University of Ibadan
Journal of Science and Logics in ICT Research

Botnet Attack Detection in Internet of Things Using Selected Learning Algorithms

[1]✉ Aremu B. L., [2]Aro T. O., [3]Saka K. K., [4]Seriki A. A. and [5]Raji R. O.
[1,2,3,4,5]Department of Computer Science, Al-Hikmah University Ilorin, Nigeria.
[1]lbaremu@alhikmah.edu.ng, [2]taiwo774@gmail.com, [3]kamilsaka67@gmail.com, [4]Aliyuadebayo7@gmail.com, [5]muhsinah2010@gmail.com

Abstract

The Internet of Things (IoT) refers to a network of everyday devices, such as smartphones and industrial sensors, all connected to the Internet, allowing them to communicate and share data. IoT networks comprise various devices with different functions, communication protocols, and computational capabilities. This heterogeneity complicates the development of a one-size-fits-all solution for botnet detection. Developing effective botnet detection systems for IoT environments is challenging due to the diversity of devices, each with unique characteristics and behaviours. This study focuses on creating a robust model to identify botnet attacks across various IoT devices. Using the NB-IoT-23 datasets, which include data from five distinct devices, supervised machine learning techniques, namely Logistic Regression, Linear Regression, Artificial Neural Network (ANN), K-nearest neighbours (KNN), and Bagging, were employed to identify the most accurate and efficient method. The research highlights the Bagging ensemble technique as particularly effective. The Bagging model demonstrated remarkable performance, achieving an accuracy of 99.96%, precision of 99.93%, recall of 99.98%, an F1 score of 99.96%, and a Receiver Operating Characteristic Area Under the Curve (ROC-AUC) score of 99.96%, all within a training time of 27.59 seconds. These results suggest that the Bagging model is highly effective and very efficient, making it a strong candidate for real-world IoT botnet detection. The model's high accuracy and low computational overhead make it a viable solution for real-world applications of Botnet detection, contributing significantly to the ongoing efforts of stakeholders in securing IoT networks against botnet threats.

Keywords: *Botnet Attack Detection, IoT networks, IoT Devices, IoT Environment, Bagging Model*

1. Introduction

In recent years, the Internet of Things has revolutionized various industries by enabling interconnected devices to communicate and share data. However, with this increased connectivity of IoT devices, network integrity can be compromised, particularly in the form of botnet attacks [1]. IoT device proliferation presents a number of benefits, but it also presents new security challenges because of the built-in constraints on processing speed, memory size, and battery life [2]. These devices become vulnerable to targeted attacks as a result, and when botnets enter them, the consequences can be dire, ranging from network outages to illegal data access and device hijacking. This emphasizes how critical it is to identify and stop botnet attacks in IoT networks as soon as possible.

IoT device proliferation presents several benefits, but it also presents new security challenges because of the built-in constraints on processing speed, memory size, and battery life [2]. These devices become vulnerable to targeted attacks as a result, and when botnets enter them, the consequences can be dire, ranging from network outages to illegal data access and device hijacking. This emphasizes how critical it is to identify and stop botnet attacks in IoT networks as soon as possible.

According to Yang, Wu, Yin and Li [3], the diversity of IoT devices presents a significant challenge for botnet detection. IoT networks consist of various devices with different functions, communication protocols, and computational capabilities. This heterogeneity complicates the development of a one-size-fits-all solution for botnet detection. Machine

Aremu B. L., Aro T. O., Saka K. K., Seriki A. A. and Raji R. O. (2024). Botnet Attack Detection in Internet of Things Using Selected Learning Algorithms. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 12 No. 2, pp. 18 - 26

learning models must be trained on diverse datasets that represent the wide range of devices and network behaviours found in IoT environments. Achieving this level of diversity in training data is difficult but necessary to develop robust detection systems capable of generalizing across different types of IoT devices [3]. Yanbu Wang, Linqing Liu and Chao Wang [4] emphasize the critical privacy and ethical challenges associated with deploying machine learning models for botnet detection in IoT networks. These challenges arise because machine learning algorithms necessitate vast datasets to achieve optimal performance, often involving the collection and processing of sensitive information from IoT devices. The inclusion of personal or confidential data in these datasets raises significant concerns regarding data privacy and ethical responsibility. Consequently, it is imperative to implement stringent data protection measures and adhere to ethical guidelines when developing and deploying such models. This ensures that while enhancing security against botnet threats, the privacy rights of individuals and the integrity of their data are not compromised. Balancing data privacy and security with the need for effective botnet detection presents a challenging issue that requires careful consideration and resolution. Moreover, the deployment of these models must comply with various regulatory frameworks and ethical guidelines to prevent misuse or unintended consequences. Balancing the need for effective security measures with the protection of user privacy and compliance with regulations is a critical aspect of developing and implementing machine learning-based botnet detection systems.

