have a distribution with a mean of 0 and a standard deviation of 1. The following represents an equation for standard scaling. Given a feature Y with m data points $X1, X2, \ldots, Xm$, the equation of the standard scalar is

$$y_i = \frac{y_i - \mu}{\sigma} \tag{1}$$

where y_i is the actual attribute value, μ is the mean of the attribute in the dataset, and σ is the standard deviation of the attribute.

3.3. Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence describes methods and approaches that make AI and ML models' decision-making processes understandable and interpretable to people. By investigating how these systems arrive at their conclusions, XAI assists users in validating, trusting, and understanding the predictions and judgments made by AI systems. In AI and ML models, various approaches are utilized to gain explainability. Selecting the inputs that impact model predictions most is known as feature importance. In contrast, local explanations give consumers insights into specific forecasts and help them comprehend specific outcomes. Global explanations offer an expanded view of model behavior, which considers biases, capacities for generalization, and ambiguities. Another strategy, in contrast to intricate black-box models like DNN, is to use inherently interpretable models, like decision trees. This study utilized the SHAP (SHapley Additive exPlanations) to identify the average impact of each feature on model performance.

Shapley Additive Explanations: The SHAP (SHapley Additive Explanations) method emphasizes the importance of particular characteristics, providing insightful information about machine learning model predictions. When used for feature prediction, SHAP calculates each feature's influence on predicting a particular instance. An analysis of SHAP values reveals how features contribute to the expected outcome; positive values denote a positive impact, while negative values signify the opposite. Visualizing SHAP data using various plots can help one better comprehend the importance of each feature and how it affects predictions. Moreover, SHAP considers feature interactions, which are crucial for effective protection against attacks that simultaneously exploit numerous features. Because of its probabilistic interpretation, analysts can quantify each feature's impact on model predictions, enabling them to prioritize protection strategies and assess how susceptible their framework is to hostile exploitation. Furthermore, feature importance and summary charts, two of SHAP's many visualization tools, make complex data on feature relationships and significance easier to interpret. Finally, SHAP offers feature attribution values that show how each feature contributes to the model's predictions. This allows analysts to determine which features are most vulnerable to adversarial manipulation and then reinforce the model's defenses. Figure 2 visualizes the mean value of all class features utilized in this study.

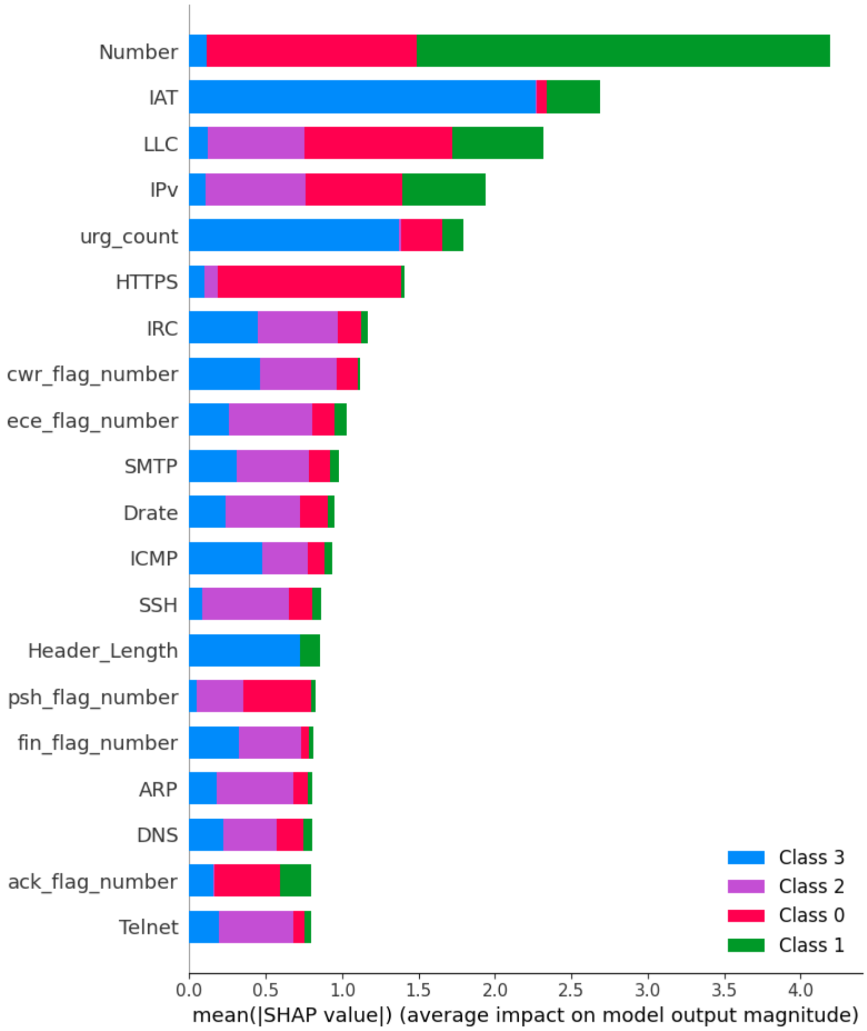


Figure 2. SHAP mean value of all features for each class.

The dataset is divided into test, validation, and training sets to assess the interpretation of ML models. This keeps overfitting from happening and helps assess how generalizable the models are. First, we split the MI dataset into 80% training and 20% testing sets for this study. The mathematical formula used for the training and testing split is as follows:

Training Set Size

$$Z_{train} = round(Z \times train\,ratio), \tag{2}$$

Test Set Size

$$Z_{test} = Z - Z_{train}. \tag{3}$$

The training and testing sizes are shown in Equations (2) and (3). The dataset's total number of occurrences is X. The ratio of instances assigned to the training set is the train_ratio. The ratio of instances assigned to the test set is the test_ratio. These equations provide the dimensions of the training and test sets based on the specified ratios. While the remaining samples implicitly determine the size of the test set after assigning to the training set, adjusting the train_ratio will directly change the size of the training set.

3.4. Experimental Models

Selecting the best statistical model or ML method for a particular dataset or issue involves considering many aspects, such as interpretability, generalization, and predictive accuracy. The diverse ML, ensemble voting, and DL models were used in this study to handle a variety of scenarios, such as noisy, large, and small datasets, along with the ability of weak learning classifiers to boost detection rates.

K-Nearest Neighbor is a simple method for solving classification and regression issues. During training, it keeps track of feature vectors and the labels that correlate with them. In the prediction phase, it calculates the distances to each learning instance, selects the K-nearest neighbors, and sets the majority class as the forecast. The smoothness and noise sensitivity of the decision limit are significantly affected by K selection. KNN is computationally expensive for large datasets, since it is sensitive to feature scaling and necessitates distance computation for each forecast. It can handle sparse or high-dimensional data but performs best on small to medium-sized datasets with few distinct characteristics.

Support Vector Classifier (SVC) is an approach for supervised ML used for classification tasks. It belongs to the Support Vector Machine (SVM) algorithm series. The primary goal of SVC is to find the hyperplane in an N-dimensional space, where N is the total number of features that classify the data points. The following is the equation for selecting a boundary (hyperplane):

$$Y \cdot z + c = 0. \tag{4}$$

where Y is the weight vector (normal to the hyperplane), z is the feature vector, and c is the bias term.

