Date of publication SEP-30, 2024, date of current version AUG-23, 2024.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.23.3.3659

Beyond Performance Metrics: The Critical Role of Resource-Based Evaluation in Assessing IoT Attack Detectors

JEAN-MARIE KUATE FOTSO[1,4,5], FRANKLIN TCHAKOUNTE[1,5], ISMAEL ABBO[1,5], NAOMI DASSI TCHOMTE[2], CLAUDE FACHKHA[3]

[1]Department of Mathematics and Computer Science, Faculty of Science", University of Ngaoundéré, Po. Box 454, Cameroon
[2]Department of Computer Engineering, University Institute of Technology, University of Ngaoundéré Po.Box 455, Cameroon
[3]College of Engineering and IT (CEIT), University of Dubai, UAE
[4]National Committee for Development of Technologies/ Ministry of Scientific Research and Innovation, Yaoundé Po. Box 1457
[5]Cybersecurity with Computational and Artificial Intelligence, University of Ngaoundere, Cameroon

Corresponding authors: Jean-Marie Kuate Fotso and Franklin Tchakounte. (e-mails: jeanmarie.kuatefotso@gmail.com, tchafros@gmail.com).

ABSTRACT The proliferation of threats within the Internet of Things (IoT) environment is intensifying, largely due to the inherent limitations of this technology. The panoply of anti-threats based on artificial intelligence suffer from the complete embedment of models in limited resources. Tiny Machine Learning (TinyML) is presented as an opportunity in optimizing and selecting machine learning algorithms specifically tailored for intrusion detection systems (IDS) on limited-resource devices. This article addresses the challenges that must be overcome to enable the deployment of machine learning models on devices with constrained resources. In particular, it introduces additional indicators that could influence the algorithmic design of IoT models. Utilizing the PyCaret tool on the TON_IoT dataset, which encompasses nine distinct attacks, we developed and evaluated our approach for selecting the optimal algorithm from fourteen supervised learning models. The proposed tool, beyond the traditional six performance metrics, emphasizes resource consumption metrics, including memory, processor usage, battery life, and execution time – key considerations for TinyML in model refinement and selection. This study has identified less resource-intensive models suitable for developers in the design of IDS for IoT systems. We believe this research offers a foundational framework for the development of lightweight and efficient IoT vulnerability detection solutions.

KEYWORDS IoT; IDS; Tiny ML; Attacks; Cybersecurity.

I. INTRODUCTION

THE Internet of Things (IoT) represents a technology that is seamlessly integrated across various sectors of society [1]. However, in contrast to conventional computing systems, the IoT's heterogeneous nature introduces significant limitations that exacerbate its security vulnerabilities [2, 3]. The frequency of IoT attacks is increasing exponentially, with distinct variations in their manufacturing processes, usage patterns, and testing environments. A notable example is the emergence of the RapperBot malware in June 2022, a variant of the Mirai class that specifically targets Linux servers through SSH brute-force attacks [2], adding to the prevalence of attacks such as Broken Access Control, identified as the most common threat in the OWASP-IoT-2022 report [4]. The IoT ecosystem relies heavily on wireless sensor networks, which inherently lack robust security, making them susceptible to advanced attacks like the Google Attack, AWS DDoS attack, Mirai Krebs, and OVH DDoS attacks [3]. For companies utilizing smart devices, there is a pressing need for cybersecurity solutions to effectively manage the services these devices provide. The development of such security solutions typically involves the application of machine learning techniques, including Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Gaussian Naive Bayes (GNB). Although multiple solutions exist for addressing these security challenges, further optimization is required to ensure that these programs are both efficient and lightweight, enabling them to adapt to evolving IoT security threats [5-7].