An important challenge is how to choose the most effective and efficient methods for detecting botnets in the IoT network traffic. Hence, there is a need for a comparative study to be carried out.

2. Related Works

Ali *et al*. [5] introduced a novel botnet identification system called ACLR, which integrates artificial neural networks (ANN), long short-term memory (LSTM), convolutional neural networks (CNN), and recurrent neural networks (RNN). Through experimentation using the UNSW-NB15 dataset, comprising nine types of attack that include 'Exploits', 'Normal', 'Generic', and others, the ACLR model achieves remarkable testing accuracy (0.9698) by capturing intricate attack patterns. Additionally, its generalizability and robustness are validated using K-fold cross-validation (k = 5) with an accuracy score of 0.9749. Moreover, the proposed model exhibits high performance in botnet detection, with ROC-AUC of 0.9934 and PR-AUC of 0.9950.

Mishra, Paliwal and Srivastava [6] proposed a new approach for botnet attack detection with the help of Deep Learning, specifically the Convolutional Neural Network-Pelican Optimization Algorithm (CNN-POA). The proposed approach was evaluated against the existing approach and other well-known algorithms for botnet attack detection using standard evaluation metrics. The simulation result shows the proposed CNN-POA as an effective and reliable method of IoT network intrusion; it outperforms a couple of other existing metaheuristic algorithms with 99.5% accuracy.

Wardana, Kołaczek, Warzyński and Sukarno [7] introduced a novel IDS approach that employed an ensemble of traffic models from nine heterogeneous IoT devices. Using DNN, they created a unique model and used it to train each type of device, and combined traffic predictions from these models using ensemble averaging to enhance detection accuracy. To detect botnet attacks in a wide range of IoT devices, they validated the effectiveness of this Intrusion Detection System using N-BaIoT dataset, demonstrating that the ensemble-averaged DNN achieves, on average, a value of 97.21% for accuracy, 91.41% for precision, 87.31% for recall, and 88.48% for F1-score.

Mishra, Paliwal and Srivastava [6] proposed a weighted stacked ensemble model that merges deep convolutional generative adversarial networks with bidirectional LSTM networks. This model underwent hyperparameter optimization and rigorous regularization. Evaluation of four publicly available IoT datasets demonstrated notable enhancements in standard performance metrics for binary and also for multiclass classification tasks. By leveraging L2 regularization to mitigate

overfitting, the proposed model achieved a reduction in generalization error by 0.005%. Notably, the model attained the following exceptional values of accuracy: BOT-IoT - 99.99%, IoT23 - 99.08%, UNSWNB15 - 99.82% and ToN_IoT datasets - 99.96%. These results are accompanied by improvements in Recall, Precision, and F1-score. The research presented an ensemble model comprising DCGAN and Bi-LSTM for network intrusion detection, demonstrating notable enhancements in performance metrics across various datasets.

Marshan, Nizar, Ioannou and Spanaki [8] provided a clear overview of a study examining the use of machine learning (ML) and deep learning (DL) techniques to address the issue of online harassment by detecting the severity of abusive comments on social media, given the increasing usage of social platforms. The models employed include Naive Bayes, Random Forest, Support Vector Machine, CNN, and Bi-LSTM. The researchers also used unigrams, bigrams, and word embeddings in feature set creation. They compared multiple ML and DL models, and the specific performance metrics used, such as accuracy, precision, recall, and F1 score. They especially highlighted Random Forest as the best performer.

A study by Sharma, Mansotra and Singh [9] underscores how the widespread deployment of IoT and Industrial IoT (IIoT) devices in previously secure areas can compromise critical infrastructure, such as intranet and database servers, due to the extensive data collection and monitoring capabilities of these devices. The paper evaluates potential security weaknesses that could lead to successful IoT attacks, highlighting prevalent Mirai botnet attack subtypes like ACK, SYN, Plain UDP, UDP flood, and Scan. Utilizing the N-BaIoT dataset, the researchers assessed the effectiveness of detection algorithms, where the CNN model demonstrated superior performance in accuracy, precision, recall, and F1-score compared to LSTM and GRU models.