Gaussian Naive Bayes (GNB) is a classification technique based on the premise that features have a Gaussian distribution and the predictors' independence, which is made possible by the Bayes theorem. GNB identifies new data points by calculating probabilities based on Gaussian distribution variables. It also uses prior probabilities and the likelihood of observing the data given to each class to determine the probability of a class given the data. GNB is a starting point for more complex algorithms in text classification and related fields.

Ensemble Voting: An ensemble voting model is an ML method that aggregates the predictions from several separate models to produce a single final forecast. Using the diversity of individual models to increase efficiency, robustness, and generalization capacity is the concept behind ensemble approaches. The study used multiple ML models to design ensemble voting (i.e., classifier) for adversarial attacks.

Multi-Layer Perceptron: A Multi-Layer Perceptron (MLP) is an instance of a feedforward ANN consisting of multiple interconnected layers of nodes. The essential component of an MLP is the perceptron, sometimes referred to as a neuron. It takes in inputs, weighs them, and then produces an output by passing the result via an activation function [50]. The input layer comprises i neurons, the hidden layer of j neurons, and the output layer of k neurons. The input of the network is denoted by $X = [x1, x2, ..., xn.]$, and $y = [y1, y2, ..., yn]$

is the outcome of it. The weights between the input layer and the hidden layer are represented by W, and the weights between the hidden layer and the output layer by V. The following equation can be used to determine the hidden layer's output:

$$y = \sigma(W.x + b), \tag{5}$$

Weight matrices W of size $m.n$ with σ indicate the activation function (sigmoid, tanh, ReLU) and b indicates the bias vector of size m. Likewise, the network's output is determined in this way:

$$x = softmax(V.z + c), \tag{6}$$

Weight matrices of size $k.m$ with $softmax$ indicate the activation function (sigmoid, tanh, ReLU) and c indicates the bias vector of size k. The definition of the softmax function is:

$$softmax(u_i) = \frac{b^{ui}}{\sum^k j = 1^{b^{ui}}}. \tag{7}$$

where u is the k vector length. When training an MLP, the weights and biases are usually adjusted using an optimization algorithm, like gradient descent, to reduce a loss function, which calculates the distinction between the expected and actual outputs. Cross-entropy loss is frequently used as the loss function in classification issues.

Meta Classifier: A "meta classifier" is a more complex classifier that utilizes the forecasts of lower-level classifiers, also called "base classifiers". Instead of classifying instances directly, a meta classifier uses the predictions or outputs from many base classifiers to arrive at the final classification.

Deep Neural Network: According to Dong and Deng [51], the deep learning approach integrates the learning domain that uses organizational frameworks to employ irregular data at many phases. Deep learning combines neural networks, pattern recognition, and graphic design. The deep learning model forecasts well with large datasets [51]. The proposed deep learning technique examines an instance of network stream packets for organic patterns. Furthermore, multi-task training, which considers the attributes of each structure form using a single-layer DNN, logically benefits from deep learning. The majority of the self-learning units in this network have two or more layers. DNN uses hidden units in both the input and output layers. To convert the scalar variable l_q of the next layers to the input x_q below, the hidden unit p can employ a logistic function. In a DNN network, (4) and (5) can predict the output of ith neuron l_i:

$$l_i = t(\xi_i), \tag{8}$$

$$f(\xi_i) = \vartheta + \Sigma_{h\varepsilon\tau_i} - 1N_iX_j, \tag{9}$$

where $f(\xi_i)$ indicates the transfer function and ξ_i is the potential of the ith neuron. The transfer function is shown below:

$$t(\xi_i) = \frac{1}{1 + exp(-\xi_i)}, \tag{10}$$

The total objective cost function can be expressed by the sum of squared errors when goal values are determined by output neurons, l_o, and $\hat{l_o}$.

$$C = \Sigma 1/2(l_o - \hat{l_o}). \tag{11}$$

Algorithm 1 presents a strategy for generating a new dataset with adversarial attacks based on an initial dataset and predicting those infected attacks. The original dataset undergoes a data-cleaning process to prepare it for injecting adversarial attacks. This step might involve handling missing values. The cleaned dataset is divided into four parts to add different types of adversarial attacks. These attacks include adverse benign, noise, outlier, and perturbation. For each type of adversarial attack, a function is called to generate and add the corresponding adversarial instances to the divided parts of the dataset. After adding adversarial attacks to each part of the dataset, a function dataset merge is called to merge all datasets containing adversarial attacks into a new dataset. Finally, the algorithm returns the new dataset containing the data, adds adversarial attacks, and predicts those attacks using ML and DL (DNN and MLP) models. The forward and backward pass complexity of DNN and MLP is $\mathcal{O}(n)$, where n is the total amount of parameters (weights and biases) in the network. This is so because the activation function, which requires a fixed number of operations per parameter, occurs upfront in computing each neuron's output, which is determined by adding the weighted sum of its inputs. The total training complexity of a DNN or MLP is determined by the number of epochs and dataset size, as training these models requires many forward and backward passes (epochs). The entire training complexity is $\mathcal{O}(m.d.n)$, which indicates the size of the dataset as d and the number of epochs as m. To maximize the DL model's real-time performance, techniques such as quantization, hardware acceleration (such as GPUs and TPUs), and model optimization (such as parameter pruning and smaller architectures) require less processing while preserving accuracy. Additionally, using algorithmic improvements, caching strategies, and model parallelism simplifies calculations for effective inference that is customized to meet application requirements in resource-constrained settings.

Algorithm 1 Proposed Algorithm for Defending AI Models Against Adversarial Attacks

```
Require: $D_s$ (Adversarial Dataset)
Pred: $P_{MI}$ (Adversarial attacks)
function Injected Adversarial Attacks $GA$
  $Def_{benign} \rightarrow$ benign
  $Def_{Noise} \rightarrow$ noise injection
  $Def_{Noise} \rightarrow$ outlier injection
  $Def_{Noise} \rightarrow$ data perturbation
return $MergedData$
function Data Preprocessing $D_p$
    $PearsonCorrelated = \frac{\Sigma(xß-\bar{x})(yß-\bar{y})}{\sqrt{\Sigma(xß-\bar{x})^2(yß-\bar{y})^2}}$
    $SHAP \rightarrow$ Mean Feature Value
    $y_{stad} = \frac{y-\mu}{\sigma}$
return $y_{stad}$
$D_{split} \leftarrow N_{train}, N_{test}$
function TrainMLClassifiers
   $PredictingModel \leftarrow$ test_set ($T_s$)
function TrainDLClassifiers
   $Y_{Predict}$ = modtrain.predict($T_d$)
return $Y_{Predict}$
$Y_{Predict} \leftarrow DLmodel$ , $T_d$
$E_m \leftarrow$ Acc, Pre, Recall, F1-score
return Adversarial Attacks $Y_{Predict}$
```

4. Experimental Results and Analysis

Experimental findings and assessment include evaluating and interpreting data from experiments or analyzing data carried out as a component of an investigation. This study assesses the framework's efficacy using extensive evaluation criteria, each offering valuable perspectives on the model's operation. The first metric, accuracy, is typically used as the standard to evaluate performance. It is computed as the part of accurately recognized samples based on the total sample amount. The procedure is made simpler by Equation (12), which emphasizes the measure's simplicity despite its substantial influence.