Extensive efforts are undertaken to propose solutions for binary and multi-class classifications of IoT attacks, which are currently being developed across various layers, from gateway sensors to the broader Internet [8, 9]. The primary function of an Intrusion Detection System (IDS) is to safeguard IoT systems against unauthorized access, as failure to do so can compromise fundamental security principles [5]. IDSs play a critical role in real-time traffic analysis, enabling the detection of abnormal behaviors that arise from various IoT-specific attacks [10, 11]. To enhance understanding of IoT cybercrime, several studies have utilized the TON_IoT dataset, which provides heterogeneous data suitable to build machine learning based models for intrusion detection systems [12-14]. Such models that running in computers cannot be suitable in low-resources devices. TinyML comes into play to compress existing machine learning models for microcontrollers without relying on the cloud servers to run the algorithms. The following operations serve to reach these objectives for compressing models [5, 15, 16]. Knowledge Distillation (KD) evaluates whether a large model trained on a computer can be effectively scaled down for deployment on embedded systems. Pruning reduces dense neural networks to lighter, more efficient versions by eliminating redundant connections. Quantization minimizes the number of parameters, thereby reducing the model's size and complexity. Encoding, particularly through Huffman coding, can decrease the network's initial size by up to 49 times while preserving accuracy by using fewer bits for frequent weights and more bits for less frequent ones. Finally, Compilation involves transforming the program written in any language into a format that can be executed by the microcontroller.

The TON_IoT dataset adopted in this study reflects reality of IoT with heterogeneity in terms of attacks and target devices. Seven distinct attack types are included: scanning attack that involves reconnaissance activities where attackers gather network information by probing devices within the target system [6], denial of service (DoS) that aims to incapacitate a system by overwhelming it with excessive requests, rendering it unable to provide its intended services [17], ransomware that restricts user access to systems or personal data, demanding a ransom in exchange for restoring access, backdoor that refers to malware that enables unauthorized remote access to a compromised system, allowing attackers to control it covertly [8], injection attack which entails injecting malicious code into SQL queries, exploiting vulnerabilities within the application to execute unauthorized commands, Cross-Site Scripting (XSS) that involves injecting malicious scripts into web content, which are then executed in the victim's browser, leading to potential data theft or unauthorized actions [18] and password attack that comprises techniques used to bypass or exploit user authentication mechanisms to gain unauthorized access to accounts.

The selection of these attack types is substantiated by the established reputation and ranking of certain certified platforms used for evaluating vulnerabilities and IoT security solutions [19-21]. In contrast, other datasets featuring representative IoT scenarios lack diversity in their characteristics [22-25]. While other studies have proposed effective models with promising performance, they still require further experimentation to establish confidence in the employed learning models [26, 27]. The significant limitation of these works lies in the insufficient confidence elements within the learning models, which are likely influenced by factors such as network traffic flow (whether balanced or unbalanced), data quantity, evaluation metrics, and the number of features incorporated by the models. Additionally, we acknowledge the substantial progress made in data sampling techniques and the optimization of hyperparameters in machine learning [7, 28].

The objective of this study is to introduce additional indicators for assessing the reliability of supervised learning models in IoT intrusion detection. We aim to emphasize that selecting appropriate metrics is critical in evaluating machine learning models and that the effectiveness of a classification model is influenced by factors beyond the commonly used metrics. The quality of a model cannot be fully determined by performance metrics alone, even when those metrics have been optimized through hyperparameter tuning.

The proposed approach involves evaluating several machine learning techniques using six distinct performance metrics. We also emphasize the importance for developers to carefully consider the selection of high-performance models, as there are additional challenges beyond performance metrics that must be addressed – namely, data size, the number of features, and testing time, which we explore in this study.

The rest of the paper is organized as follows. Section 2 outlines the mathematical formulations used to evaluate the models during training. Section 3 details the experimental methodology. Section 4 describes the hardware configuration and provides a comprehensive account of the various evaluations conducted. The penultimate section discusses the results, highlights key observations from the experiments, and compares them with existing work. Finally, we identify limitations that will be the focus of future research.

II. BACKGROUND

To realize the innovative potential desired by IoT users, memory, power, and processing capabilities are critical elements that necessitate the application of Artificial Intelligence techniques for efficient utilization. Memory is essential for storing and processing data collected by IoT devices; without sufficient memory, data may be lost or become corrupted. Energy powers the operation of the network of connected devices, while the processor is responsible for managing information and making decisions [10].

Supervised learning provides effective tools for designing cybersecurity measures [29, 30], with machine learning models being particularly suited for analyzing IoT data in the context of intrusion detection. Typically, trust and model selection are guided by specific performance metrics [19, 31, 32].