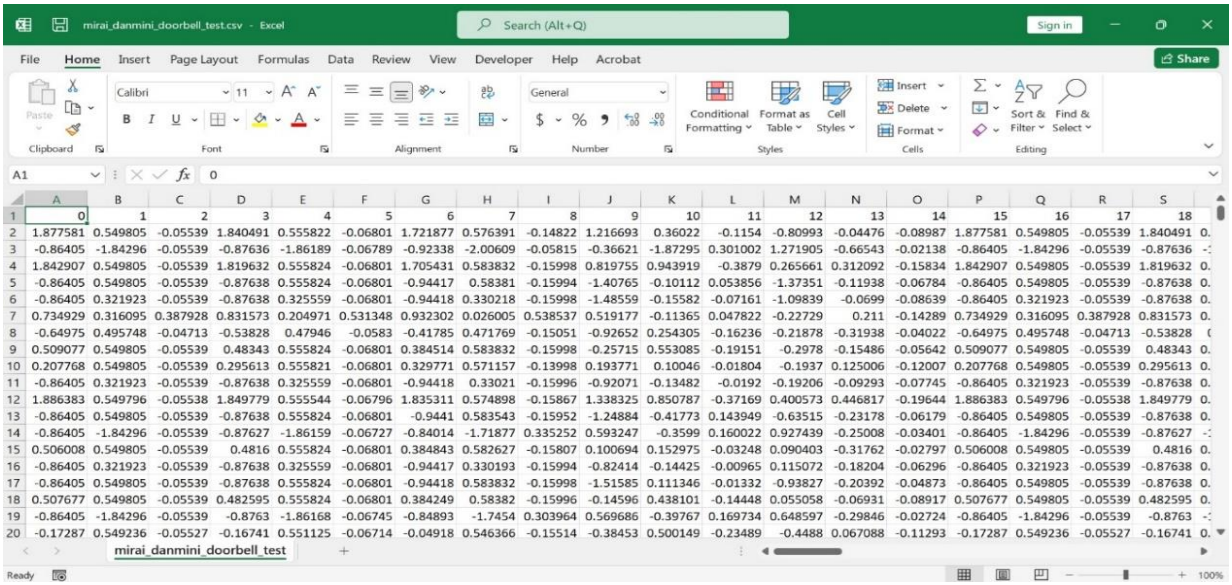
Shwartz-Ziv and Armon [10] focused on comparing the efficacy of traditional tree ensemble models, specifically XGBoost, against newer deep-learning models for handling tabular data. The methodology used is a rigorous comparison across various datasets, not only assessing performance but also considering the computational efficiency and tuning requirements of the models. The results indicated that XGBoost consistently outperforms newer deep learning models on the tested datasets, including those datasets previously used to advocate for these deep learning models. The researchers found that an ensemble of XGBoost and deep models outperform XGBoost alone.

3. Methodology

This section is an overall scheme, plan or structure conceived to aid the research in answering the raised research objective. To classify botnet attacks in IoT, an ensembled machine learning model is developed using Principal Component Analysis (PCA) and the following selected learners: Linear Regression, Logistic Regression, K-Nearest Neighbours (KNN), and Artificial Neural Networks (ANN).