Accuracy: The accuracy is the ratio of all positive forecasts the model produces to effectively precise projections; it is a crucial assessment metric utilized in performance evaluation. Equation (12) proportionally illustrates this value, making the metric notional equation easier to understand.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Precision: A model or system's precision indicates how it forecasts the positive class. It represents the accuracy of the model and the degree of confidence in its ability to produce good predictions. This value is shown proportionately in Equation (13), facilitating comprehension of the metric basic equation.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

Recall: The ratio of each positive case to the percentage of accurate positive forecasts is the focal point of the assessment metric, termed recall. This balanced viewpoint offers a special benefit while estimating, as the computation of Equation (14) demonstrates.

$$Recall = \frac{TP}{TN + FN} \tag{14}$$

F1-Score: The effectively labeled F1 score may represent the essence of a well-balanced performance, acting as a memory and precision equilibrium. The result of combining these two measures is the F1-score, a widely used performance estimate for models that is particularly helpful in assessments. This basic estimating procedure is thoroughly described by Equation (15), which looks complicated but provides much information.

$$F1\text{-}score = 2 \times \frac{Precision + Recall}{Precision + Recall} \tag{15}$$

MLP Model Outcomes Analysis: The Multi-Layer Perceptron (MLP) model's results have been displayed in a Table 1 for each of the four classes: "Perturbation Injection", "Noise Injection", "Outlier Injection", and "Benign". With a mean F1-score of 0.96, the model obtained an overall accuracy of 0.96%. However, performance varies significantly during the various classes. For instance, the model appears to have a recall of 0.85 for "Perturbation Injection" yet an impeccable rating of 1.00% for "Outlier Injection". This implies that while the model might perform better in detecting outliers, it might have difficulties accurately identifying certain "Perturbation Injection" cases and performed well for benign cases by achieving 0.98% precision, 1.00% recall, and 0.99% f1-score.

Table 1. MLP model results.

Classes	Precision	Recall	F1-Score	Support
Perturbation Injection	0.99	0.85	0.91	10,837
Noise Injection	0.88	0.99	0.93	10,797
Outlier Injection	1.00	1.00	1.00	10,849
Benign	0.98	1.00	0.99	11,109
Weighted Avg.	0.96	0.96	0.96	43,592

KNN Classifier Outcomes Analysis: Table 2 presents the performance results of the KNN Classifier for adversarial attacks. The KNN Classifier achieves a precision of 0.89, a recall of 0.67, and a recall of 0.76 for the perturbation class. For the Noise Injection class, the classifier demonstrates a precision of 0.75, a recall of 0.96, and an F1-score of 0.84. This suggests that the KNN Classifier achieves the best adversarial noise injection class outcomes. The classifiers achieve a precision of 1.00, a recall of 0.92, and an F1-score of 0.84 for the Adversarial Outlier injection class. For the benign class, the classifiers demonstrate relatively high performance with a precision of 0.97, a recall of 1.00, and an

F1-score of 0.98. This indicates that the classifier effectively identifies adversarial benign. The overall accuracy of the KNN Classifier is 0.89, suggesting that 89% of all instances are classified correctly. The macro-average precision, recall, and F1-score are 0.90, 0.89, and 0.89, respectively. The weighted average precision, recall, and F1-score are also 0.90, 0.89, and 0.89, respectively.

Table 2. KNN model results.

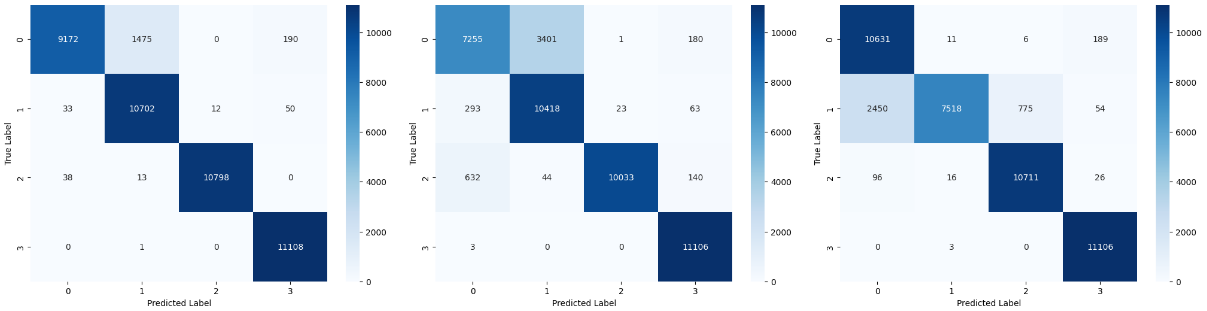
Classes	Precision	Recall	F1-Score	Support
Perturbation Injection	0.89	0.67	0.76	10,837
Noise Injection	0.75	0.96	0.84	10,797
Outlier Injection	1.00	0.92	0.96	10,849
Benign	0.97	1.00	0.98	11,109
Weighted Avg.	0.90	0.89	0.89	43,592

SVC Classifier Results Analysis: Table 3 summarizes the outcome of an SVC model. The SVC model's performance differs depending on all the classes. With a recall of 0.98 and a precision of 0.81, Perturbation Injection exhibited acceptable identification but some incorrect classifications. The F1-score for Noise Injection was balanced at 0.82, with perfect precision and reduced recall at 0.70. With an F1-score of 0.96, Outlier Injection demonstrated strong precision and recall, with scores of 0.93 and 0.99, respectively. Benign cases have been identified with remarkable recall and precision of 0.98 and 1.00, yielding an excellent F1-score of 0.99. The overall model accuracy was 92%, with macro average precision, recall, and F1-score of 0.93, 0.92, and 0.91, respectively, reflecting all classes equally. Furthermore, considering the support of each class, the weighted average precision, recall, and F1-score are, respectively, 0.93, 0.92, and 0.91.

Table 3. SVC model results.

Classes	Precision	Recall	F1-Score	Support
Perturbation Injection	0.81	0.98	0.89	10,837
Noise Injection	1.00	0.70	0.82	10,797
Outlier Injection	0.93	0.99	0.96	10,849
Benign	0.98	1.00	0.99	11,109
Weighted Avg.	0.93	0.92	0.91	43,592

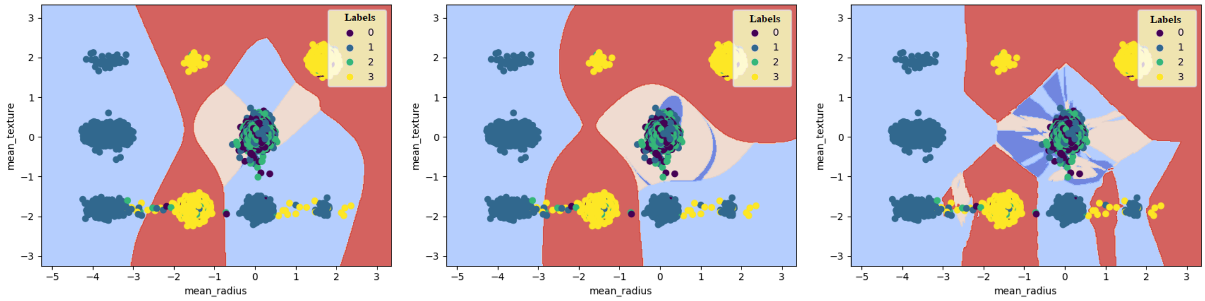
Figure 3 represents the CM of the proposed models (MLP, KNN, and SVC) for adversarial attack detection. Figure 3a demonstrates that the class represents the anticipated proportion labels 0, 1, 2, and 3 for MLP. For class 0, 9172 instances were projected correctly, 10,702 were diagnosed accurately for class 1, and for classes 2 and 3, 10,798 and 11,108 adversarial attack instances were accurately forecasted, respectively. Figure 3b represents the anticipated proportions for the KNN model. A total of 17,255 instances were projected correctly for class 0, 10,418 were protected accurately for class 1, 10,033 instances were forecasted for class 2, and 11,106 instances for class 3 were detected accurately. Similarly, Figure 3c represents the SVC classifier CM; for class 0, 10,631 attack instances were forecasted accurately, 7518 instances for class 1, 10,711 instances for class 2, and 11,106 attacks for class 3 were detected correctly.



