information

Article

A Deep Learning Methodology for Predicting Cybersecurity Attacks on the Internet of Things

Omar Azib Alkhudaydi [1], Moez Krichen [1,2,*] and Ans D. Alghamdi [1]

[1] Faculty of Computer Science and Information Technology, Al-Baha University, Al-Baha 65779, Saudi Arabia; 443040465@stu.bu.edu.sa (O.A.A.); ans@bu.edu.sa (A.D.A.)
[2] ReDCAD Laboratory, National School of Engineers of Sfax, University of Sfax, Sfax 3099, Tunisia
* Correspondence: dr.moez.krichen@redcad.org

Abstract: With the increasing severity and frequency of cyberattacks, the rapid expansion of smart objects intensifies cybersecurity threats. The vast communication traffic data between Internet of Things (IoT) devices presents a considerable challenge in defending these devices from potential security breaches, further exacerbated by the presence of unbalanced network traffic data. AI technologies, especially machine and deep learning, have shown promise in detecting and addressing these security threats targeting IoT networks. In this study, we initially leverage machine and deep learning algorithms for the precise extraction of essential features from a realistic-network-traffic BoT-IoT dataset. Subsequently, we assess the efficacy of ten distinct machine learning models in detecting malware. Our analysis includes two single classifiers (KNN and SVM), eight ensemble classifiers (e.g., Random Forest, Extra Trees, AdaBoost, LGBM), and four deep learning architectures (LSTM, GRU, RNN). We also evaluate the performance enhancement of these models when integrated with the SMOTE (Synthetic Minority Over-sampling Technique) algorithm to counteract imbalanced data. Notably, the CatBoost and XGBoost classifiers achieved remarkable accuracy rates of 98.19% and 98.50%, respectively. Our findings offer insights into the potential of the ML and DL techniques, in conjunction with balancing algorithms such as SMOTE, to effectively identify IoT network intrusions.

Keywords: cybersecurity; DoS; DDoS; IoT; machine learning; deep learning; Bot-IoT dataset

Citation: Alkhudaydi, O.A.; Krichen, M.; Alghamdi, A.D. A Deep Learning Methodology for Predicting Cybersecurity Attacks on the Internet of Things. *Information* **2023**, *14*, 550. https://doi.org/10.3390/info14100550

Academic Editor: Sherali Zeadally

Received: 9 September 2023
Revised: 25 September 2023
Accepted: 2 October 2023
Published: 7 October 2023

1. Introduction

The Internet of Things (IoT) is the linking of materially moving items implanted with intelligent machine, sensing, and other equipment and connected to the internet [1]. The IoT interconnects systems, apps, data storage, and services, which may serve as new entry points for cyberattacks as the IoT continually provides services inside an enterprise [2]. Furthermore, to maintain the security of IoT systems, continual surveillance and evaluation are required. Predicting kinds of attacks is essential for the defence analysis and tracking of IoT devices [3,4]. This allows for the adaptation of unanticipated circumstances, the taking of safety measures, the protection of data, the provision of stability, and the minimization of various risks. Current attack prediction technologies are unable to keep up with the massive number and variety of attacks; therefore, this remains a challenge for continuous study. Due to their good performance in a range of prediction-based fields, researchers have recently focused on machine learning (ML) methodologies, especially deep learning (DL) techniques [5,6].

In the context of the IoT [7,8], the use of artificial intelligence (AI) algorithms such as ML and DL algorithms may offer an efficient method for making use of data to forecast and identify potential cybersecurity threats [9]. The deep learning approach, as a strategy for identifying cyberattacks, is growing in popularity more rapidly than previous methods, which enables more efficient mitigation [10]. Deep learning is a subcategory of AI that concentrates on the processing of computing machine applications that can recognize

complex, nonlinear patterns and then utilize those patterns to create predictions [11,12]. In the world of cybersecurity, deep learning models are becoming an increasingly popular tool, and they are quickly becoming an essential component of effective defense strategies against harmful assaults [13,14]. This technology can detect, react to, and prevent a broad variety of assaults that are launched against linked items, such as the IoT [15]. Since IoT devices have become more networked, the likelihood of hacks has increased. Deep learning may be applied to aid in the detection of harmful assaults on connected devices, the mitigation of these risks, and the proactive prevention of future attacks [16].

The main contributions of this work are as follows:

- We propose an AI model-based DL and different machine and ensemble learning classifiers to detect cyber-attacks on the IoT with SMOTE (Synthetic Minority Oversampling Technique) implementation to yield significant results [17];
- We improve the accuracy and confidence of cybersecurity attack detection in IoT environments compared to current works;
- We produce more accurate and reliable predictions, leading to improved IoT security by preventing unauthorized access, data breaches, and service interruptions;
- We enhance the generalization capabilities of the developed models by addressing the class imbalance issues commonly observed in IoT cybersecurity datasets through the application of SMOTE [18];
- We bring an understanding of the optimal application of DL and ensemble learning models as cybersecurity attack prediction classifiers.

This work will contribute to knowledge in the fields of cybersecurity and IoT security by investigating the performance and efficacy of various ensemble learning techniques in conjunction with deep learning models. These insights can assist practitioners and researchers in developing more robust and efficient security solutions for Internet of Things (IoT) systems, thereby increasing their resilience against emergent cyberthreats. This study has the potential to advance current efforts in cybersecurity attack prediction with respect to the Internet of Things. In addition, this research has the potential to considerably enhance the safeguarding posture of the IoT, protect critical data and services from malicious attacks, and facilitate the development of more resilient and secure IoT infrastructures.

2. Literature Review

The authors of [19] proposed an ML method for malware detection in IoT networks that does not need feature engineering. Their suggested methodology significantly speeds up the IoT edge with minimal power consumption. FEL-ML provided resource-sensitive internet traffic protection with the extra advantage of avoiding unnecessary the substantial efforts of subject material experts in feature engineering.

Because of the unreliability inherent in IoT systems, such as in the dynamic communication that might occur between different IoT devices, these systems have several security flaws. Following from this, the authors of [20] suggested merging three DL algorithms, namely, the RNN (Recurrent Neural Network), LSTM-RNN (Long Short-Term Memory-RNN), and CNN (Convolutional Neural Network), to construct a bidirectional CNN-BiLSTM (Bidirectional Long Short-Term Memory) DDoS (Distributed Denial of Service) detection model [21]. The RNN, CNN, LSTM, and CNN-BiLSTM models are put through their paces to establish which one is the most effective model in defending against DDoS assaults and being able to effectively identify and differentiate DDoS traffic from normal traffic. The CICIDS2017 is used to give detection that is more realistic. A rate of accuracy of 99.00% was acquired from the performance of the models, except for the CNN model, which obtained a rate of 98.82% accuracy. The accuracy of the CNN-BiLSTM was measured at 99.76%, while its precision was measured at 98.90%.

Ref. [22] proposes a heuristic distributed scheme (HIDE) to address the challenges of validating the mobility pattern of autonomous vehicles in the Internet of Vehicles (IoV). HIDE penalizes or rewards vehicles based on the conformity of their claimed mobility patterns, using a time-homogeneous semi-Markov process (THSMP) to predict pattern

accuracy. Results demonstrate that HIDE effectively identifies malicious vehicles and assigns a lower weight of impact to them compared to normal vehicles, improving the performance of traffic management systems.

In [23], a DL model was implemented to forecast the most prevalent cybersecurity assaults. The assessment metrics of the suggested SET-based model were evaluated, and the suggested model attained an efficacy of 0.99% with a test duration of time 2.29 ms. In this study [24], the authors presented an exploratory study of federated DL using several DL techniques. They examined the efficacy of three IoT traffic databases in ensuring the privacy of IoT systems data and enhancing the accuracy of DL-based attack detection.

In [25], the authors suggested FDL for the detection of zero-day attacks to prevent data privacy leaks in IoT edge devices. Utilizing an ideal DNN architecture, this approach classifies network traffic. A parametric server remotely coordinates the separate training of DNN models in many IoT edge devices, while the Federated Averaging (FedAvg) method aggregates local model updates. After a series of communication cycles between the parametric server and the IoT edge devices, a global DNN model was generated.

In [26], the authors suggested employing the encoding phase of the LSTM Autoencoder to decrease the feature dimensionality of large-scale IoT network traffic data (LAE). According to the findings, 91.89% less memory was required to store large-scale network traffic data due to LAE. To reduce the potential threats posed by IoT devices, it is now very necessary to ensure that DDoS is quickly identified.

A Local–Global best Bat Algorithm for Neural Networks (LGBA-NN) was presented in [27] to choose both selected features and hyperparameters for the purpose of the effective detection of botnet assaults, which were inferred from nine commercial IoT systems. The updated bat velocity in the swarm was calculated using the local–global best-based inertia weight, which was accepted by the Bat Algorithm (BA) that was developed. To address the issue of BA swarm diversity, they suggested employing a Gaussian distribution for population initialization. In addition, the local searching strategy was executed by the Gaussian density function and the local–global best function to improve exploration in each iteration. This action was taken to achieve the greatest outcomes. The neural network hyperparameters and weights were then optimized using an enhanced Bayesian analysis to classify 10 distinct botnet attacks and one benign target class. An N-BaIoT dataset consisting of substantial actual traffic data was used to evaluate the proposed LGBA-NN method. This dataset included both benign and malicious target classes. The effectiveness of LGBA-NN was evaluated in comparison to that of several recently developed advanced methods, including weight optimization by means of particle swarm optimization (PSO-NN) and BA-NN. The findings of the experiments showed that LGBA-NN is better than BA-NN and PSO-NN in the identification of multi-class botnet attacks. LGBA-NN achieved an accuracy of 90%.