3.1 Data Acquisition

Recent and comprehensive, high-dimensionality datasets that include both normal traffic and botnet traffic from IoT environments are gathered. Recent NB-IoT-23 datasets are gathered from the Kaggle repository, a publicly available source [11].
Dataset Link:
https://www.kaggle.com/datasets/saurabhshahane/anomaly-detection-using-deep-learning?resource=download.
A screenshot of the Mirai_danmini_doorbell_test dataset is shown in Figure 1.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2	1.877581	0.549805	-0.05539	1.840491	0.555822	-0.06801	1.721877	0.576391	-0.14822	1.216693	0.36022	-0.1154	-0.80993	-0.04476	-0.08987	1.877581	0.549805	-0.05539	1.840491
3	-0.86405	-1.84296	-0.05539	-0.87636	-1.86189	-0.06789	-0.92338	-2.00609	-0.05815	-0.36621	-1.87295	0.301002	1.271905	-0.66543	-0.02138	-0.86405	-1.84296	-0.05539	-0.87636
4	1.842907	0.549805	-0.05539	1.819632	0.555824	-0.06801	1.705431	0.583832	-0.15998	0.819755	0.943919	-0.3879	0.265661	0.312092	-0.15834	1.842907	0.549805	-0.05539	1.819632
5	-0.86405	0.549805	-0.05539	-0.87638	0.555824	-0.06801	-0.94417	0.58381	-0.15994	-1.40765	-0.10112	0.053856	-1.37351	-0.11938	-0.06784	-0.86405	0.549805	-0.05539	-0.87638
6	-0.86405	0.321923	-0.05539	-0.87638	0.325559	-0.06801	-0.94418	0.330218	-0.15998	-1.48559	-0.15582	-0.07161	-1.09839	-0.0699	-0.08639	-0.86405	0.321923	-0.05539	-0.87638
7	0.734929	0.316095	0.387928	0.831573	0.204971	0.531348	0.932302	0.026005	0.538537	0.519177	-0.11365	0.047822	-0.22729	0.211	-0.14289	0.734929	0.316095	0.387928	0.831573
8	-0.64975	0.495748	-0.04713	-0.53828	0.47946	-0.0583	-0.41785	0.471769	-0.15051	-0.92652	0.254305	-0.16236	-0.21878	-0.31938	-0.04022	-0.64975	0.495748	-0.04713	-0.53828
9	0.509077	0.549805	-0.05539	0.48343	0.555824	-0.06801	0.384514	0.583832	-0.15998	-0.25715	0.553085	-0.19151	-0.2978	-0.15486	-0.05642	0.509077	0.549805	-0.05539	0.48343
10	0.207768	0.549805	-0.05539	0.295613	0.555821	-0.06801	0.329771	0.571157	-0.13998	0.193771	0.10046	-0.01804	-0.1937	0.125006	-0.12007	0.207768	0.549805	-0.05539	0.295613
11	-0.86405	0.321923	-0.05539	-0.87638	0.325559	-0.06801	-0.94418	0.33021	-0.15996	-0.92071	-0.13482	-0.0192	-0.19206	-0.09293	-0.07745	-0.86405	0.321923	-0.05539	-0.87638
12	1.886383	0.549796	-0.05538	1.849779	0.555544	-0.06796	1.835311	0.574898	-0.15867	1.338325	0.850787	-0.37169	0.400573	0.446817	-0.19644	1.886383	0.549796	-0.05538	1.849779
13	-0.86405	0.549805	-0.05539	-0.87638	0.555824	-0.06801	-0.9441	0.583543	-0.15952	-1.24884	-0.41773	0.143949	-0.63515	-0.23178	-0.06179	-0.86405	0.549805	-0.05539	-0.87638
14	-0.86405	-1.84296	-0.05539	-0.87627	-1.86159	-0.06727	-0.84014	-1.71877	0.335252	0.593247	-0.3599	0.160022	0.927439	-0.25008	-0.03401	-0.86405	-1.84296	-0.05539	-0.87627
15	0.506008	0.549805	-0.05539	0.4816	0.555824	-0.06801	0.384843	0.582627	-0.15807	0.100694	0.152975	-0.03248	0.090403	-0.31762	-0.02797	0.506008	0.549805	-0.05539	0.4816
16	-0.86405	0.321923	-0.05539	-0.87638	0.325559	-0.06801	-0.94417	0.330193	-0.15994	-0.82414	-0.14425	-0.00965	0.115072	-0.18204	-0.06296	-0.86405	0.321923	-0.05539	-0.87638
17	-0.86405	0.549805	-0.05539	-0.87638	0.555824	-0.06801	-0.94418	0.583832	-0.15998	-1.51585	0.111346	-0.01332	-0.93827	-0.20392	-0.04873	-0.86405	0.549805	-0.05539	-0.87638
18	0.507677	0.549805	-0.05539	0.482595	0.555824	-0.06801	0.384249	0.58382	-0.15996	-0.14596	0.438101	-0.14448	0.055058	-0.06931	-0.08917	0.507677	0.549805	-0.05539	0.482595
19	-0.86405	-1.84296	-0.05539	-0.8763	-1.86168	-0.06745	-0.84893	-1.7454	0.303964	0.569686	-0.39767	0.169734	0.648597	-0.29846	-0.02724	-0.86405	-1.84296	-0.05539	-0.8763
20	-0.17287	0.549236	-0.05527	-0.16741	0.551125	-0.06714	-0.04918	0.546366	-0.15514	-0.38453	0.500149	-0.23489	-0.4488	0.067088	-0.11293	-0.17287	0.549236	-0.05527	-0.16741

Figure 1: Mirai_danmini_doorbell_test data extracted from NB-IoT-23 dataset with malicious and benign IoT network traffic.