(**a**) MLP-CM (**b**) KNN-CM (**c**) SVC-CM

Figure 3. (**a**) Multi-Layer Perceptron model confusion matrix for adversarial attack. (**b**) Confusion matrix of KNN for adversarial attacks in IoT devices. (**c**) Decision Tree confusion matrix for adversarial attacks.

Figure 4 represents the boundary wall of the proposed models (KNN, SVC, and LR) for adversarial attack detection. The feature space is partitioned into regions corresponding to different classes by a decision boundary. An instance of a physical barrier in a spatial context is a wall. A "decision boundary wall" helps understand how these boundaries divide the feature space into distinct class zones by representing them as spatial obstacles.



(**a**) MLP-Feature Space (**b**) SVC-Feature Space (**c**) KNN-Feature Space

Figure 4. (**a**) Decision boundary of MLP for feature space. (**b**) SVC decision boundary for class features. (**c**) Decision boundary of KNN for feature space.

LR Classifier Results Analysis: Table 4 summarizes the outcome of a logistic regression model. According to Perturbation Injection, 31% of real instances were accurately detected, with a precision of 0.30, indicating that only 30% of predictions were true. The recall is similar at 0.31. Recall and precision are not well-balanced, as the F1-score of 0.30 highlights. With precisions of 0.52 and 0.25 and recalls of 0.33 and 0.35, respectively, similar trends are seen for noise and outlier injection, demonstrating poor predictive accuracy. With an accuracy of just 0.46 overall, the model predicted less than half of the outcomes correctly for every class. The weighted average and macro measures demonstrate the model's inadequate effectiveness, where recall and precision averages are less than 0.5. The weighted average and macro metrics, where recall and precision average less than 0.5, further highlight the model's insufficient effectiveness.

Gaussian NB Classifier Results Analysis: The Gaussian NB model's results are presented in Table 5. The Gaussian Naive Bayes (NB) model performed magnificently for each class. It obtained a precision of 0.78 for Perturbation Injection, suggesting that 78% of predictions were right, and a recall of 0.95, meaning that 95% of the real instances of Perturbation Injection were correctly detected. With 10,837 occurrences in the dataset, the F1-score of 0.86 indicates a fair balance between recall and precision for this class. The model demonstrated even greater precision (0.97) in the Noise Injection scenario, but the recall was lower (0.58), yielding an F1-score of 0.73. With 10,797 instances in the dataset, its performance remained reasonably balanced while considering accuracy and recall. Outlier Injection was detected with 10,849 instances in the dataset, yielding an excellent F1-score of 0.95, indicating a solid balance between accuracy and recall, with a precision of 0.92 and a recall of 0.98. Identifying benign occurrences with high precision (0.90) and optimal recall (1.00) yielded an F1-score of 0.94, indicating significant performance. Macro and weighted average metrics verified the model's 88% overall accuracy by indicating satisfactory results across classes.

Table 4. Logistic regression model results.

Classes	Precision	Recall	F1-Score	Support
Perturbation Injection	0.30	0.31	0.30	10,837
Noise Injection	0.52	0.33	0.40	10,797
Outlier Injection	0.25	0.35	0.29	10,849
Benign	0.91	0.86	0.89	11,109
Weighted Avg.	0.50	0.46	0.47	43,592

Table 5. Gaussian NB model results.

Classes	Precision	Recall	F1-Score	Support
Perturbation Injection	0.78	0.95	0.86	10,837
Noise Injection	0.97	0.58	0.73	10,797
Outlier Injection	0.92	0.98	0.95	10,849
Benign	0.90	1.00	0.94	11,109
Weighted Avg.	0.89	0.88	0.87	43,592

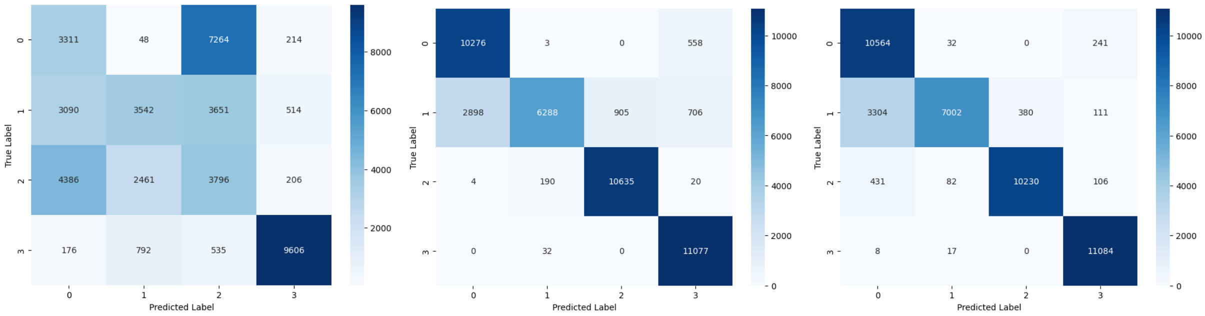
Ensemble Voting Classifier Results Analysis: Table 6 demonstrates the Ensemble Voting model's outstanding performance across multiple classes. It identified 97% of real instances with accuracy for Perturbation Injection, with a precision of 0.74, yielding a balanced F1-score of 0.84. Similarly, the model for Noise Injection has a medium F1-score of 0.78 because of its high precision of 0.98 but low recall of 0.65. With a high F1-score of 0.95, Outlier Injection was detected with a precision of 0.96 and an impressive recall of 0.94. The Benign class stood out for having a high F1 score of 0.98 and exceptional recall and precision. With an overall accuracy of 0.89, the model correctly achieved 89% of its predictions.

Table 6. Ensemble Voting model results.

Classes	Precision	Recall	F1-Score	Support
Perturbation Injection	0.74	0.97	0.84	10,837
Noise Injection	0.98	0.65	0.78	10,797
Outlier Injection	0.96	0.94	0.95	10,849
Benign	0.96	1.00	0.98	11,109
Weighted Avg.	0.91	0.89	0.89	43,592

Figure 5 represents the CM of the proposed models (LR, NB, and Ensemble Voting) for adversarial attack detection. Figure 5a demonstrates that the class represents the anticipated proportions labels 0, 1, 2, and 3 for LR. For class 0, 3311 instances were projected correctly, 3542 were diagnosed accurately for class 1, and for classes 2 and 3, 3796 and

9606 adversarial attack instances were accurately forecasted, respectively. Figure 5b represents the anticipated proportions for the Gaussian NB model. A total of 10,276 instances were projected correctly for class 0, 6288 were protected accurately for class 1, 10,635 instances were forecasted for class 2, and 11,077 instances for class 3 were detected accurately. Similarly, Figure 5c represents the ensemble voting classifier CM; for class 0, 10,560 attack instances were forecasted accurately, 7002 instances for class 1, 10,230 instances for class 2, and 11,084 attacks for class 3 were detected correctly.