Ref. [28] presents a unique hybrid deep random NN (HDRaNN) for the detection of cyberattacks in the Industrial Internet of Things (IIoT). The Deep Dropout Regularized Random Neural Network (HDRaNN) is a hybrid model that combines a Deep Random Neural Network with a multilayer perceptron. The suggested approach was assessed using two datasets that are linked to IIoT security. Different performance measures were used to conduct the performance analysis for the suggested plan. The HDRaNN was able to classify a total of sixteen distinct categories of cyberattacks with an accuracy of 0.98 to 0.99.

In [29], the authors proposed the RPL loophole attack, which targets the commonly used IPv6 routing protocol in IoT-based systems. A security technique based on ML was described. The evaluation of the gathered data revealed that the machine learning-based algorithms identified the loophole attack correctly.

To overcome the difficulties associated with protecting IoT networks, which are amplified by the volume and variety of deployments and the rapidly changing environment of cyber threats, the authors of [30] developed a technique that makes use of powerful deep learning to identify cyber assaults that are directed against IoT equipment. Their method involves incorporating LSTM algorithm into an existing solution; then, a decision

tree is used to bring together these individual modules, so that an aggregated result can be produced. They achieved an accuracy rate of over 99% when it comes to the identification of cyber threats against IoT devices by evaluating the efficacy of their technique using a Modbus dataset.

The authors of [31] provided a model based on a variety of ML techniques. On the BoT-IoT dataset, the KNN, Naive Bayes, and MLP ANN models were used to build a model. Using an initial number according to efficiency and the ROC AUC result, the optimal algorithm was determined. Incorporating machine learning methods with feature engineering and oversampling methodology (SMOTE), the performances of three algorithms were evaluated on class-imbalanced and class-balanced datasets.

Using machine learning approaches, many cybersecurity threats were anticipated in [32]. A novel predictive system based on Random Neural Networks (RaNN) was developed. Several assessment parameters were developed to be tested with the ANN, SVM, and decision tree to determine the accuracy of the RaNN-based predictive model. According to the evaluation results, the proposed RaNN model achieved an efficiency of 99.20%, with a learning rate of 0.01 and a time length of 34.51 milliseconds.

3. Materials and Methods

This work presents an automated network detection model for the Internet of Things. Our proposed model gathers sensor-collected flow data, which are subsequently transmitted to feature engineering algorithm techniques. It will utilize feature engineering techniques such as feature selection and feature imbalance. Feature selection techniques, such as Recursive Feature Elimination and Principal Component Analysis, can overcome numerous data problems, such as lowering overfitting, training time, and enhancing the overall model accuracy. In [31], the authors used the SMOTE approach for balancing the provided data to address a class imbalance to their model. Several deep learning models will be executed to determine the performance and time complexity of each unique model.

3.1. Bot-IoT Dataset

A new development dataset Bot-IoT is used for the purpose of simulated assault identification in the experiment using the IoT network [33]. The collection includes data from the Internet of Things collected from Cyber Range Lab of UNSW Canberra, as well as ordinary traffic flows and traffic flows caused by botnets because of various types of attacks [34].

A realistic testbed was used to create a valuable dataset with comprehensive traffic information. Additional features were added and labeled to improve the machine learning models' performance. Three subcomponents contributed to the extraction of characteristics: simulated IoT services, networking structure, and investigative analyses. The IoT system can gather real-time meteorological data and utilize them to adjust settings. A smart cooling fridge communicates cooling and temperature details, while a smart device manages lighting. These lights function as motion detectors and turn on automatically when motion is detected. The list also includes an IoT smart door with probabilistic input and an intelligent thermostat that can adjust the temperature autonomously. Table 1 describes the attack characteristics for the data.

Targets in an IoT system help differentiate network traffic into benign or malicious activities, making it easier to distinguish between harmless and dangerous actions. The following are the target categories in the BoT-IoT dataset:

- Benign category: normal, legitimate IoT network activity without malicious intent; DDoS TCP attacks flood a network with TCP requests, rendering it inaccessible to authorized users;
- UDP-focused DDoS attacks: these flood networks with packets, causing disruptions and service outages;
- DDoS HTTP attacks: these flood web servers with HTTP requests, causing degraded performance or service disruption;

- TCP DoS attacks: these exploit TCP stack vulnerabilities to exhaust device/network resources and render them unresponsive/unavailable;
- UDP DoS attacks: these flood the target with many packets, leading to resource exhaustion and service disruptions;
- HTTP-based DoS attacks: these overload web servers with excessive requests, causing degraded performance or unavailability;
- Keylogging: the covert monitoring and recording of keystrokes on a compromised device, used for malicious purposes to steal sensitive information;
- Capture of data: the unauthorized capture and exfiltration of information from compromised IoT networks/devices.

Table 1. Bot-IoT dataset.

Type	Target	Count
BENIGN	Benign	9543
DDoS TCP	Attack	19,547,603
DDoS UDP	Attack	18,965,106
DDoS HTTP	Attack	19,771
DoS TCP	Attack	12,315,997
DoS UDP	Attack	20,659,491
DoS HTTP	Attack	29,706
Keylogging	Keylogging	1469
Data theft	Data theft	118
Total	-	73,370,443

3.2. The Proposed Model

3.2.1. Data Pre-Processing

The pre-processing of data is an essential component of model development. We applied the following pre-processing techniques to enhance the proposed model during this procedure. In the pre-processing phase, data cleansing comprises data filtration, the conversion of data, and checking for missing data. In the data filtration phase, null and duplicate values are obtained and eliminated. In the data transformation procedure, the data are converted into the appropriate format, such as from categorial to a numerical. Various Python utilities help prepare data for analysis by cleaning it [35–37].

3.2.2. Feature Engineering Techniques

1. Correlation Coefficient

The correlation coefficient measures the relationship between two factors in a given dataset. In the BoT-IoT dataset, analyzing the correlation coefficient can provide valuable insights into the interdependencies and associations between different variables. Thereby enhancing comprehension of the dataset and its potential patterns, as seen in Figure 1.

The BoT-IoT dataset contains information about IoT devices that have been infiltrated by botnets, which are networks of infected devices controlled by malicious actors. This dataset contains a variety of attributes and characteristics describing the behavior and characteristics of compromised IoT devices, ensuring that the dataset is formatted properly and that any missing values or outliers are handled appropriately. The data must be pre-processed to ensure accurate and reliable results.

In specifying the BoT-IoT dataset variables for which the correlation coefficient is to be computed, these variables may include device type, communication protocols, network traffic patterns, and any other pertinent factors that may be present in the dataset. We obtain the correlation coefficient using an appropriate statistical method once the variables have

been selected. We identify both the magnitude and the direction of the connection between the variables by analyzing the computed correlation coefficient. A high level of correlation shows that as one factor rises, the other is usually increasing as well, while a single factor rising and the other factor tending to go down is indicative of a negative correlation. A correlation coefficient near 0 indicates a non-existent relationship between the variables. The correlation coefficient quantifies the relationship between two variables within a dataset. In the BoT-IoT dataset, analyzing the correlation coefficient can provide valuable insights into the interdependencies between different variables, thereby enhancing performance of the dataset and its potential patterns. The BoT-IoT dataset contains information about IoT devices that have been compromised by botnets, which are networks of infected devices controlled by malicious actors. This dataset contains a variety of attributes and characteristics that describe the behavior and characteristics of compromised IoT devices.

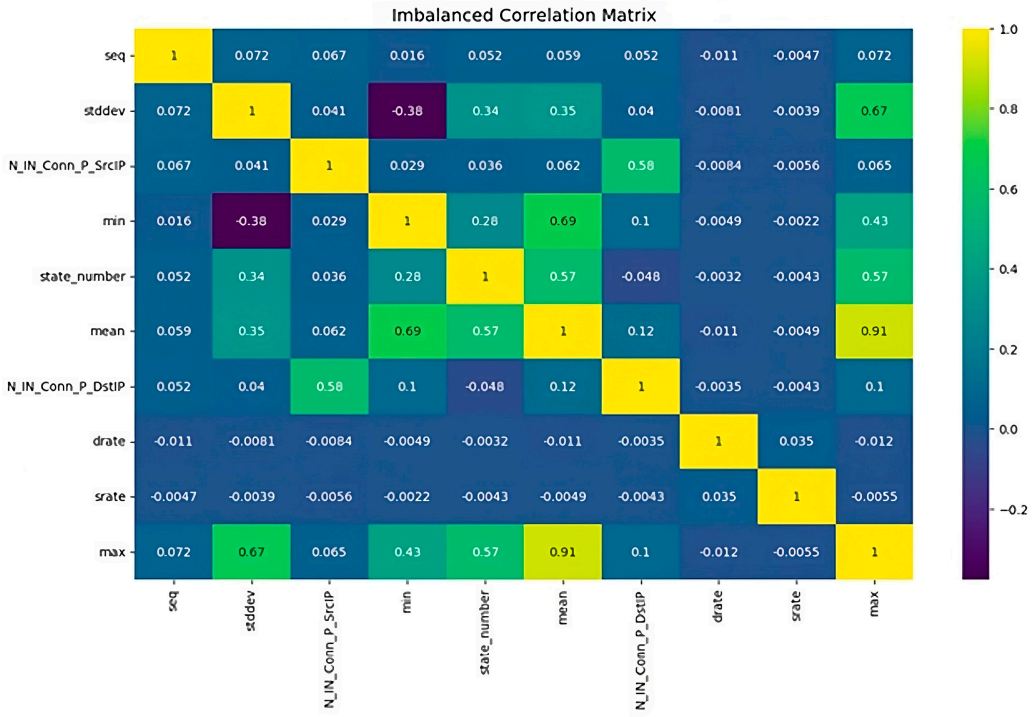

Figure 1. A correlation coefficient between features of the BoT-IoT dataset.