This dataset was originally developed and has been specifically pre-processed for this study by Meidan *et al*. [11]. The NB-IoT-23 datasets were collected from five distinct IoT devices which are listed as follows:

a. Danmini doorbell (4 datasets)
b. Ecobee thermostat (4 datasets)
c. Philips baby monitor (4 datasets)
d. Provision security camera (4 datasets)
e. Samsung webcam (2 datasets)

The total number of datasets is 18 (9 training sets and 9 testing sets). The datasets include various IoT devices targeted by Gafgyt and Mirai botnets.

3.2 Data Pre-processing

The data was cleaned to remove noise and irrelevant information. This includes handling missing values, data normalization, and if necessary, encoding of categorical features. The following steps were taken:

a. Feature Selection: A broad set of features potentially relevant to network behaviour, such as packet size, rate, IP addresses, protocol type, and timing information are selected.
b. PCA Application: PCA is applied to the features to reduce the dimensionality. This step is crucial for improving the efficiency of the machine learning model by focusing on the most informative features and reducing computational cost.

3.3 Model Development

A supervised machine learning model is developed using Google Colab and Python to detect botnet activities. The following steps are taken:

a. Model Selection: Different supervised learning algorithms such as Linear Regression, Logistic Regression, KNN and ANN are evaluated to determine the best performer in terms of accuracy, precision, and computational efficiency.
b. Model Training: The model is trained using the training dataset. To handle large datasets like NB-IoT-23 and at the same time maintain lower computational demands, some machine learning techniques are inherently more efficient and practical, especially in constrained environments like those often found with IoT devices. The resampled training data is split into training and validation sets to evaluate model performance during training. The dataset is now divided into two parts: 80% is used for training, and 20% is used for testing.

The learning models are trained independently using the training dataset. Their predictions are aggregated by using the bagging ensemble method. Figure 2 illustrates the ensemble approach.

3.4 Performance Evaluation

The effectiveness and efficiency of the developed model are assessed to ensure it meets the expected standards for accuracy, speed, and adaptability in real-world scenarios, using the following model assessment metrics:

Accuracy

It measures the proportion of total correct predictions (both true positives and true negatives) out of all predictions made. It can be calculated from equation (1).

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (1)$$

TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

Precision

It assesses the accuracy of positive predictions and can be calculated from equation (2).

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (2)$$

Recall (Sensitivity)

It indicates the ability of the model to detect all actual positives, measuring the percentage of true positives identified correctly. It can be calculated from equation 3.

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (3)$$

F1-Score

It is the harmonic mean of precision and recall. It provides a balance between the two in cases where an equilibrium is needed for effective performance evaluation. It can be calculated from equation (4).

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

ROC

This metric evaluates the ability of the model to discriminate between the classes at various threshold settings. It can also be written as ROC-AUC or AUC-ROC and can be obtained from equation (5).

$$\text{ROC-AUC} = \int_0^1 TPR(FPR)dFPR \quad (5)$$

TPR = True Positive Rate (Recall)
FPR = False Positive Rate, and can be calculated from equation (6).

$$\text{FPR} = \frac{FP}{(FP+TN)} \quad (6)$$

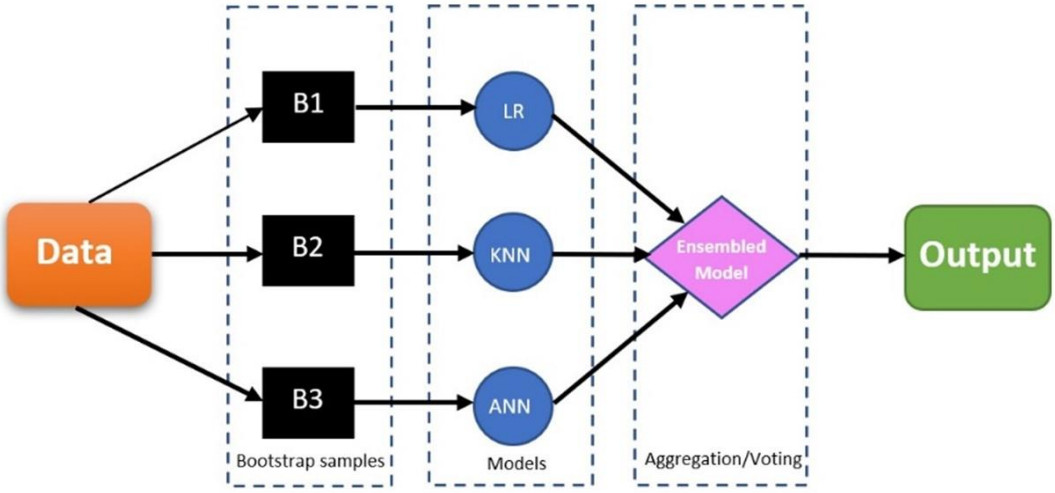


Figure 2: Illustration of bagging ensemble approach

4. Results and Discussion

4.1 Results

To efficiently detect botnet attacks, Jupyter Lab is used because it offers greater control over the local environment. Unlike Google Colaboratory, there are no usage limits. In addition, there is enhanced privacy and security. Jupyter Lab allows offline access, seamless integration with local tools and resources, and full utilization of system hardware. A core i5 system was used in this experiment.