(**a**) LR-CM (**b**) Gaussian NB-CM (**c**) Ensemble Voting-CM

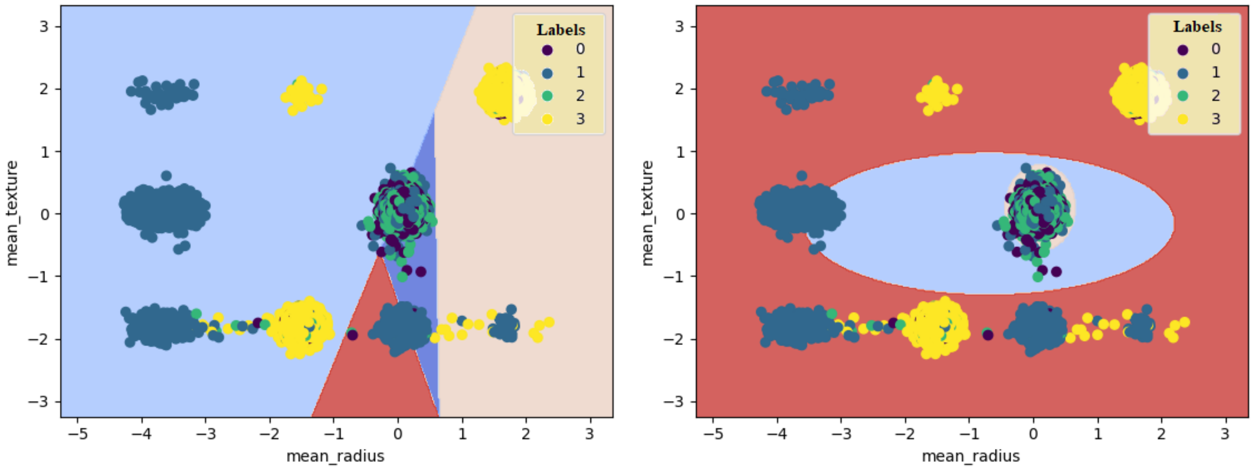
Figure 5. (**a**) CM logistic regression for adversarial attacks. (**b**) Gaussian NB confusion matrix for adversarial attacks in IoT devices. (**c**) Confusion matrix of Ensemble Voting for adversarial attacks in IoT devices.

Meta Classifier Results Analysis: In Table 7, the Meta Classifier performed effectively for each assessed class. It effectively detected 98% of real occurrences with a precision of 0.83 for perturbation injection, yielding a balanced F1-score of 0.90. Despite having a lower recall of 0.75, Noise Injection was anticipated with an extraordinary precision of 0.99, resulting in an F1-score of 0.85, which is impressive. With impeccable recall and great precision (0.96), Outlier Injection was detected, yielding an astounding F1-score of 0.98. The Benign class achieved an excellent F1-score of 0.99, primarily because of their remarkable recall and precision of 1.00 and 0.98, respectively. With an overall accuracy of 0.93, 93% of the predictions made by the model were accurate. The meta classifier model demonstrated significant predictive capabilities, particularly in correctly predicting occurrences across distinct classes, with an impressive overall accuracy of 93%.

Table 7. Meta Classifier model results.

Classes	**Precision**	**Recall**	**F1-Score**	**Support**
Perturbation Injection	0.83	0.98	0.90	10,837
Noise Injection	0.99	0.75	0.85	10,797
Outlier Injection	0.96	1.00	0.98	10,849
Benign	0.98	1.00	0.99	11,109
Weighted Avg.	0.94	0.93	0.93	43,592

Figure 6 represents the boundary wall of the proposed models (LR and Gaussian NB) for adversarial attack detection. A decision boundary divides the feature space into areas that correspond to distinct classes. A wall is an example of a physical barrier in a spatial equivalent. By visualizing these limits as spatial barriers, a “decision boundary wall” helps comprehend how they partition the feature space into discrete class zones.



(**a**) LR-Feature Space.

(**b**) Gaussian NB-Feature Space

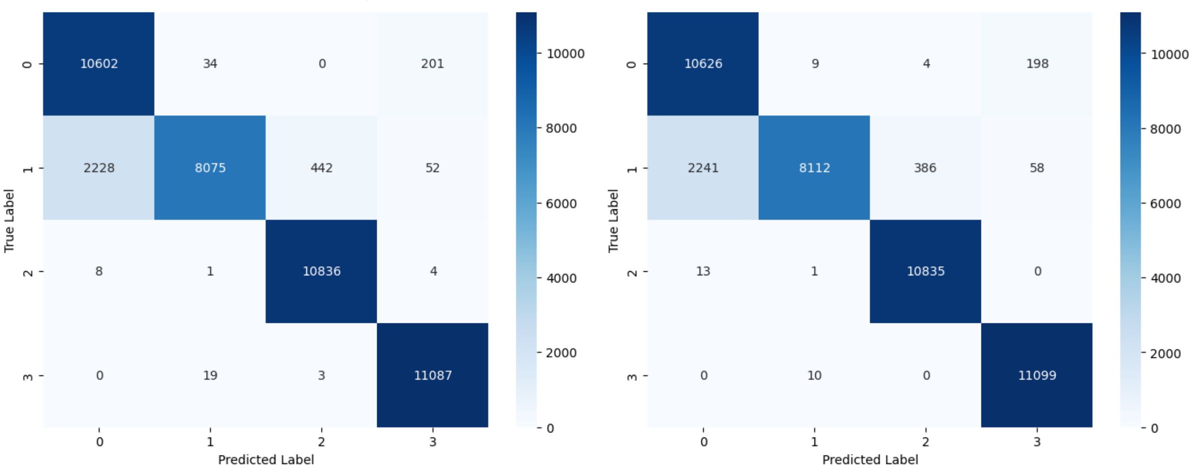
Figure 6. (**a**) LR feature decision boundary. (**b**) Gaussian NB decision boundary for feature space.

DNN Model Results Analysis: Table 8 demonstrates the exceptional performance of the DNN (Deep Neural Network) model in multiple classes. An F1-score of 0.90 was obtained using Perturbation Injection predictions that had a precision of 0.82, and 98% of real instances were properly recognized. The F1-score of 0.86 was obtained from Noise Injection predictions, which displayed an impeccable precision of 1.00 and a 75% recall. A high precision of 0.97 and faultless recall allowed for identifying Outlier Injection, yielding an F1-score of 0.98. With an F1-score of 0.99, benign predictions showed remarkable recall (0.98) and precision (0.98). A total of 93% of the predictions made by the model were accurate, with an accuracy of 0.93. The adequate performance across categories is further demonstrated by macro and weighted average measures, which emphasize strong precision and recall for the Benign, Outlier, and Perturbation Injection classes. As a result, the DNN model achieves an impressive 93% overall accuracy. It is especially noteworthy for its high predictive capabilities in correctly detecting occurrences across several classes.