2. Feature Importance using Random Forest

Feature importance analysis utilizing Random Forest is an effective method for determining the significance of various features of the BoT-IoT dataset. This analysis reveals which characteristics have the greatest impact on the dependent variable. Thereby facilitating the identification of critical factors that contribute to the behavior of the attack that compromised the IoT devices, as shown in Figure 2.

When applying feature importance to the BoT-IoT dataset using Random Forest, the BoT-IoT dataset is divided into subsets for training and testing. The training subset will be used to construct the Random Forest model, while the testing subset will assess the models' performance and generalizability. The Random Forest is capable of handling high-dimensional datasets and provides an intrinsic measure of feature significance. Using the Random Forest model, we calculate the feature importance. This can be accomplished by investigating the mean decrease impurity or Gini importance, which measures the degree to which each feature reduces the impurity or variability in the target variable across the Random Forest's decision trees. Alternately, feature importance can be evaluated using permutation importance or mean decrease accuracy. The calculated feature importance scores can be visualized using techniques such as bar charts and heat maps. This facilitates interpretation of the results and provides a clear comprehension of which Botnet IoT

dataset features are most influential. It is essential to note that interpretation must be based on domain-specific knowledge and the characteristics of the dataset. Using feature importance analysis with Random Forest on the BoT-IoT dataset permits the identification of important characteristics that influence the behavior of harmed IoT devices. Based on feature importance analysis using Random Forest, the attributes 'pkSeqID', 'proto', 'saddr', 'sport', 'daddr', 'dport', and 'category', which have low significant features in the BoT-IoT dataset, were dropped. In considering the feature importance analysis using Random Forest, including these attributes in machine learning models decreases the accuracy and effectiveness of the predictive models, feature selection techniques, and exploratory analyses applied to the Botnet IoT dataset.

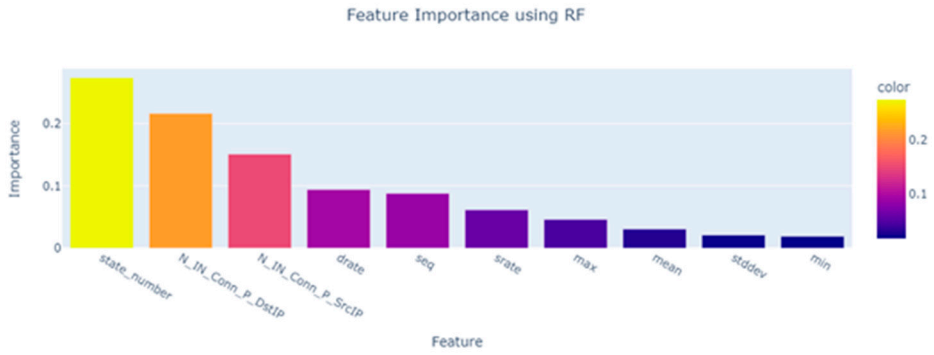

Figure 2. A feature importance analysis using Random Forest.

3. SMOTE Approach

An enhanced approach for handling unbalanced data is shown in Figure 3. The SMOTE algorithm was first presented in [38]. The SMOTE algorithm generates new samples by performing random linear interpolation between a select number of samples and the samples that are located nearby [39].

To enhance the classification impact of the unbalanced dataset and thus raise the data imbalance ratio, a given number of false minority samples are generated, as shown in Figure 4. This causes the data imbalance ratio to transform into balanced data.

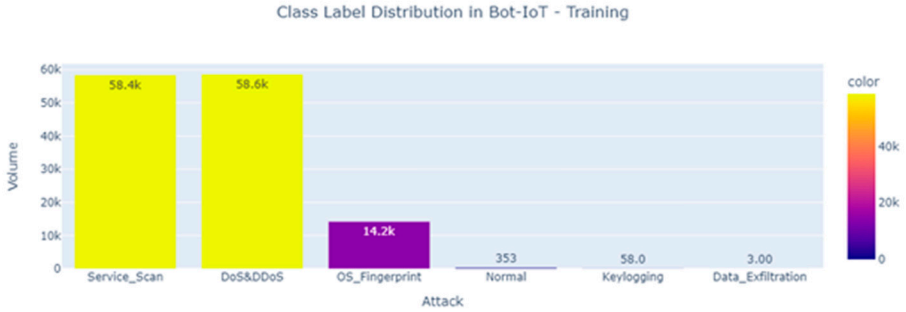

Figure 3. The attack class distributions.

3.3. Ensemble Learning

- Extra Trees classifier

The Extra Trees algorithm is a variant of the Random Forest algorithm. The algorithm exhibits resemblances to Random Forests, but includes extra randomness in the construction of decision trees [40]. The Extra Trees classifier employs an ensemble learning technique by aggregating multiple base classifiers to generate predictions, thereby harnessing the collective intelligence of the group [41]. It is an ensemble method that aggregates the outputs of numerous trees that have been trained independently. This technique can be used for classification tasks, where the final prediction is determined via majority voting.

In our specific context, the Extra Trees classifier ensemble is utilized for detecting various types of attacks. The Extra Trees classifier can mitigate overfitting and improve the accuracy of generalization by aggregating its results [42].

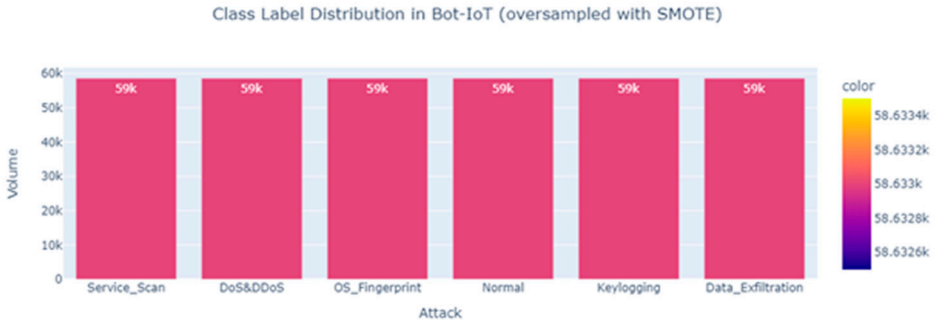

Figure 4. Attack class distributions after oversampling the BoT-IoT dataset using the SMOTE algorithm.

- Histogram-based Gradient Boosting classifier

This involves the utilization of gradient boosting, which includes the continually evolving training of an ensemble of weak learners [43]. Weak learners commonly use decision trees to rectify the inaccuracies of preceding models [44]. This classifier uses a methodology that employs histograms to enhance both computational efficiency and predictive accuracy.

Histograms are generated to provide statistical information regarding the distribution of data within each bin [45]. This includes metrics such as the count of samples and the aggregate of gradients or gradients squared. The utilization of these statistics facilitates proficient calculations throughout the training procedure, resulting in enhanced scalability and performance. The approach involves the sequential application of weak learners to the negative gradients of the loss function, leading to a gradual reduction in residual errors. The learning rate parameter regulates the weight assigned to each individual weak learner, thereby achieving a delicate balance between the complexities of the model and the ratio of convergence [46].

- Adaptive Boosting classifier

Adaptive Boosting (AdaBoost) combines weak learners iteratively to produce a robust classifier [47]. The weak learners are usually decision trees, and their predictions have weights according to their training outcomes. This classifier provides several noteworthy advantages that contribute to its popularity and efficacy in the field of machine learning. AdaBoost has an exceptional ability to enhance classification accuracy compared to a single weak learner. AdaBoost focuses on challenging samples by iteratively training weak learners on various subgroups of the data and allocating higher weights to misclassified instances, thereby reducing bias and increasing overall accuracy [48]. AdaBoost is a flexible algorithm applicable to a variety of classification problems, including binary classification and multi-class classification. It can manage both discrete and continuous characteristics, making it applicable to a wide variety of datasets [49]. AdaBoost assigns feature weights according to their classification usefulness. During training, AdaBoost modifies its weights to give misclassified instances and outliers less weight. This robustness allows AdaBoost to effectively handle noisy data and outliers that would otherwise adversely affect other classifiers.

- LGBM classifier

The Light Gradient Boosting Machine (LightGBM) classifier is a powerful ML algorithm that falls under the category of gradient boosting frameworks [50]. It is specifically designed to deliver high performance, efficiency, and accuracy in solving classification tasks. The LGBM classifier is based on the gradient boosting framework, which involves

iteratively training an ensemble of weak learners, typically decision trees, to sequentially correct the errors made by the previous models. LightGBM has several unique optimizations to enhance the overall efficiency and effectiveness of the boosting process. It utilizes a histogram-based approach for binning the continuous features, which significantly reduces the memory footprint and speeds up the training process [51]. It constructs histograms for each feature and uses these histograms to find the best splits for building decision trees efficiently.