The NB-IoT-23 datasets were used to train and evaluate the following machine learning models: Linear Regression, Logistic Regression, K-nearest neighbours (KNN), Artificial Neural Networks (ANN), and Bagging Classifier. The datasets include various IoT devices targeted by two botnets called Gafgyt and Mirai. The latest Python version 3.12.4 was installed in the system. The following libraries are also installed: pandas, scikit-learn, imbalanced-learn, matplotlib, and numpy.

Finally, results were compared in a table using a styled data frame. This is a table that compares the training times of all the learners and the voting classifier. Table 1 compares the performance metrics, expressed in percentage, and training times of the five models: Linear Regression, Logistic Regression, KNN, ANN and Bagging (Voting) classifier.

4.2 Discussion

For the accuracy metric, logistic regression, KNN and ANN have the highest value of 0.9988 and therefore outperform linear regression and bagging classifier. For the precision metric, KNN outperforms all the other four models. For the recall metric, ANN has the highest value of 1.0000 and therefore outperforms all the other four models. For the F1 Score metric, logistic regression, KNN and ANN have the highest value of 0.9988 and therefore outperform linear regression and bagging classifier.

For the ROC-AUC metric, logistic regression, KNN and ANN have the highest value of 0.9988 and therefore outperform linear regression and bagging classifier.

For training time, KNN outperforms all the other models and therefore has the highest computational efficiency, while the bagging classifier performs the least and therefore has the lowest computational efficiency.

Figure 3 is the visual representation of the ROC-AUC curve for the Bagging (Voting) classifier.

Comparison with Previous Works in the Literature

In this section, the results of the research are compared with the previous studies that were reviewed in the literature. Table 2 shows that the Bagging Classifier results are better than those obtained from the previous studies.

Table 1: Comparison of Performance Metrics and Training Times of All Algorithms

Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC	Training Time (s)
Linear Regression	91.37%	99.39%	83.26%	90.61%	91.37%	0.43
Logistic Regression	99.88%	99.85%	99.91%	99.88%	99.88%	3.34
KNN	99.98%	99.97%	99.99%	99.98%	99.98%	0.13
ANN	99.98%	99.96%	100.00%	99.98%	99.98%	23.64
Bagging Classifier	99.96%	99.93%	99.98%	99.96%	99.96%	27.59

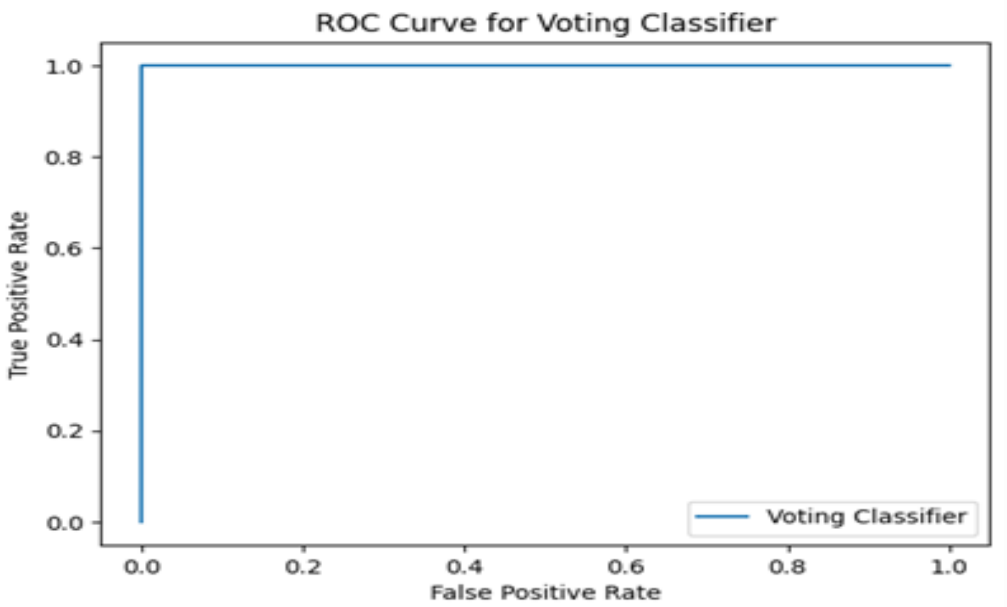


Figure 3: A Visual representation of the ROC-AUC curve for the Bagging classifier