Table 8. DNN model results.

Classes	Precision	Recall	F1-Score	Support
Perturbation Injection	0.82	0.98	0.90	10,837
Noise Injection	1.00	0.75	0.86	10,797
Outlier Injection	0.97	1.00	0.98	10,849
Benign	0.98	1.00	0.99	11,109
Weighted Avg.	0.94	0.93	0.93	43,592

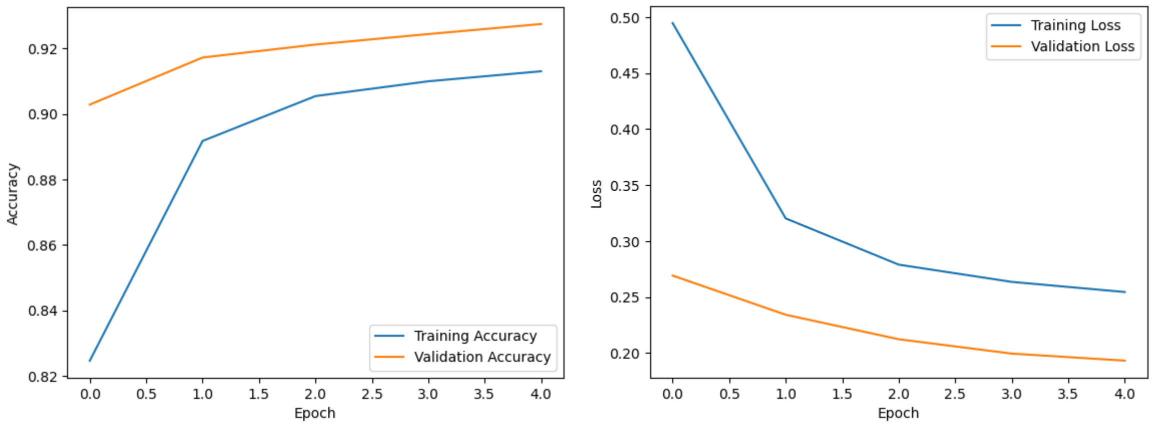
Figure 7 visualizes the CM of the Meta Classifier and DNN model for adversarial attack detection. This visualization provides an elevated summary of how the categorization method works. Larger counts of true positive and true negative values and fewer false positive and false negative outcomes suggest that this strategy performs better. Unlike the diagonal elements, which display accurate predictions, the CM displays instances when the system misclassifies records. The ML methodology performs better than traditional methods in classifying problems regarding accuracy and efficacy.



(**a**) Meta Classifier-CM (**b**) DNN-CM

Figure 7. (**a**) CM of Meta Classifier for adversarial attacks. (**b**) DNN confusion matrix for adversarial attacks in IoT devices.

Figure 8 visualizes the accuracy and loss curves of the DNN model for adversarial attacks in IoT devices. The model's performance on training data can be observed by the Training Accuracy (blue) line, which updates every epoch to improve with more training data; the Validation Accuracy (yellow) line, on the other hand, shows the model's performance on a different validation set, assessing its generalization to new data. Whereas the Validation Loss (yellow) line shows the model's loss on the validation data after each epoch, helping to assess its prediction performance, the Training Loss (blue) line depicts the model's loss on the training data after each epoch, ideally decreasing as training progresses. The performance of several ML classifiers, such as MLP, KNN, SVC, LR, Gaussian NB, Ensemble Voting, and Meta Classifier DNN, is assessed using the bar graph in Figure 9. The graph's bars each show a classifier's score based on a certain metric. For instance, the MLP classifier, which has a score of 0.96, is represented by the first bar under the "Precision" column. With the help of this visual representation, one can compare the performance of several classifiers across the assessed parameters and gain insight into their relative advantages and disadvantages in terms of predicted accuracy.



(**a**) DNN-Accuracy Curve (**b**) DNN-Loss Curve

Figure 8. (**a**) Training and validation accuracy of DNN model. (**b**) Training and validation loss curve of DNN model.

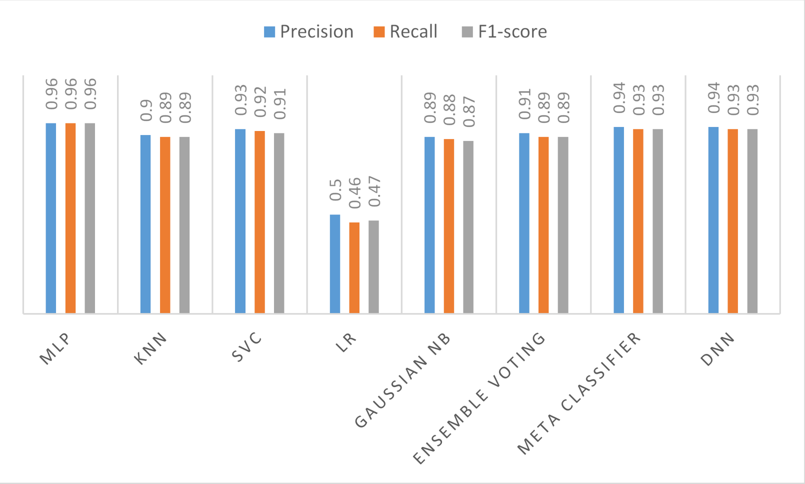


Figure 9. Comparison of proposed model results.

Discussion

Adversarial attacks against IoT technology leverage DL and ML techniques, raising severe security and privacy vulnerabilities. IoT devices, physical objects that are networked and share data over the Internet, use ML to carry out operations, including pattern recognition and decision making. DL, a branch of machine learning, uses neural networks to extract complex patterns from data. Adversarial attacks, which can take various shapes, including data poisoning, evasion attacks, and model extraction, can compromise device security and performance. Some strategies required to protect against vulnerabilities and enhance device resilience amid such attacks include adversarial modeling, model tracking, validation of entries, encrypted communication, and regular update maintenance.

The efficiency of the proposed model is evaluated using optimal indicators needed for statistical analysis. Statistical analysis, which provides quantitative measurements, hypothesis testing, diagnostic resources, and methods for generalization assessment and determining features, is crucial for evaluating, contrasting, and enhancing the efficacy of ML and DL models. It is the cornerstone for making decisions on creating and applying evidence-based models. Adaptability makes mitigation techniques more complicated since attacks can pierce models, and adversarial resilience demands that models be able to endure perturbations and generalize efficiently. In high-dimensional component areas, it is challenging to identify significant adversarial perturbations, and model adaptation is necessary for keeping up with evolving threats. Comprehending correlations and patterns in data and fine-tuning and optimizing the structure of the model are significant signs of a DL model's overall complexity. The DL model's intricacy depends on several architectural elements. The number of parameters in a model rises with its complexity. Complex models are more likely to overfit when inadequately regularized, even though they can capture complex patterns in the data. A model's complexity can be affected by the kind and number of features it uses. Even though adding more features may render a model more intricate, not all characteristics will necessarily significantly impact the model's performance. There are several methods for reducing the complexity of the model, including adding penalty factors and regularizing the loss function. Avoiding extremely complex metrics reduces the likelihood of overfitting. The study uses ML techniques, ensemble voting, and the DNN model to tackle adversarial attacks in IoT smart devices.