- CatBoosting classifier

CatBoost is a robust machine learning technique specifically intended for classification tasks [52]. It falls within the category of gradient boosting structures and provides several unique characteristics that make a difference to its efficiency and efficacy. It employs a unique technique that combines gradient-based pre-sorting, ordered boosting, and symmetric decision trees. CatBoost can manage categorical characteristics with a variety of cardinalities, such as variables with high cardinality. CatBoost guarantees stability by combining, at random, the order of each category throughout training, preventing the model from depending exclusively on the order of the categories [53]. In addition, CatBoost includes a symmetric building of trees technique that takes the statistical characteristics of the dataset into consideration.

3.4. Evaluation Metrics

Once a model has been trained, its performance must be evaluated. In this study, we evaluate the effectiveness of suggested DL models using five widely accepted evaluation metrics: precision, recall, computation time, accuracy, and F1-score [54,55]. The evaluation metrics will be computed using Equations (1)–(4), which demonstrate related formulations for each of these measures based on TPR, TNR, FPR, and FPR results.

- True positive rate (TPR): ratio of observed positives to expected positives;
- False positive rate (FPR): ratio of values that are truly negative but are expected to be positive;
- False negative rate (FNR): ratio values that are in fact positive but are projected to be negative;
- True negative rate (TNR): ratio values that are negative and anticipated to become negative;
- Precision: the capacity of a system to accurately detect the existence of an attack or security breach; it illustrates the relationship between precisely predicted attacks and actual consequences:

$$\text{Precision} = \text{TPR}/(\text{TPR} + \text{FPR}); \quad (1)$$

- Recall: the system's ability to correctly recognize a botnet attack when it occurs on a network:

$$\text{Recall} = \text{TPR}/(\text{TPR} + \text{FPR}); \quad (2)$$

- Accuracy: the system's ability to effectively classify attack and non-attack packets; it represents the percentage of accurate predictions relative to the total number of samples:

$$\text{Accuracy} = (\text{TNR} + \text{TPR})/(\text{TPR} + \text{FNR} + \text{FPR} + \text{TNR}); \quad (3)$$

- F1-score: average of recall and precision; it provides the percentage of normal and attacking flow samples accurately anticipated in the testing sample:

$$\text{F1-Score} = 2 \times (\text{Recall} \times \text{Precision})/(\text{Recall} + \text{Precision}); \quad (4)$$

- Time complexity: how quickly or slowly an algorithm performs in the same relation to the amount of data.

4. Results

4.1. Experimental Settings

To perform our experiments, we used the Python programming language, as well as several AI and deep learning frameworks and packages that serve as benchmarks. These included the TensorFlow and Keras libraries, which were run on the Google CoLab GPU environment. Following up on what was covered previously, the first thing we did was apply data pre-processing and feature engineering methods to the BOT-IoT dataset. After that, we trained deep learning models using the training and test set, and finally evaluated all the learned models.

To execute the experiment, the database was initially partitioned into three parts: 70% for the training, 20% for the validation development, and 10% for the testing.

The performance measurements, such as precision, recall, accuracy, and F1-score, are reported via weighted average outcomes and other metrics such as model size and computation time.

4.2. Experimental Results

In this section, we overview the experimental outcomes of our study, which evaluated the performance of ten separate ML models for detecting malware. These models consist of two single classifiers, ensemble classifiers, and four architectures for deep learning. As shown in Tables 2 and 3, we also compare the efficacy of these models with and without the SMOTE algorithm for managing imbalanced data.

Table 2. Performance results for detecting IoT network attacks without using the SMOTE algorithm.

Metric	Accuracy	Precision	Recall	F1-Score	CPU Time	Model Size (MB)
Random Forest	0.9518	0.9538	0.9284	0.9403	21.6 s	23.6
Extra Trees	0.9674	0.9652	0.9517	0.9582	47.6 s	598.7
KNN	0.9083	0.9036	0.8869	0.8947	3.29 s	13.6
SVM	0.6121	0.6280	0.3598	0.3695	21 min 50 s	12.2
HistGBoost	0.9560	0.7488	0.7332	0.7321	13.4 s	1.2
AdaBoost	0.1211	0.4552	0.3482	0.0826	1 min 19 s	0.31
LGBM	0.9323	0.4665	0.4739	0.4690	36.1 s	1.8
CatBoost	0.9819	0.9686	0.9608	0.9646	2 min 55 s	3.5
XGBoost	0.9852	0.9806	0.9654	0.9727	2 min 43 s	1.1
MLP	0.7539	0.3031	0.2942	0.2850	31.5 s	0.005
ANN	0.8308	0.3308	0.4789	0.3701	13 min 48 s	0.027
LSTM	0.7701	0.4887	0.3476	0.3682	10 min 10 s	7.7
GRU	0.8536	0.6058	0.4517	0.4902	11 min 1 s	7.7
RNN	0.8682	0.9189	0.7631	0.8013	10 min 50 s	1.6
Bagging	0.9398	0.9324	0.9160	0.9238	2 min 54 s	240.5

Table 3. Performance results for detecting IoT network attacks using the SMOTE algorithm from the BoT-IoT dataset.

Metric	Accuracy	Precision	Recall	F1-Score	CPU Time	Model (MB)
CatBoost	0.97661	0.91249	0.9815	0.94369	7 min 43 s	3.48
XGBoost	0.97986	0.94868	0.98084	0.96383	7 min 53 s	1.22
MLP	0.53336	0.31119	0.63571	0.32423	4 min 47 s	0.02
ANN	0.76594	0.61794	0.89682	0.63602	31 min 41 s	0.03
LSTM	0.83418	0.75511	0.92699	0.76773	30 min 6 s	7.69
GRU	0.87806	0.78463	0.93476	0.83175	29 min 50 s	7.69
RNN	0.87147	0.77572	0.94066	0.8257	27 min 3 s	1.62
Bagging	0.94099	0.91357	0.93127	0.92205	9 min 31 s	350.73
Random Forest	0.9425	0.90961	0.9635	0.9304	1 min 7 s	29.60
Extra Trees	0.90922	0.88756	0.8952	0.8906	3.43 s	35.19

Table 3. *Cont.*

Metric	Accuracy	Precision	Recall	F1-Score	CPU Time	Model (MB)
KNN	0.90922	0.88756	0.8952	0.8906	3.43 s	35.19
SVM	0.59398	0.4853	0.63258	0.48259	1 h 18 min 19 s	25.34
HistGboost	0.97437	0.97758	0.97437	0.97511	47.6 s	1.90
AdaBoost	0.43068	0.32098	0.34041	0.25093	3 min 55 s	0.31
LGBM	0.98242	0.96029	0.98055	0.96986	4 min 5 s	11.05

4.2.1. Experiments without Using the SMOTE Algorithm

The performance results for the deep learning models on the BoT-IoT dataset, presented in Table 2, reveal varying levels of performance in terms of accuracy, precision, recall, and F1-score. It is important to note that these results were obtained without utilizing the SMOTE algorithm.

From the results, it is observed that Random Forest, Extra Trees, and KNN achieved competitive performance in terms of accuracy, precision, recall, and F1-score. These models were able to effectively classify instances in the dataset without the need for oversampling techniques. Notably, Random Forest achieved the highest accuracy of 95.183%, closely followed by Extra Trees, with an accuracy of 96.741%. These models also exhibited high precision and F1-score, indicating their ability to correctly classify positive instances and achieve a balance between precision and recall.

Table 2 presents the performance results of various machine learning models without using the SMOTE oversampling technique on the BoT-IoT dataset for detecting IoT network attacks.

On the other hand, models such as SVM and AdaBoost showed lower performance compared to the ensemble models. SVM exhibited relatively lower accuracy and F1-score, indicating challenges in effectively handling the imbalanced nature of the dataset. AdaBoost, while having a low accuracy of 12.11%, achieved higher precision compared to other metrics, suggesting a bias towards correctly classifying positive instances.

The models' size varied across the different models, with Extra Trees having the largest model size of 598.7 MB, followed by SVM with 12.2 MB. Meanwhile, models such as AdaBoost and HistGBoost had considerably smaller model sizes. Overall, the results indicate that certain models, particularly Random Forest and Extra Trees, performed well without the need for SMOTE oversampling. These models were able to effectively capture the underlying patterns in the dataset and achieve satisfactory classification performance. However, further investigation and experimentation may be required in order to understand the impact of the dataset characteristics and the specific requirements of the problem domain on the model performance.

All models were evaluated based on their accuracy, precision, recall, and F1-score, as shown in Figure 5.

The performance results for the DL models on the BoT-IoT dataset, presented in Figure 6, show varying levels of performance in terms of accuracy, precision, recall, and F1-score. It is important to note that the models obtained these results without utilizing the SMOTE algorithm.

Among the deep learning models, MLP demonstrated the lowest performance across all metrics; it achieved an accuracy of 0.75, indicating that it correctly classified approximately 75% of the instances. The precision, recall, and F1-score were also relatively low at 0.3, 0.29, and 0.29, respectively. These metrics indicate that the MLP model struggled to accurately detect IoT network attacks, showing a significant number of false positives and false negatives. Furthermore, the relatively small model size of 0.005 MB suggests that MLP is a lightweight model.

The ANN model performed relatively better than MLP, with an accuracy of 0.831. It exhibited improved precision of 0.33 and recall of 0.48, suggesting a better balance between true positives and false negatives. However, the F1-score of 0.37 indicates that the model's

ability to achieve a balance between precision and recall is still limited. Despite the longer CPU time of 13 min and 48 s, the ANN model maintained a small model size of 0.027 MB.