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

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

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

$$F1Score = \frac{2 * TP}{2 * TP + FP + FN}, \quad (4)$$

$$Areaunderthecuve(AUC) = \int TPd(FP), \quad (5)$$

$$Kappa(K) = \frac{P_0 - P_e}{1 - P_e}, \quad (6)$$

$$P_0 = \frac{TP + TN}{TP + TN + FP + FN}, \quad (7)$$

$$Pe= P_1+P_2 , \tag{8}$$

$$P_1 = \frac{(TN + FP\ TN + FN)(TN + FN)}{(TN + FP + FN + TP)(TN + FP + FN + TP)},$$

$$P_2 = \frac{(FN + TP)(FP + TP)}{(TN + FP + FN + TP)(TN + FP + FN + TP)}.$$

In Equation (6), Po represents the observed agreement, while Pe denotes the expected agreement. These values indicate the classifier's performance relative to a classifier that makes random predictions based solely on class frequencies. The last two metrics in Equations (5) and (6) are particularly suitable for evaluating unbalanced datasets, unlike the four previous metrics [33, 34]. The Matthews Correlation Coefficient (MCC) is a more robust statistical measure, yielding a high score only when the classifier performs well across all four categories of the confusion matrix: true positives, false negatives, true negatives, and false positives.

The advancement of computer technology has transitioned from lightweight learning models to micro models based on TinyML. Consequently, the quality of service provided by IoT security solutions now hinges on additional factors such as network scalability and resource consumption, including memory, energy, processing power, and execution time [10]. These resources function in a highly interdependent manner; for instance, a high-performance processor may experience delays if memory access is restricted. Currently, some researchers focus on developing optimal resource management strategies and addressing attacks targeting these resources. They have also demonstrated that reducing these factors minimally impacts system performance and can enhance operational efficiency post-deployment [23, 35].