In order to make the models more responsive to changes in the input data, this study investigates the vulnerability of compact DNNs to adversarial attacks. These attacks originate from the loss of information during stress. Through testing against a variety of adversarial approaches, the paper examines the robustness, accuracy, and resilience of DNNs. Unpredictable enlargement introduces diversity into the condensed form, making it more difficult for attackers to create effective adversarial instances that target certain model flaws than deterministic approaches, which have the potential to reject data based on predetermined rules. However, more applicability is needed for DL and ML models when confronted with opposing cases. The dependability and credibility of these models are compromised by the fact that perturbations designed for one model frequently translate to another with comparable architectures or even distinct workloads. Research initiatives focused on strengthening model robustness, expanding interpretability, and creating specialized defense mechanisms designed to lessen opposed risks are needed to address these issues. The study intends to strengthen model stability against adversarial attacks by incorporating chaotic features. This will increase uncertainty in the model's input region and hinder attackers' attempts to deceive the model consistently. According to the experiment results, the proposed MLP model against adversarial attack recognition performs more precisely and effectively than traditional methods. The suggested adversarial learning method has the potential to improve IDS defenses against hostile attacks in the CIoT. Compared to conventional techniques like anomaly detection or ML-based systems without adversarial retraining, it can offer better detection performance by precisely focusing on vulnerabilities that adversaries exploit. More precise and consistent detection results can come from the proposed approach. It is also possible that the method was created with practical implementation in mind, considering the particular difficulties and limitations of CIoT spaces. This technique can be a useful addition to the equipment of CIoT security measures, providing practical and effective protection capabilities for real-world deployments by customizing the training procedure to CIoT data and attack instances.

5. Conclusions

The domain of adversarial ML research concerning IoT device security is the focus of this study. We reviewed the literature on the vulnerability of IoT ML-based security models to malicious attacks. Adversarial attacks on IoT smart devices involve exploiting security flaws to gain unauthorized access or disrupt regular activity. Serious effects could result from this, such as privacy violations and personal injury. This research uses ML and DNN models to pinpoint these problems. The research process includes data visualization to find trends in the data before employing ML models to conduct in-depth analysis. The study used novel CICIoT2023 datasets and different statistical characteristics to assess the effectiveness of the ML and DL frameworks by introducing adversarial attacks. The following steps are taken to preprocess the dataset. The data are cleaned, labeled, divided, and normalized using standard scalar techniques; missing values are eliminated; the SHAP technique based on XAI for features of each class impact on model performance; and finally, data are split. The MLP Classifier achieves the highest performance among all models, with a value of 96% accuracy.

However, the study's limitations address some significant areas. A restricted diversity of datasets may initially generate biases and make findings less generalizable across distinct fields or data distributions. Additionally, the study's narrow focus on a selection of adversarial attack methods needs to represent the range of realistic threats adequately. Furthermore, because contemporary ML models are complicated, it remains difficult to fully comprehend and explain model behavior, even with the use of interpretability approaches like SHAP, especially when hostile examples are present. Further limitations on the study's scalability or the investigation of more expansive model architectures can result from the computational resources needed for the implementation and training of models, particularly DNNs. To support the findings of our proposed framework and address the limitation of the study, we want to develop and examine different methodologies in future work. In the future, we will also check the generalizability of newly created dataset-based adversarial attacks by applying different machine learning, deep learning, and feature optimization techniques.

Author Contributions: Conceptualization, S.O. and M.K.; data curation, M.A.A.; formal analysis, S.O. and A.M.; funding acquisition; investigation, M.A.A., M.K., S.O. and A.M.; methodology, M.A.A., M.K., S.O. and A.M.; project administration, M.A.A.; software, S.O.; supervision; validation, M.A.A.; visualization, A.M.; writing—original draft, S.O., M.A.A. and A.M.; writing—review and editing, M.K., S.O., M.A.A. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The CICIoT2023 dataset is publically available.

Acknowledgments: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R503), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khan, W.Z.; Rehman, M.; Zangoti, H.M.; Afzal, M.K.; Armi, N.; Salah, K. Industrial internet of things: Recent advances, enabling technologies and open challenges. *Comput. Electr. Eng.* **2020**, *81*, 106522. [CrossRef]
2. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
3. Celik, Z.B.; Fernandes, E.; Pauley, E.; Tan, G.; McDaniel, P. Program analysis of commodity IoT applications for security and privacy: Challenges and opportunities. *ACM Comput. Surv.* **2019**, *52*, 1–30. . [CrossRef]
4. Fernandes, E.; Jung, J.; Prakash, A. Security analysis of emerging smart home applications. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 23–25 May 2016; pp. 636–654.
5. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
6. Mahdavinejad, M.S.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P.; Sheth, A.P. Machine learning for Internet of Things data analysis: A survey. *Digit. Commun. Netw.* **2018**, *4*, 161–175. [CrossRef]
7. Shi, Y.; Erpek, T.; Sagduyu, Y.E.; Li, J.H. Spectrum data poisoning with adversarial deep learning. In Proceedings of the MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018; pp. 407–412.
8. Sadeghi, M.; Larsson, E.G. Adversarial attacks on deep-learning based radio signal classification. *IEEE Wirel. Commun. Lett.* **2018**, *8*, 213–216. [CrossRef]
9. Shi, Y.; Davaslioglu, K.; Sagduyu, Y.E. Generative adversarial network for wireless signal spoofing. In Proceedings of the ACM Workshop on Wireless Security and Machine Learning, Miami, FL, USA, 15–17 May 2019; pp. 55–60.
10. Shi, Y.; Sagduyu, Y.E.; Erpek, T.; Davaslioglu, K.; Lu, Z.; Li, J.H. Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies. In Proceedings of the 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
11. Erpek, T.; Sagduyu, Y.E.; Shi, Y. Deep learning for launching and mitigating wireless jamming attacks. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *5*, 2–14. [CrossRef]
12. Miettinen, M.; Marchal, S.; Hafeez, I.; Asokan, N.; Sadeghi, A.R.; Tarkoma, S. Iot sentinel: Automated device-type identification for security enforcement in iot. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2177–2184.
13. Ferdowsi, A.; Saad, W. Deep learning for signal authentication and security in massive internet-of-things systems. *IEEE Trans. Commun.* **2018**, *67*, 1371–1387. [CrossRef]
14. Vorobeychik, Y.; Kantarcioglu, M.; Brachman, R.; Stone, P.; Rossi, F. *Adversarial Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 12.
15. Joseph, A.D.; Nelson, B.; Rubinstein, B.I.; Tygar, J. *Adversarial Machine Learning*; Cambridge University Press: Cambridge, UK, 2018.
16. Shi, Y.; Sagduyu, Y.; Grushin, A. How to steal a machine learning classifier with deep learning. In Proceedings of the 2017 IEEE International Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 25–26 April 2017; pp. 1–5.
17. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; Roli, F. Evasion attacks against machine learning at test time. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, 23–27 September 2013; pp. 387–402.
18. Pi, L.; Lu, Z.; Sagduyu, Y.; Chen, S. Defending active learning against adversarial inputs in automated document classification. In Proceedings of the 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Washington, DC, USA, 7–9 December 2016; pp. 257–261.
19. Shi, Y.; Sagduyu, Y.E.; Davaslioglu, K.; Levy, R. Vulnerability detection and analysis in adversarial deep learning. In *Guide to Vulnerability Analysis for Computer Networks and Systems: An Artificial Intelligence Approach*; Springer: Cham, Switzerland, 2018; pp. 211–234.
20. Qin, Y.; Carlini, N.; Cottrell, G.; Goodfellow, I.; Raffel, C. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 5231–5240.