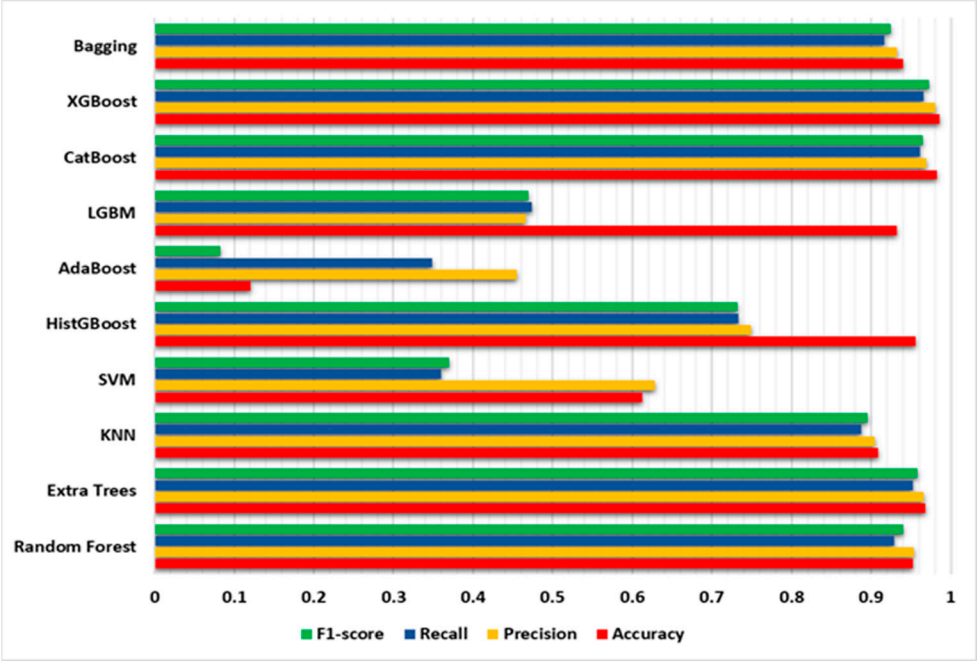

Figure 5. The proposed ML models' evaluation results on the BoT-IoT dataset without using the SMOTE algorithm.

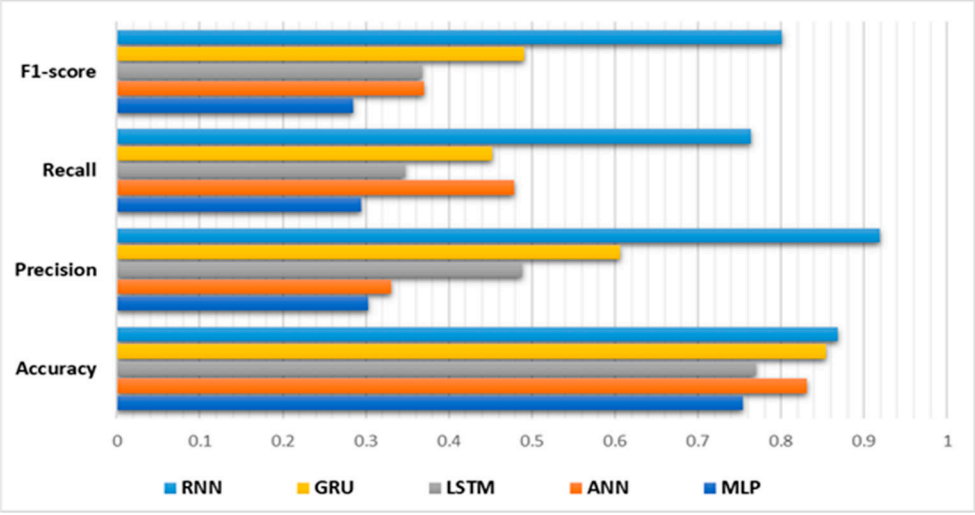

Figure 6. The proposed deep learning models' evaluation results without using the SMOTE algorithm.

Moving on to the LSTM and GRU models, they achieved comparable performance levels; the LSTM model achieved an accuracy of 0.77, while the GRU model achieved a higher accuracy of 0.85. Both models showed improvements in precision compared to MLP and ANN, with values of 0.49 and 0.61, respectively. However, the recall values for both models, at 0.35 for LSTM and 0.45 for GRU, were lower. The F1 scores for LSTM and GRU were 0.37 and 0.49, respectively, showing a moderate balance between precision and recall. Both the LSTM and GRU models had longer CPU times compared to the MLP and ANN models, with durations of 10 min and 10 s for LSTM and 11 min and 1 s for GRU. Both models had a larger model size of 7.7 MB. Finally, the RNN model exhibited the highest accuracy among all the deep learning models, with a value of 0.86819. It achieved a

significantly higher precision of 0.92, showing a strong ability to correctly classify positive instances. The recall value of 0.76 and the F1-score of 0.80 further support the model's effectiveness in getting true positives and achieving a good balance between precision and recall. However, the RNN model had a longer CPU time of 10 min and 50 s, and a larger model size of 1.6 MB compared to MLP and ANN.

4.2.2. Experiments Using the SMOTE Algorithm

The performance results for detecting IoT network intrusions using the SMOTE algorithm on the BoT-IoT dataset are presented in Table 3. The table provides a thorough review of several machine learning models based on their precision, recall, F1-score, CPU time, and model size. These metrics are essential for evaluating the effectiveness and efficacy of models in detecting attacks on IoT networks, as shown in Figure 7.

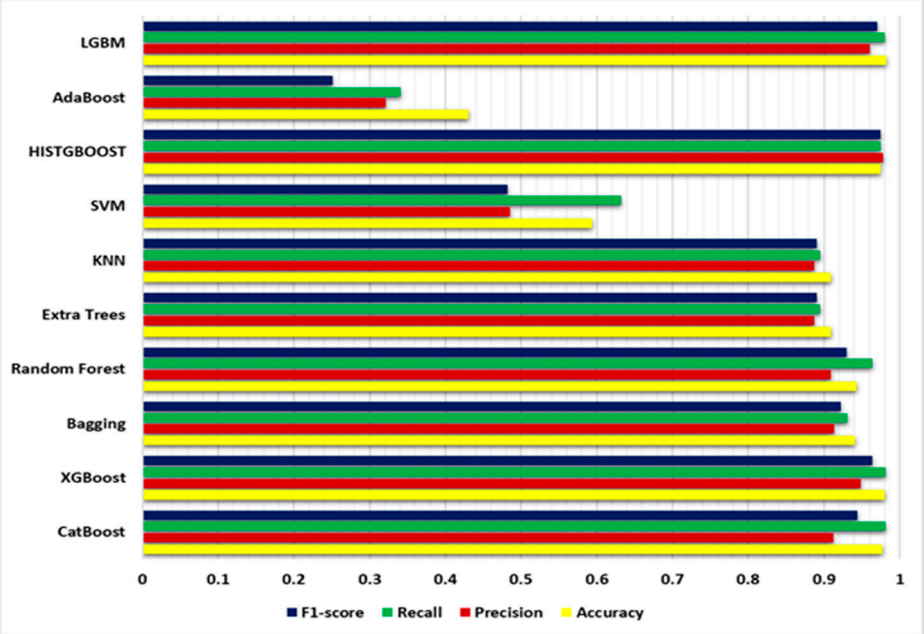

Figure 7. The proposed machine learning models' evaluation results using the SMOTE algorithm.

CatBoost, the first model listed in the table, obtained an accuracy of 97.661%. It exhibited high precision (0.91243), indicating a low rate of false positives, and high recall (0.9815, indicating a low rate of false negatives. The F1-score of 0.94369 demonstrates the balance between accuracy and recall. The model required 7 min and 43 s of CPU time to train and was 3.48 MB in size.

The accuracy of the second model, XGBoost, was slightly greater at 97.986%. It demonstrated a greater precision of 0.94866 and a comparable recall of 0.98084. The F1-score of 0.96383 indicates a marginally better overall performance than CatBoost. The XGBoost training duration was 7 min and 53 s, and the size of the model was 1.22 MB.

In contrast, the performance of the MLP model was substantially inferior across all metrics. It attained a precision of 0.5336, indicating a substantial number of misclassifications. The precision of 0.31 and recall of 0.64 reflect the model's inability to identify positive instances accurately. The F1-score of 0.32, therefore, indicates a poor overall performance. However, the MLP model's training duration was only 4 min and 47 s, and its model size was only 0.02 MB.

The ANN model performed better than the MLP model, but it still lagged behind CatBoost and XGBoost. Its accuracy was 0.76594, its precision was 0.62, and its recall was 0.897. The F1-score of 0.64 indicates a satisfactory equilibrium between precision and recall. However, the training period for the ANN model was significantly longer at 31 min and 41 s, and the model size was slightly larger at 0.03 MB.

The performances of the LSTM, GRU, and RNN models, as shown in Figure 8, were superior to those of the MLP and ANN models. The LSTM model obtained an accuracy of 0.83418, along with precision, recall, and F1-scores of 0.755, 0.927, and 0.77, respectively. The GRU and RNN models exhibited comparable performance, with accuracies of 0.878 and 0.87, respectively. However, all three models required highly training periods of approximately 30 min, and their model sizes are larger at 7.69 MB.