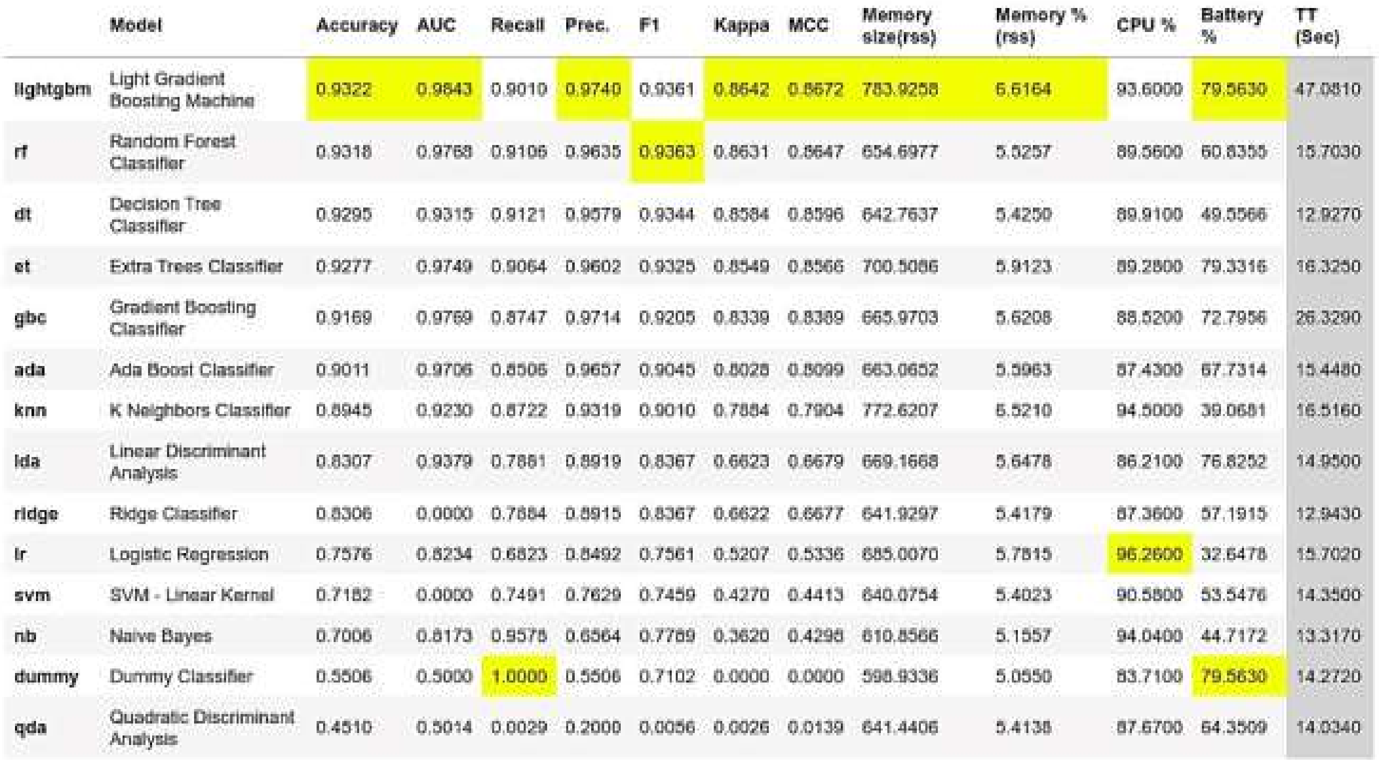
III. METHODOLOGY

We focused on analyzing a range of algorithms, including Light Gradient Boosting Machine, Random Forest, Decision Tree, Extra Trees, Gradient Boosting, AdaBoost, K-Nearest Neighbors, Linear Discriminant Analysis, Ridge Regression, Logistic Regression, SVM-Linear, Naive Bayes, Dummy Classifier, and Quadratic Discriminant Analysis to develop an intrusion detection system (IDS) for IoT. The objective was to assess their specific capabilities in detecting IoT vulnerabilities.

The objective is to provide a high-performance, lightweight solution suitable for embedded systems. Following the training and testing of our AI models, we evaluated their performance using metrics such as accuracy, AUC, recall, precision, F1 score, Kappa, and MCC. Considering the resource constraints of small devices, we also assessed the models based on their memory usage, processor demand, energy consumption, and training time.

The dataset selected for this study is TON_IoT, a recent and highly heterogeneous dataset comprising 2,540,044 attack events and 49 input features. We partitioned the data into one-third for testing and two-thirds for training.

Based on these evaluations, we present an IoT intrusion detection solution with a comprehensive analysis of its performance and detailed insights into its operational state post-deployment (see Fig. 1).



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	Memory size(rss)	Memory % (rss)	CPU %	Battery %	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.9322	0.9843	0.9010	0.9740	0.9361	0.8642	0.8672	783.9258	6.6164	93.6000	79.5630	47.0810
rf	Random Forest Classifier	0.9318	0.9768	0.9106	0.9635	0.9363	0.8631	0.8647	654.6977	5.5257	89.5600	60.8355	15.7030
dt	Decision Tree Classifier	0.9295	0.9315	0.9121	0.9579	0.9344	0.8584	0.8596	642.7637	5.4250	89.9100	49.5566	12.9270
et	Extra Trees Classifier	0.9277	0.9749	0.9064	0.9602	0.9325	0.8549	0.8566	700.5086	5.9123	89.2800	79.3316	16.3250
gbc	Gradient Boosting Classifier	0.9169	0.9769	0.8747	0.9714	0.9205	0.8339	0.8389	665.9703	5.6208	88.5200	72.7956	26.3290
ada	Ada Boost Classifier	0.9011	0.9706	0.8506	0.9657	0.9045	0.8028	0.8099	663.0652	5.5963	87.4300	67.7314	15.4480
knn	K Neighbors Classifier	0.8945	0.9230	0.8722	0.9319	0.9010	0.7884	0.7904	772.6207	6.5210	94.5000	39.0681	16.5160
lda	Linear Discriminant Analysis	0.8307	0.9379	0.7881	0.8919	0.8367	0.6623	0.6679	669.1668	5.6478	86.2100	76.8252	14.9500
ridge	Ridge Classifier	0.8306	0.0000	0.7884	0.8915	0.8367	0.6622	0.6677	641.9297	5.4179	87.3600	57.1915	12.9430
lr	Logistic Regression	0.7576	0.8234	0.6823	0.8492	0.7561	0.5207	0.5336	685.0070	5.7815	96.2600	32.6478	15.7020
svm	SVM - Linear Kernel	0.7182	0.0000	0.7491	0.7629	0.7459	0.4270	0.4413	640.0754	5.4023	90.5800	53.5476	14.3500
nb	Naive Bayes	0.7006	0.8173	0.9578	0.6564	0.7789	0.3620	0.4298	610.8566	5.1557	94.0400	44.7172	13.3170
dummy	Dummy Classifier	0.5506	0.5000	1.0000	0.5506	0.7102	0.0000	0.0000	598.9336	5.0550	83.7100	79.5630	14.2720
qda	Quadratic Discriminant Analysis	0.4510	0.5014	0.0029	0.2000	0.0056	0.0026	0.0139	641.4406	5.4138	87.6700	64.3509	14.0340