21. Xu, H.; Ma, Y.; Liu, H.C.; Deb, D.; Liu, H.; Tang, J.L.; Jain, A.K. Adversarial attacks and defenses in images, graphs and text: A review. *Int. J. Autom. Comput.* **2020**, *17*, 151–178. [CrossRef]
22. Zhang, W.E.; Sheng, Q.Z.; Alhazmi, A.; Li, C. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.* **2020**, *11*, 1–41. . [CrossRef]
23. Akhtar, N.; Mian, A.; Kardan, N.; Shah, M. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access* **2021**, *9*, 155161–155196. [CrossRef]
24. Naitali, A.; Ridouani, M.; Salahdine, F.; Kaabouch, N. Deepfake attacks: Generation, detection, datasets, challenges, and research directions. *Computers* **2023**, *12*, 216. [CrossRef]
25. Ibitoye, O.; Abou-Khamis, R.; Shehaby, M.e.; Matrawy, A.; Shafiq, M.O. The Threat of Adversarial Attacks on Machine Learning in Network Security—A Survey. *arXiv* **2019**, arXiv:1911.02621.
26. Jmila, H.; Khedher, M.I. Adversarial machine learning for network intrusion detection: A comparative study. *Comput. Netw.* **2022**, *214*, 109073. [CrossRef]
27. Akhtar, N.; Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **2018**, *6*, 14410–14430. [CrossRef]
28. Barriga, J.K.D.; Romero, C.D.G.; Molano, J.I.R. Proposal of a standard architecture of IOT for Smart Cities. In Proceedings of the Learning Technology for Education in Cloud—The Changing Face of Education: 5th International Workshop, LTEC 2016, Hagen, Germany, 25–28 July 2016; pp. 77–89.
29. Stergiou, C.L.; Plageras, A.P.; Psannis, K.E.; Gupta, B.B. Secure machine learning scenario from big data in cloud computing via internet of things network. In *Handbook of Computer Networks and Cyber Security: Principles and Paradigms*; Springer: Cham, Switzerland, 2020; pp. 525–554.
30. Stergiou, C.; Psannis, K.E. Recent advances delivered by mobile cloud computing and internet of things for big data applications: A survey. *Int. J. Netw. Manag.* **2017**, *27*, e1930. [CrossRef]
31. Firouzi, F.; Farahani, B.; Ye, F.; Barzegari, M. Machine learning for IoT. In *Intelligent Internet of Things: From Device to Fog and Cloud*; Springer: Cham, Switzerland, 2020; pp. 243–313.
32. Madakam, S. Internet of things: Smart things. *Int. J. Future Comput. Commun.* **2015**, *4*, 250. [CrossRef]
33. Benaddi, H.; Jouhari, M.; Ibrahimi, K.; Benslimane, A.; Amhoud, E.M. Adversarial Attacks Against IoT Networks using Conditional GAN based Learning. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 2788–2793.
34. Singh, A.; Sikdar, B. Adversarial attack for deep learning based IoT appliance classification techniques. In Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 14 June–31 July 2021; pp. 657–662.
35. Albattah, A.; Rassam, M.A. Detection of Adversarial Attacks against the Hybrid Convolutional Long Short-Term Memory Deep Learning Technique for Healthcare Monitoring Applications. *Appl. Sci.* **2023**, *13*, 6807. [CrossRef]
36. Luo, Z.; Zhao, S.; Lu, Z.; Sagduyu, Y.E.; Xu, J. Adversarial machine learning based partial-model attack in IoT. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, Virtual, 13 July 2020; pp. 13–18.
37. Vitorino, J.; Praça, I.; Maia, E. Towards adversarial realism and robust learning for IoT intrusion detection and classification. *Ann. Telecommun.* **2023**, *78*, 401–412. [CrossRef]
38. Khoda, M.E.; Imam, T.; Kamruzzaman, J.; Gondal, I.; Rahman, A. Robust malware defense in industrial IoT applications using machine learning with selective adversarial samples. *IEEE Trans. Ind. Appl.* **2019**, *56*, 4415–4424. [CrossRef]
39. Abusnaina, A.; Khormali, A.; Alasmary, H.; Park, J.; Anwar, A.; Mohaisen, A. Adversarial learning attacks on graph-based IoT malware detection systems. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; pp. 1296–1305.
40. Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Wibowo, S.; Gordon, S.; Fortino, G. Adversarial training for deep learning-based cyberattack detection in IoT-based smart city applications. *Comput. Secur.* **2022**, *120*, 102783. [CrossRef]
41. Bhattacharya, S.; Manousakas, D.; Ramos, A.G.C.; Venieris, S.I.; Lane, N.D.; Mascolo, C. Countering acoustic adversarial attacks in microphone-equipped smart home devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2020**, *4*, 1–24. . [CrossRef]
42. Yu, Z.; Li, L.; Xie, J.; Wang, C.; Li, W.; Ning, X. Pedestrian 3D Shape Understanding for Person Re-Identification via Multi-View Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2024**. . [CrossRef]
43. Ning, E.; Wang, Y.; Wang, C.; Zhang, H.; Ning, X. Enhancement, integration, expansion: Activating representation of detailed features for occluded person re-identification. *Neural Netw.* **2024**, *169*, 532–541. [CrossRef]
44. Wang, C.; Ning, X.; Li, W.; Bai, X.; Gao, X. 3D person re-identification based on global semantic guidance and local feature aggregation. *IEEE Trans. Circuits Syst. Video Technol.* **2023**. . [CrossRef]
45. Ning, E.; Wang, C.; Zhang, H.; Ning, X.; Tiwari, P. Occluded person re-identification with deep learning: A survey and perspectives. *Expert Syst. Appl.* **2023**, *239*, 122419. [CrossRef]
46. Ning, E.; Zhang, C.; Wang, C.; Ning, X.; Chen, H.; Bai, X. Pedestrian Re-ID based on feature consistency and contrast enhancement. *Displays* **2023**, *79*, 102467. [CrossRef]
47. Clements, J.; Yang, Y.; Sharma, A.A.; Hu, H.; Lao, Y. Rallying adversarial techniques against deep learning for network security. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021; pp. 1–8.

48. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors* **2023**, *23*, 5941. [CrossRef] [PubMed]
49. Sharma, N.; Bhandari, H.V.; Yadav, N.S.; Shroff, H. Optimization of IDS using Filter-Based Feature Selection and Machine Learning Algorithms. *Int. J. Innov. Technol. Explor. Eng* **2020**, *10*, 96–102. [CrossRef]
50. Popescu, M.C.; Balas, V.E.; Perescu-Popescu, L.; Mastorakis, N. Multilayer perceptron and neural networks. *WSEAS Trans. Circuits Syst.* **2009**, *8*, 579–588.
51. Deng, L.; Yu, D. Foundations and Trends in Signal Processing: DEEP LEARNING—Methods and Applications. *Now Found. Trends* **2014**, *2014*, 206.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.