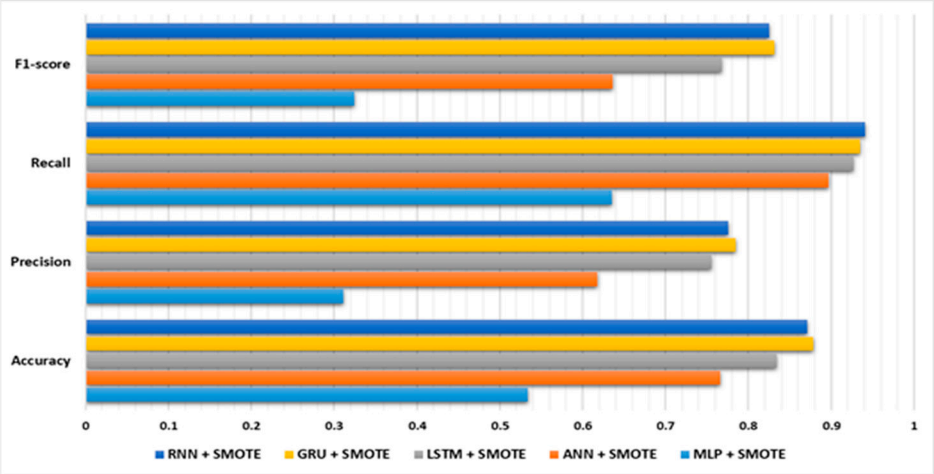

Figure 8. The proposed deep learning models' evaluation results using the SMOTE algorithm.

Moving on to ensemble methods, the Bagging model obtained an accuracy of 0.94000, with a high level of precision (0.91357) and recall (0.93127). The F1-score of 0.92205 indicates a balanced combination of precision and recall. The model required 9 min and 31 s of training time and had a larger model dimension of 350.73 MB. Similarly, the Random Forest model had an accuracy of 0.94, precision of 0.91, recall of 0.9635, and F1-score of 0.93. It had a training duration of one minute and seven seconds and a model size of 29.60 MB, which was shorter and smaller, respectively.

Both the Extra Trees and KNN models achieved an accuracy of 0.90922, with precision, recall, and F1-scores of 0.89, 0.895, and 0.8906, respectively. These models had much reduced training times of 3.43 s and larger model sizes of 35.19 MB.

In contrast, the SVM model performed significantly worse across all metrics. It obtained a precision of 0.59, indicating a substantial number of misclassifications. Precision of 0.49 and recall of 0.63 indicate the model's inability to accurately identify positive instances. Overall, the F1-score of 0.48259 indicates poor performance. The training duration for the SVM model was 1 h, 18 min, and 19 s, and the size of the model was 25.34 MB.

Finally, the HISTGBOOST, AdaBoost, and LGBM models demonstrated enhanced performance. The accuracy of the HISTGBOOST model was 0.97437, and its precision, recall, and F1-scores were 0.97758, 0.97437, and 0.97511, respectively. It had a training time of 47.6 s and a model size of 1.90 MB, which are both relatively brief. With an accuracy of 0.43068, lower precision and recall ranging from 0.32098 to 0.34041, and an F1-score of 0.25093, the AdaBoost model demonstrated lower performance. It required 3 min and 55 s of training time and was 0.31 MB in size. The LGBM model demonstrated the highest accuracy, with a value of 0.98242, as well as excellent precision, recall, and F1-scores of 0.96029, 0.98055, and 0.96986, respectively. The LGBM training duration was 4 min and 5 s, and the size of the model was 11.05 MB.

Using the SMOTE algorithm on the BoT-IoT dataset, the CatBoost and XGBoost models demonstrated superior performance in detecting IoT network attacks based on the performance metrics presented in Table 3. These models attained high levels of accuracy, precision, recall, and F1-scores, demonstrating their ability to identify both positive and negative instances. In addition, they had shorter training durations and smaller model sizes than other models, making them possible choices for detecting IoT network attacks.

5. Discussion

The performance results presented in Tables 2 and 3 provide information on the efficacy of various classifiers in detecting IoT network intrusions on the BoT-IoT dataset. A comparison of these tables reveals the effect that the SMOTE algorithm has on the performance metrics.

Figure 9 displays the comparison between the ensemble learning models' performance with and without using the SMOTE algorithm. It demonstrates that best results from several classifiers, including, Extra Trees, CatBoost, and XGBoost, attained high accuracies. Additionally, these classifiers exhibited favorable precision, recall, and F1-scores, indicating their ability to accurately identify IoT network attacks. Notably, CatBoost and XGBoost consistently demonstrated superior performance across a variety of metrics, i.e., even better than using these classifiers with the SMOTE algorithm.

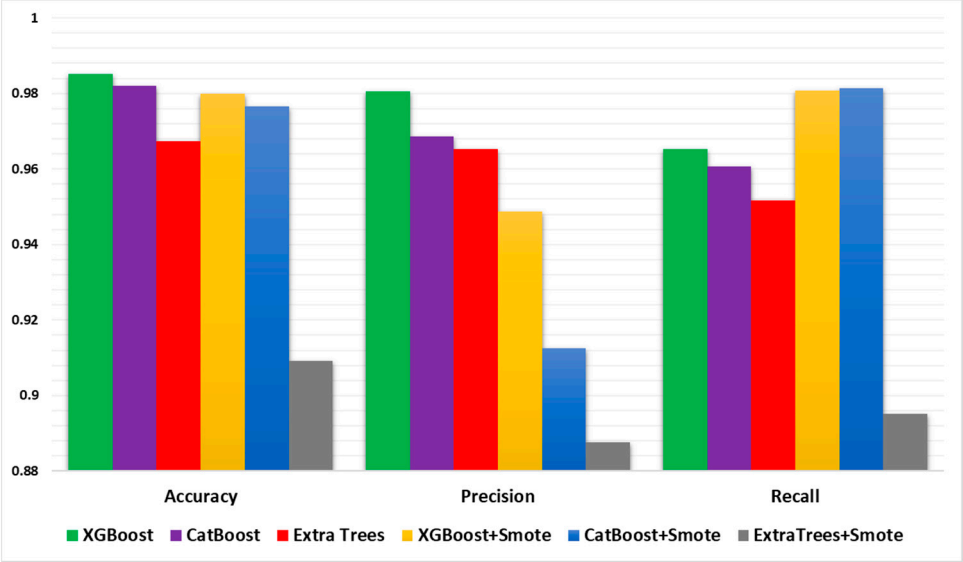

Figure 9. Comparison of the best ensemble learning models' results with and without using the SMOTE algorithm.

In contrast, the comparison between deep learning models' performance with and without using the SMOTE algorithm is shown in Figure 9. Comparing the two tables reveals that the SMOTE algorithm has affected the performance many different classifiers. From Figure 9, CatBoost and XGBoost, which performed exceptionally better (as shown in Table 2), maintained their high accuracy and obtained competitive precision, recall, and F1-scores, as shown in Table 3. This indicates that these classifiers are robust and that the use of SMOTE has minimal effect on their efficacy.

On the other hand, the application of SMOTE altered the performance of some classifiers significantly. In Table 3 and Figure 10, MLP, ANN, LSTM, and GRU models exhibited lower accuracies, precision, recall, and F1-scores than they did in Table 2. This suggests that even with SMOTE, these classifiers cannot be as effective when dealing with imbalanced datasets. However, it is important to note that these models still obtained a respectable level of accuracy and other metrics. In addition, classifiers such as Bagging, Random Forest, Extra Trees, KNN, and LGBM showed consistent performance across both tables, demonstrating their robustness in dealing with imbalanced datasets. While the accuracies and F1-scores remained relatively stable, the application of SMOTE marginally improved the precision and recall values for these classifiers.

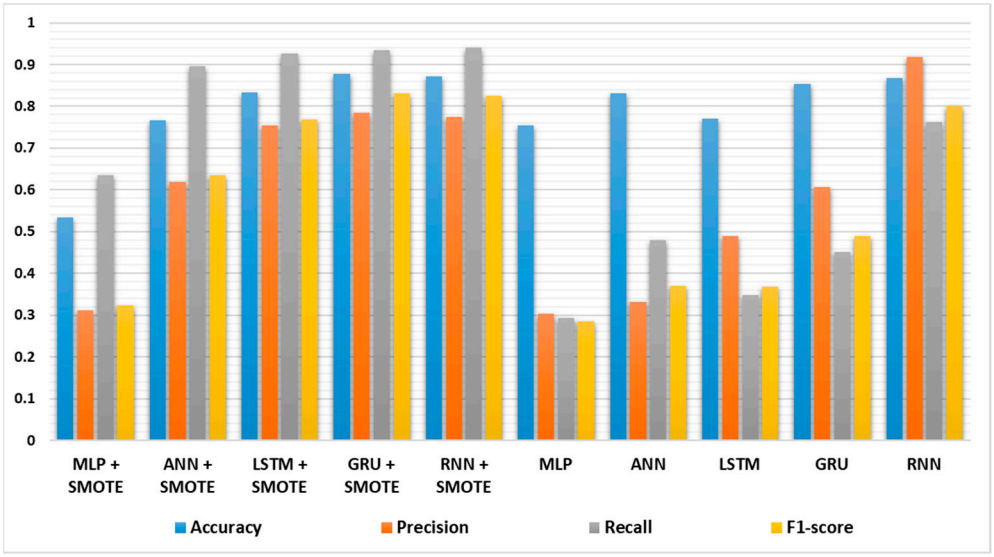

Figure 10. Comparison of deep learning models' performance with and without using the SMOTE algorithm.

With lower accuracies, precision, recall, and F1-scores, these classifiers struggled to detect IoT network attacks with accuracy. Moreover, SVM had significantly greater CPU times than other classifiers in both scenarios, indicating its computational complexity, as seen in Figure 11.