Figure 1. Execution of the dataset on the different models

IV. EXPERIMENTATION

A. HARDWARE SETUP

Some materials have been utilized during experiments: a Lenovo laptop equipped with a Linux 20.04 operating system, an Intel Core i7 processor, and 12 GB of RAM. The program code was developed using the PyCaret tool in conjunction with various Python libraries. Figure 1 illustrates the network traffic patterns from which the TON_IoT dataset was derived.

B. RESULTS ANALYSIS

The dataset utilized for supervised learning in this study is TON_IoT, which represents a new generation of heterogeneous datasets designed for Industry 4.0, the Internet of Things (IoT), and Industrial IoT (IIoT). It is specifically intended for evaluating the accuracy and effectiveness of various AI-based cybersecurity applications. The dataset is categorized into two main classes: normal (45,332 instances) and attacks (37,000 instances). Additionally, it provides a statistical overview of IoT data from Linux and Windows systems, as well as network data and connected devices.

Fourteen supervised learning techniques have been applied to our samples and evaluated using eight performance metrics, ranked in descending order of effectiveness.

Rating 1

This initial analysis results from training all models using PyCaret, a tool that facilitates the implementation of efficient, optimized, precise, and effective code. In this section, we focus exclusively on examining the performance metrics of the evaluated models.

Overall, the seven models ranked in descending order of performance are identified as the most optimal based on the experiment: LightGBM, Random Forest, Decision Tree, Extra Trees, Gradient Boosting Classifier, AdaBoost, and K-Nearest Neighbors. Conversely, seven other models exhibited metrics with zero values, including Ridge and SVM (with AUC = 0) and Dummy Classifier (with Kappa and MCC both at 0). An AUC of zero indicates poor model performance during training, while a zero MCC suggests inadequate classification quality.

Light Gradient Boosting Machine is considered the optimal model, despite being surpassed by two other models on the Recall and F1 score metrics. It outperforms in most metrics, including accuracy, ROC curve, F1 score, Kappa, and MCC. The remaining two metrics are shared between the Random Forest Classifier and Dummy Classifier models. Random Forest is also the second most efficient model, while the Dummy Classifier, despite its very low and sometimes zero performance values, is among the least effective in this experiment. According to the PyCaret tool, Quadratic Discriminant Analysis (QDA) ranks last, but a more objective analysis reveals that it exhibits better AUC performance compared to SVM and Ridge Regression and provides slightly better classification than the Dummy Classifier.

In alignment with our goals for a lightweight and compact security solution based on TinyML, relying solely on these metrics is insufficient. It is essential to validate the reliability of these metric values through considerations of the training process, execution time, and resource consumption (including memory, processor, and energy). These resources are critical to monitor during the refinement of IoT solutions.

Rating 2

In this evaluation, we investigated the effect of data size on the performance of learning models. As the data size decreased, the previously top-performing model exhibited reduced effectiveness on certain metrics. Specifically, at a data threshold of 500, the Extra Trees classifier outperformed the Light Gradient Boosting Machine, which was previously the best model and dropped to 5th place. At data sizes of 1000, 10,000, and 50,000, the Light Gradient Boosting Machine emerged as the leading model. However, the rankings of the other models were also affected, with the Gradient Boosting Classifier and the Random Forest Classifier securing the second place in these datasets.