Table 2: Comparison of Results with Results of Previous Studies

S/N	Author(s)	Title	Research Gap	Methodology	Result(s)
1	Ali *et al.*, [5]	Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment	An innovative botnet identification system, ACLR, which combined ANN, CNN, LSTM and RNN.	Model stacking is employed. It combines the outputs of various deep learning algorithms ANN, CNN, LSTM, and RNN for final prediction.	The ACLR model performed much better than all other deep learning models with an accuracy score of 0.9698. ANN showed the poorest performance with an accuracy score of 0.7568.
2	Thota & Menaka, [12]	Botnet detection in the Internet-of-things networks using a convolutional neural network with a pelican optimization algorithm	Addressing the problem of severely imbalanced network traffic data in the training sets used for botnet attack detection in IoT networks.	Designing and optimizing a Convolutional Neural Network (CNN) with a Pelican Optimization Algorithm (POA) to detect botnet attacks in IoT networks.	The proposed CNN-POA outperformed other existing metaheuristic algorithms (LGBA-NN, SDN, BMM and BiLSTM-CNN) with 99.5% accuracy.
3	Wardana *et al.*, [7]	Ensemble averaging deep neural network for botnet detection in heterogeneous Internet of	Detecting botnet attacks within heterogeneous IoT devices.	Development and training of individual DNN models. After training, the prediction outputs from each device-specific DNN model were aggregated. This aggregation was	The ensemble-averaged DNN achieved an average of 97.21% for accuracy, 87.31% for recall, 91.41% for precision and 99.48% for F1-score.

		Things devices.		done using the ensemble averaging method.	
4	Mishra, Paliwal and Srivastava [6]	Anomaly detection using deep convolutional generative adversarial networks in the Internet of Things	A weighted stacked ensemble model that merges deep convolutional generative adversarial networks with Bi-LSTM networks.	Employing an ensemble of neural networks for intrusion detection, utilizing datasets BOT-IoT, IoT-23, UNSWNB15, and ToN-IoT. BOT-IoT dataset This dataset captured various types of attacks for real-time analysis within the proposed framework.	The model achieved a reduction in generalization error by 0.005%. It attained an exceptional accuracy of 99.99% for BOT-IoT, 99.08% for IoT23, 99.82% for UNSWNB15, and 99.96% for ToN_IoT datasets, accompanied by improvements in Precision, Recall, and F1-score.
5	Aremu Bolakale Lawal (2024)	Botnet attack detection in the Internet of Things using selected learning algorithms	An ensemble machine learning model (Bagging) is developed using PCA, Linear Regression, Logistic Regression, KNN and ANN.	The NB-IoT-23 datasets were used to train the following machine learning models: Linear Regression, Logistic Regression, K-nearest neighbours (KNN), Artificial Neural Networks (ANN), and Bagging Classifier.	The model attained the following exceptional results: Accuracy: 99.96% Precision: 99.93% Recall: 99.98% F1 Score: 99.96% ROC-AUC: 99.96% Training Time: 27.59s

5. Conclusion

This study presented a comprehensive analysis and evaluation of various machine-learning techniques for detecting botnet attacks in the Internet of Things (IoT) environment. Following an extensive review of existing methodologies, key shortcomings were identified in the approaches previously used, particularly concerning their adaptability to different IoT scenarios and datasets. To address these issues, four learners (Linear Regression, Logistic Regression, KNN and ANN) were trained and evaluated, using PCA for dimensionality reduction of the NB-IoT-23 datasets. These datasets encompassed a range of IoT devices targeted by the Gafgyt and Mirai botnets. The learners were then ensembled to build a powerful Bagging Classifier of very high accuracy. The results obtained in this research demonstrate that the ensemble learning technique, which is a Bagging approach, is very effective for detecting botnet attacks in IoT environments. The Bagging model achieved an accuracy of 99.96%, precision of 99.93%, recall of 99.98%, F1 score of 99.96%, and a ROC-AUC score of 99.96%, with a training time of 27.59 seconds. These experimental results confirmed that:

a. PCA is reliable and acceptable for dimensionality reduction when using large NB-IoT-23 datasets, for training a machine learning model.
b. The bagging classifier is very effective and reliable for predicting botnet attacks because it achieves better results than some of the other models reviewed in the literature.