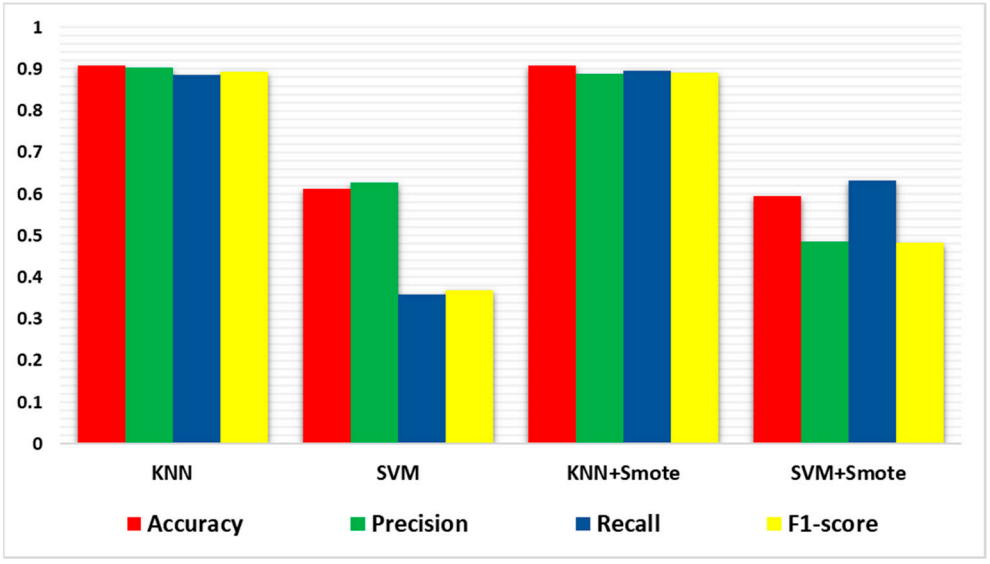

Figure 11. Comparison of single-classifier models' performance with and without using the SMOTE algorithm.

Recently, several studies have employed deep learning algorithms for boosting the efficiency of training models, as shown in Table 3. However, these studies do not study networks in the IoT domain completely, i.e., to identify irregularities quickly and accurately in order to detect and react to IoT attacks and to overcome machine and deep learning issues such as acquiring the optimal number of neurons, overfitting, and parameters that accomplish an ideal model objective.

The Table 4 presents a comparative analysis of the latest IoT cybersecurity solutions, showcasing the performance results achieved by different studies in the field. Each row represents a specific research reference, including the year of the study, the dataset used, the methodology employed, the algorithms utilized, and the accuracy results as percentages.

Table 4. Comparative analysis of the latest IoT cybersecurity solutions.

Ref.	Data Used	Methodology Used	Accuracy (%)
Mendonça et al. [23]	DS2OS, CICIDS2017	Deep learning	98
Popoola et al. [26]	BoT-IoT	LAE for dimensionality reduction and BLSTM classifier	91.89
Alharbi et al. [27]	N-BaIoT	A Local–Global best Bat Algorithm for Neural Networks	90
Saharkhizan et al. [30]	Modbus/TCP network traffic	LSTM and Ensemble learning	98.99
Pokhrel et al. [31]	BoT-IoT	Deep learning	87.4
Proposed	**BoT-IoT**	**CatBoosting XGBoosting**	**98.19 98.52**

6. Conclusions

The objective of this dissertation is to implement an intelligent system for IoT protection devices using a novel deep learning-based model to manage extremely complex datasets. Additional research has led to the development of intrusion detection systems with centralized architecture, deep learning, and machine learning.

To overcome numerous obstacles, such as overfitting, extended training times, and low model accuracy, the proposed models will combine deep learning approaches with feature engineering. On class-imbalanced data, the oversampling technique (SMOTE) was applied, whereas the efficacy of fifteen algorithms was evaluated on class-balanced data. CatBoost and XGBoost outperform deep learning models that learn from experience, especially when identifying future cyberattacks against IoT networks. A real-time dataset BoT-IoT represents enormous volumes of traffic that are affected by multiple types of attacks. CatBoost and XGBoost classifiers attained respective accuracy rates of 98.19% and 98.50%. The best classifiers are consistent and dependable across the BoT-IoT dataset, making them viable options for detecting IoT network attacks regardless of the implementation of the SMOTE algorithm.

The future of this research project will include comparing distributed deep learning with other data using different ensemble learning algorithms and neural network architectures. Additionally, we plan to study the detection of mobile network intrusions with ensemble learning algorithms, feature engineering, and optimization techniques.

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

Funding: This research received no external funding.

Data Availability Statement: The publicly available dataset BoT-IoT dataset was analyzed in this study. This data can be found here: https://ieee-dataport.org/documents/bot-iot-dataset, accessed on 24 September 2023.

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

References

1. Vermesan, O.; Friess, P.; Guillemin, P.; Giaffreda, R.; Grindvoll, H.; Eisenhauer, M.; Serrano, M.; Moessner, K.; Spirito, M.; Blystad, L.-C. Internet of Things beyond the Hype: Research, Innovation and Deployment. In *Building the Hyperconnected Society-Internet of Things Research and Innovation Value Chains, Ecosystems and Markets*; River Publishers: Gistrup, Denmark, 2022; pp. 15–118; ISBN 1-00-333745-7.
2. Madina, S.F.; Islam, M.S.; Alamgir, F.M.; Ferdous, M.F. Internet of Things (IoT)-Based Industrial Monitoring System. In *Industrial Internet of Things*; CRC Press: Boca Raton, FL, USA, 2022; pp. 55–86; ISBN 1-00-310226-3.
3. Huang, L. Design of an IoT DDoS Attack Prediction System Based on Data Mining Technology. *J. Supercomput.* **2022**, *78*, 4601–4623. [CrossRef]
4. Krichen, M. A Survey on Formal Verification and Validation Techniques for Internet of Things. *Appl. Sci.* **2023**, *13*, 8122. [CrossRef]

5. Idrissi, I.; Azizi, M.; Moussaoui, O. IoT Security with Deep Learning-Based Intrusion Detection Systems: A Systematic Literature Review. In Proceedings of the 2020 Fourth International Conference on Intelligent Computing in Data Sciences (ICDS), Fez, Morocco, 21–23 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–10.
6. Krichen, M. Strengthening the Security of Smart Contracts through the Power of Artificial Intelligence. *Computers* **2023**, *12*, 107. [CrossRef]
7. Abdalzaher, M.S.; Krichen, M.; Yiltas-Kaplan, D.; Ben Dhaou, I.; Adoni, W.Y.H. Early Detection of Earthquakes Using IoT and Cloud Infrastructure: A Survey. *Sustainability* **2023**, *15*, 11713. [CrossRef]
8. Majid, M.; Habib, S.; Javed, A.R.; Rizwan, M.; Srivastava, G.; Gadekallu, T.R.; Lin, J.C.-W. Applications of Wireless Sensor Networks and Internet of Things Frameworks in the Industry Revolution 4.0: A Systematic Literature Review. *Sensors* **2022**, *22*, 2087. [CrossRef]
9. Oseni, A.; Moustafa, N.; Creech, G.; Sohrabi, N.; Strelzoff, A.; Tari, Z.; Linkov, I. An Explainable Deep Learning Framework for Resilient Intrusion Detection in IoT-Enabled Transportation Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1000–1014. [CrossRef]
10. Nasir, M.; Javed, A.R.; Tariq, M.A.; Asim, M.; Baker, T. Feature Engineering and Deep Learning-Based Intrusion Detection Framework for Securing Edge IoT. *J. Supercomput.* **2022**, *78*, 8852–8866. [CrossRef]
11. Baduge, S.K.; Thilakarathna, S.; Perera, J.S.; Arashpour, M.; Sharafi, P.; Teodosio, B.; Shringi, A.; Mendis, P. Artificial Intelligence and Smart Vision for Building and Construction 4.0: Machine and Deep Learning Methods and Applications. *Autom. Constr.* **2022**, *141*, 104440. [CrossRef]
12. Boulila, W.; Driss, M.; Alshanqiti, E.; Al-Sarem, M.; Saeed, F.; Krichen, M. Weight Initialization Techniques for Deep Learning Algorithms in Remote Sensing: Recent Trends and Future Perspectives. In *Advances on Smart and Soft Computing*; Saeed, F., Al-Hadhrami, T., Mohammed, E., Al-Sarem, M., Eds.; Springer: Singapore, 2022; pp. 477–484.
13. Islam, U.; Muhammad, A.; Mansoor, R.; Hossain, M.S.; Ahmad, I.; Eldin, E.T.; Khan, J.A.; Rehman, A.U.; Shafiq, M. Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models. *Sustainability* **2022**, *14*, 8374. [CrossRef]
14. Vadivelan, N.; Bhargavi, K.; Kodati, S.; Nalini, M. Detection of Cyber Attacks Using Machine Learning. In *AIP Conference Proceedings*; AIP Publishing LLC: Melville, NY, USA, 2022; Volume 2405, p. 030003.
15. Ahmad, R.; Alsmadi, I.; Alhamdani, W.; Tawalbeh, L. A Comprehensive Deep Learning Benchmark for IoT IDS. *Comput. Secur.* **2022**, *114*, 102588. [CrossRef]
16. Iwendi, C.; Rehman, S.U.; Javed, A.R.; Khan, S.; Srivastava, G. Sustainable Security for the Internet of Things Using Artificial Intelligence Architectures. *ACM Trans. Internet Technol.* **2021**, *21*, 1–22. [CrossRef]
17. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-Year Anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 63–905. Available online: https://www.jair.org/index.php/jair/article/view/11192 (accessed on 24 September 2023). [CrossRef]
18. Torgo, L.; Ribeiro, R.P.; Pfahringer, B.; Branco, P. SMOTE for Regression. In *Progress in Artificial Intelligence*; Correia, L., Reis, L.P., Cascalho, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 378–389.
19. Khan, A.; Cotton, C. Efficient Attack Detection in IoT Devices Using Feature Engineering-Less Machine Learning. *arXiv* **2023**, arXiv:2301.03532. [CrossRef]
20. Krichen, M. Convolutional Neural Networks: A Survey. *Computers* **2023**, *12*, 151. [CrossRef]
21. Aswad, F.M.; Ahmed, A.M.S.; Alhammadi, N.A.M.; Khalaf, B.A.; Mostafa, S.A. Deep Learning in Distributed Denial-of-Service Attacks Detection Method for Internet of Things Networks. *J. Intell. Syst.* **2023**, *32*, 20220155. [CrossRef]
22. A Heuristic Distributed Scheme to Detect Falsification of Mobility Patterns in Internet of Vehicles. Available online: https://ieeexplore.ieee.org/abstract/document/9445064 (accessed on 24 September 2023).
23. Mendonça, R.V.; Silva, J.C.; Rosa, R.L.; Saadi, M.; Rodriguez, D.Z.; Farouk, A. A Lightweight Intelligent Intrusion Detection System for Industrial Internet of Things Using Deep Learning Algorithms. *Expert Syst.* **2022**, *39*, e12917. [CrossRef]
24. Ferrag, M.A.; Friha, O.; Maglaras, L.; Janicke, H.; Shu, L. Federated Deep Learning for Cyber Security in the Internet of Things: Concepts, Applications, and Experimental Analysis. *IEEE Access* **2021**, *9*, 138509–138542. [CrossRef]
25. Popoola, S.I.; Ande, R.; Adebisi, B.; Gui, G.; Hammoudeh, M.; Jogunola, O. Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT-Edge Devices. *IEEE Internet Things J.* **2021**, *9*, 3930–3944. [CrossRef]
26. Popoola, S.I.; Adebisi, B.; Hammoudeh, M.; Gui, G.; Gacanin, H. Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks. *IEEE Internet Things J.* **2020**, *8*, 4944–4956. [CrossRef]
27. Alharbi, A.; Alosaimi, W.; Alyami, H.; Rauf, H.T.; Damaševičius, R. Botnet Attack Detection Using Local Global Best Bat Algorithm for Industrial Internet of Things. *Electronics* **2021**, *10*, 1341. [CrossRef]
28. Huma, Z.E.; Latif, S.; Ahmad, J.; Idrees, Z.; Ibrar, A.; Zou, Z.; Alqahtani, F.; Baothman, F. A Hybrid Deep Random Neural Network for Cyberattack Detection in the Industrial Internet of Things. *IEEE Access* **2021**, *9*, 55595–55605. [CrossRef]
29. Chowdhury, M.; Ray, B.; Chowdhury, S.; Rajasegarar, S. A Novel Insider Attack and Machine Learning Based Detection for the Internet of Things. *ACM Trans. Internet Things* **2021**, *2*, 1–23. [CrossRef]
30. Saharkhizan, M.; Azmoodeh, A.; Dehghantanha, A.; Choo, K.-K.R.; Parizi, R.M. An Ensemble of Deep Recurrent Neural Networks for Detecting IoT Cyber Attacks Using Network Traffic. *IEEE Internet Things J.* **2020**, *7*, 8852–8859. [CrossRef]
31. Pokhrel, S.; Abbas, R.; Aryal, B. IoT Security: Botnet Detection in IoT Using Machine Learning. *arXiv* **2021**, arXiv:2104.02231.