Rating 3

Similar to evaluation 2, this assessment examines the impact of the number of features on model performance. Using the same test thresholds as before, we evaluated the models with a reduced number of features (half of the original). Each threshold highlighted a different model as superior. At the 10,000-feature threshold, the Light Gradient Boosting Machine (LightGBM) was the top performer. For the other thresholds, the Ridge Classifier and Decision Tree Classifier emerged as the best models, respectively. It is important to note that the number of features should not be chosen arbitrarily for testing, as their quantity can significantly influence model performance.

Rating 4

The final evaluation represents the primary objective of this work, focusing on optimizing programs for deployment on an IoT network. Beyond performance metrics, this evaluation includes an objective analysis of TinyML-specific factors such as memory usage, processor demand, energy consumption, and execution time. Resource consumption is measured for each of the 14 models. Models with higher resource requirements are less favorable for our goal, which is to identify the most efficient solution with minimal resource costs. Consequently, Light Gradient Boosting Machine (LightGBM) is not yet deemed the best option in this context.

V. DISCUSSION

This initial phase of the work addresses additional challenges in selecting and adopting models according to the TinyML principles for IoT applications. Beyond traditional performance metrics, we propose incorporating other critical factors for evaluating the suitability of models for embedded devices. These factors include resource consumption (memory, processor, and battery), execution time, quality and size of training data, and the selection and number of features.

TinyML emphasizes the need for programs to be less resource-intensive. Therefore, the primary focus is on optimizing models that are less demanding in terms of resources, ensuring their deployment and operation on microcontrollers are more efficient and secure. Figure 2 illustrates performance metrics for the models. If the program were designed for a computer with generally more powerful resources, the top seven models identified through experimentation – Random Forest (RF), Decision Tree (DT), Extra Trees (ET), Gradient Boosting Classifier (GBC), AdaBoost (ADA), and K-Nearest Neighbors (KNN), with LightGBM being the most powerful – would be preferable.

However, Figure 3 introduces another critical factor: execution times for each model. Notably, two of the top-performing models (LightGBM and GBC) exhibit very long execution times, which can pose significant challenges for embedded devices where real-time performance is crucial. Execution time is also related to resource consumption, such as processor time and memory access. Excluding LightGBM, which has a significantly higher execution time than the others, the average execution time across models is 15.6 seconds. This suggests that models with average execution times, such as RF, ET, ADA, KNN, IDA, LR, SVM, Dummy, and QDA, are more suitable. Four models, as shown in Figure 4, are considered less efficient, and this criterion warrants particular attention.

This experimentation revealed that the two models previously identified would be highly resource-intensive if selected. In the subsequent sections, we will evaluate resource consumption and examine its correlation with execution time.

Beyond the effects of data thresholds and feature selection on metric values, we observed that some of the top seven models (as shown in Figure 4) exhibit long execution times, which can affect energy consumption – an important factor distinct from memory and CPU usage. We will first analyze the consumption of individual resources and then conduct a comprehensive study of all three resources to draw final conclusions.