Contribution to Knowledge

The model's high accuracy and low computational overhead make it a viable solution for real-world applications of Botnet detection, contributing significantly to the ongoing efforts of stakeholders in securing IoT networks against botnet threats.

Recommendation

While this research has demonstrated the effectiveness of the Bagging model in detecting botnet attacks in IoT environments, there are several other exciting directions that future researchers can explore to build on these findings:

a. Exploring New Datasets: Although the NB-IoT-23 datasets provided a solid foundation for this study, testing the Bagging model on other datasets that include a broader range of IoT devices and different types of botnets could be a valuable next step. This would help determine how well the model adapts to diverse real-world scenarios.
b. Real-Time Detection: Moving the Bagging model from the research setting into a real-time intrusion detection system (IDS) could make a big impact. Future researchers could focus on tweaking the model to ensure it works efficiently in real-time, especially in large, complex IoT networks where speed and scalability are crucial.
c. Comparing with Deep Learning Models: While this study focused on traditional machine learning techniques like ANN and KNN, it would be interesting to see how the Bagging model stacks up against more advanced deep learning models such as CNNs and RNNs. Exploring this could reveal important insights into the balance between accuracy, complexity, and computational needs.

References

[1] Mashaleh, A. S., Ibrahim, N. F. B., Alauthman, M., Almseidin, M., & Gawanmeh, A. (2024). IoT Smart Devices Risk Assessment Model Using Fuzzy Logic and PSO. *Computers, Materials & Continua*, 78(2), 2245–2267. doi: 10.32604/cmc.2023.047323.

[2] Sai Kiran, K. V. V. N. L., Devisetty, R. N. K., Kalyan, N. P., Mukundini, K., & Karthi, R. (2020). Building an Intrusion Detection System for IoT Environment using Machine Learning Techniques. In *Procedia Computer Science* (Vol. 171, pp. 2372–2379). Elsevier B.V. doi: 10.1016/j.procs.2020.04.257.

[3] Yang, Z., Wu, S., Yin, Z. X., & Li, Y. L. (2020). Deep Anomaly Detection for Time-series Data in Industrial IoT: A Communication-Efficient On-device Federated Learning Approach. *IEEE Internet of Things Journal* (ArXivlD 2007.09712). doi: 10.1109/JIOT.2020.3011726.

[4] Wang, Y., Liu, L., & Wang, C. (2023). Trends in Using Deep Learning Algorithms in Biomedical Prediction Systems. *Frontiers in Neuroscience* (Vol. 17 - 2023). doi: 10.3389/fnins.2023.1256351.

[5] Ali, M., Shahroz, M., Mushtaq, M. F., Alfarhood, S., Safran, M., & Ashraf, I. (2024). Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment. *IEEE Access*, 12, 40682–40699. doi:10.1109/ACCESS.2024.3376400.

[6] Mishra, A. K., Paliwal, S., & Srivastava, G. (2024). Anomaly detection using deep convolutional generative adversarial networks in the Internet of things. *ISA Transactions*, 145, 493–504. doi:10.1016/J.ISATRA.2023.12.005.

[7] Wardana, A. A., Kołaczek, G., Warzyński, A., & Sukarno, P. (2024). Ensemble averaging deep neural network for botnet detection in heterogeneous Internet of Things devices. *Scientific Reports*, 14(1). doi: 10.1038/S41598-024-54438-6.

[8] Marshan, A., Nizar, F. N. M., Ioannou, A., & Spanaki, K. (2023). Comparing Machine Learning and Deep Learning Techniques for Text Analytics: Detecting the Severity of Hate Comments Online. *Information Systems Frontiers*, 1, 1–19. doi: 10.1007/S10796-023-10446-X/FIGURES/18.

[9] Sharma, A., Mansotra, Prof. V., & Singh, K. (2023). Detection of Mirai Botnet Attacks on IoT devices Using Deep Learning. *Journal of Scientific Research and Technology*, 174–187. doi: 10.5281/ZENODO.8330561.

[10] Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, 81, 84–90. doi: 10.1016/J.INFFUS.2021.11.011.

[11] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3), 12–22. doi: 10.1109/MPRV.2018.03367731.

[12] Thota, S., & Menaka, D. (2024). Botnet detection in the Internet-of-things networks using a convolutional neural network with the Pelican optimization algorithm. *Automatika*, 65(1), 250–260. Doi 10.1080/00051144.2023.2288486.