32. Latif, S.; Zou, Z.; Idrees, Z.; Ahmad, J. A Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network. *IEEE Access* **2020**, *8*, 89337–89350. [CrossRef]
33. *The Bot-Iot Dataset*; IEEE: Piscataway, NJ, USA, 2019; Volume 5.
34. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Umar, A.M.; Linus, O.U.; Arshad, H.; Kazaure, A.A.; Gana, U.; Kiru, M.U. Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access* **2019**, *7*, 158820–158846. [CrossRef]
35. Garavand, A.; Behmanesh, A.; Aslani, N.; Sadeghsalehi, H.; Ghaderzadeh, M. Towards Diagnostic Aided Systems in Coronary Artery Disease Detection: A Comprehensive Multiview Survey of the State of the Art. *Int. J. Intell. Syst.* **2023**, *2023*, 6442756. Available online: https://www.hindawi.com/journals/ijis/2023/6442756/ (accessed on 24 September 2023). [CrossRef]
36. Fan, C.; Chen, M.; Wang, X.; Wang, J.; Huang, B. A Review on Data Preprocessing Techniques toward Efficient and Reliable Knowledge Discovery From Building Operational Data. *Front. Energy Res.* **2021**, *9*, 652801. [CrossRef]
37. Ghaderzadeh, M.; Aria, M.; Asadi, F. X-Ray Equipped with Artificial Intelligence: Changing the COVID-19 Diagnostic Paradigm during the Pandemic. *BioMed Res. Int.* **2021**, *2021*, e9942873. [CrossRef]
38. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
39. Nagisetty, A.; Gupta, G.P. Framework for Detection of Malicious Activities in IoT Networks Using Keras Deep Learning Library. In Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 633–637.
40. González, S.; García, S.; Del Ser, J.; Rokach, L.; Herrera, F. A Practical Tutorial on Bagging and Boosting Based Ensembles for Machine Learning: Algorithms, Software Tools, Performance Study, Practical Perspectives and Opportunities. *Inf. Fusion* **2020**, *64*, 205–237. [CrossRef]
41. Acosta, M.R.C.; Ahmed, S.; Garcia, C.E.; Koo, I. Extremely Randomized Trees-Based Scheme for Stealthy Cyber-Attack Detection in Smart Grid Networks. *IEEE Access* **2020**, *8*, 19921–19933. [CrossRef]
42. Seyghaly, R.; Garcia, J.; Masip-Bruin, X.; Varnamkhasti, M.M. Interference Recognition for Fog Enabled IoT Architecture Using a Novel Tree-Based Method. In Proceedings of the 2022 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Barcelona, Spain, 1–3 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
43. Alghamdi, R.; Bellaiche, M. Evaluation and Selection Models for Ensemble Intrusion Detection Systems in IoT. *IoT* **2022**, *3*, 285–314. [CrossRef]
44. Almomani, O.; Almaiah, M.A.; Alsaaidah, A.; Smadi, S.; Mohammad, A.H.; Althunibat, A. Machine Learning Classifiers for Network Intrusion Detection System: Comparative Study. In Proceedings of the 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 14–15 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 440–445.
45. Heinrich, C. On the Number of Bins in a Rank Histogram. *Q. J. R. Meteorol. Soc.* **2021**, *147*, 544–556. [CrossRef]
46. Wang, K.; Lu, J.; Liu, A.; Zhang, G.; Xiong, L. Evolving Gradient Boost: A Pruning Scheme Based on Loss Improvement Ratio for Learning under Concept Drift. *IEEE Trans. Cybern.* **2021**, *53*, 2110–2123. [CrossRef] [PubMed]
47. Ding, Y.; Zhu, H.; Chen, R.; Li, R. An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification. *Appl. Sci.* **2022**, *12*, 5872. [CrossRef]
48. Mienye, I.D.; Sun, Y. A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE Access* **2022**, *10*, 99129–99149. [CrossRef]
49. Wang, Q.; Wei, X. The Detection of Network Intrusion Based on Improved Adaboost Algorithm. In Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, Nanjing, China, 10–12 January 2020; pp. 84–88.
50. Mishra, D.; Naik, B.; Nayak, J.; Souri, A.; Dash, P.B.; Vimal, S. Light Gradient Boosting Machine with Optimized Hyperparameters for Identification of Malicious Access in IoT Network. *Digit. Commun. Netw.* **2023**, *9*, 125–137. [CrossRef]
51. Seth, S.; Singh, G.; Kaur Chahal, K. A Novel Time Efficient Learning-Based Approach for Smart Intrusion Detection System. *J. Big Data* **2021**, *8*, 111. [CrossRef]
52. Sanjeetha, R.; Raj, A.; Saivenu, K.; Ahmed, M.I.; Sathvik, B.; Kanavalli, A. Detection and Mitigation of Botnet Based DDoS Attacks Using Catboost Machine Learning Algorithm in SDN Environment. *Int. J. Adv. Technol. Eng. Explor.* **2021**, *8*, 445.
53. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased Boosting with Categorical Features. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2018; Volume 31.
54. Ghaderzadeh, M.; Aria, M.; Hosseini, A.; Asadi, F.; Bashash, D.; Abolghasemi, H. A Fast and Efficient CNN Model for B-ALL Diagnosis and Its Subtypes Classification Using Peripheral Blood Smear Images. *Int. J. Intell. Syst.* **2022**, *37*, 5113–5133. Available online: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22753 (accessed on 24 September 2023). [CrossRef]
55. Hosseini, A.; Eshraghi, M.A.; Taami, T.; Sadeghsalehi, H.; Hoseinzadeh, Z.; Ghaderzadeh, M.; Rafiee, M. A Mobile Application Based on Efficient Lightweight CNN Model for Classification of B-ALL Cancer from Non-Cancerous Cells: A Design and Implementation Study. *Inform. Med. Unlocked* **2023**, *39*, 101244. [CrossRef]