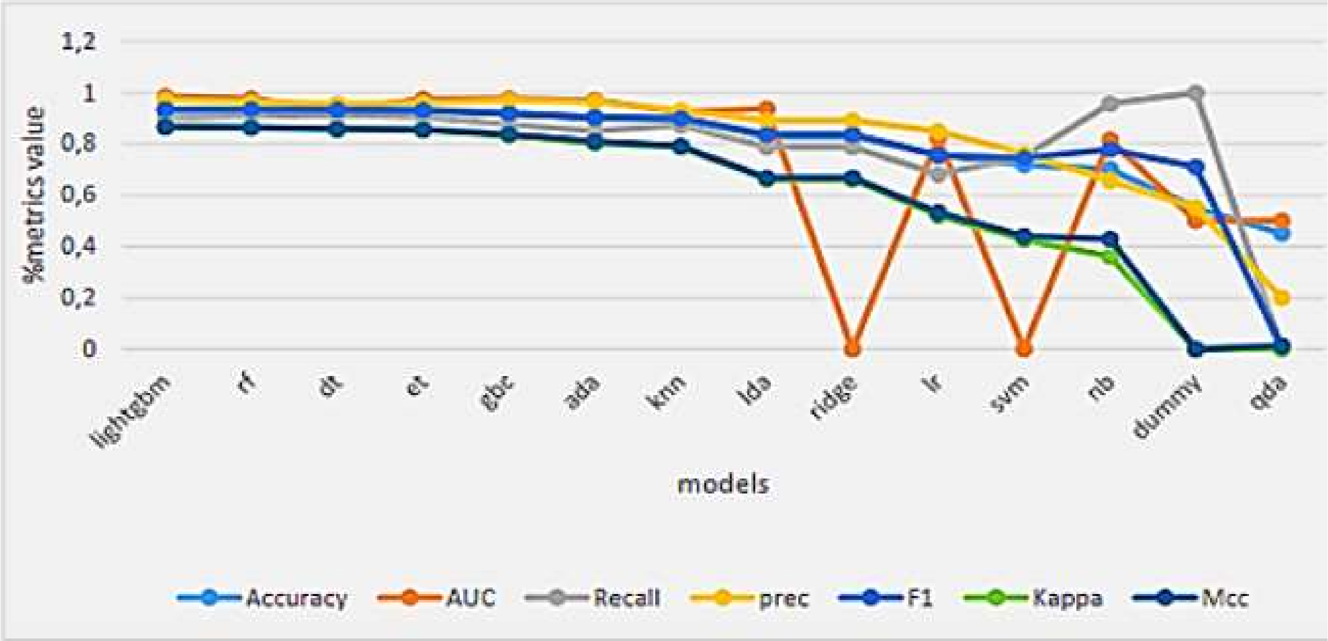


Figure 2. Model performance curve

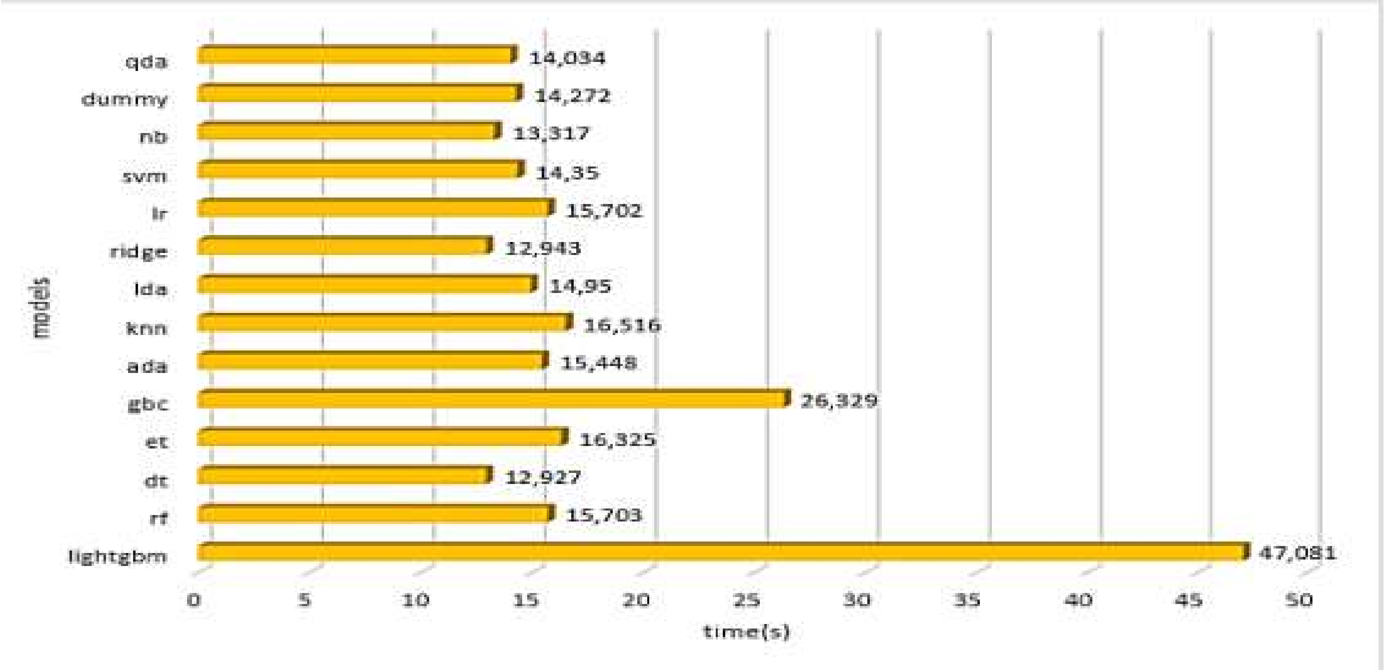


Figure 3. Model execution time

Memory: This metric reflects the percentage of RAM utilized by each model. Among the most efficient models identified earlier – LightGBM, Extra Trees (ET), and K-Nearest Neighbors (KNN) – we observe high memory consumption. In contrast, Logistic Regression (LR), although less efficient overall, also demonstrates relatively low memory usage. Other models exhibit similar memory usage, approximately 5 MB of RAM.

CPU: This metric measures the percentage of CPU usage, also referred to as processor time, during model testing. High CPU consumption is noted in two high-performance models – KNN and LightGBM – along with two less efficient models – Logistic Regression (LR) and Naive Bayes (NB), as indicated in Figure 4.

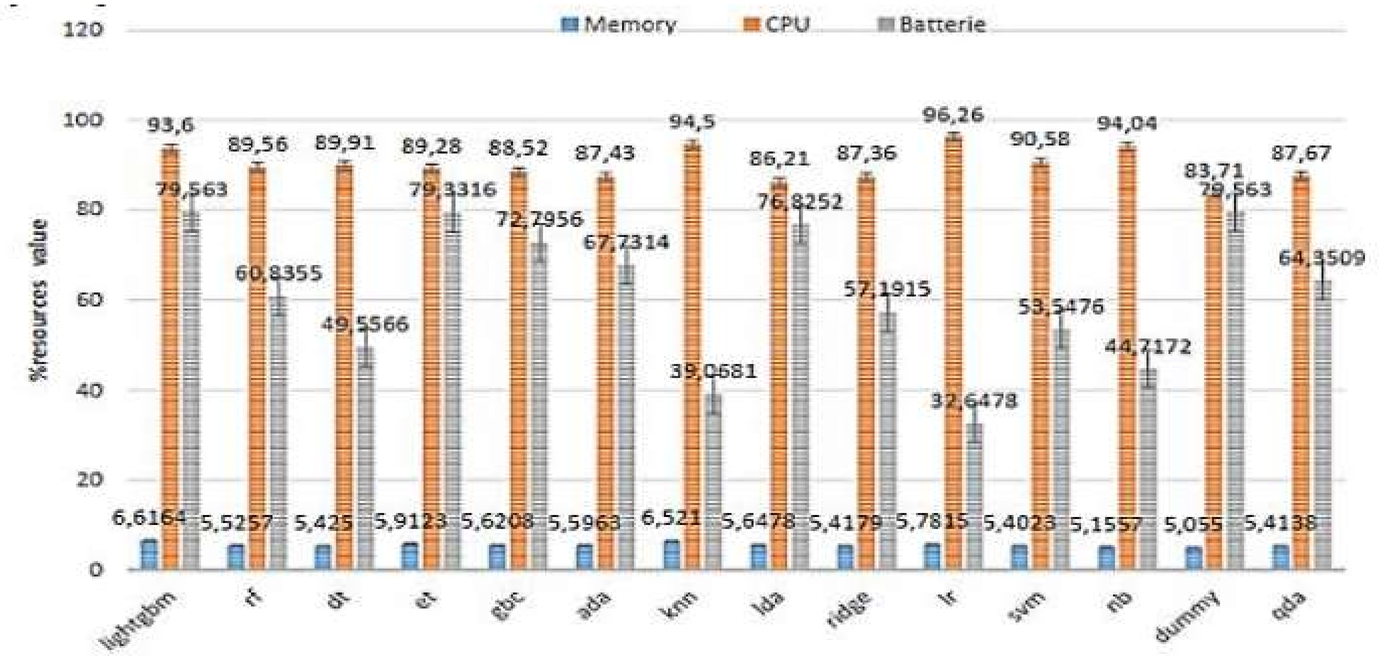


Figure 4. Resource consumption diagram

Battery: The psutil.sensors_battery() function provides information on the battery state. If no battery is present or if metrics cannot be determined, no value is returned. In this context, we measured the remaining battery power, expressed as a percentage, after training each model. Since all 14 models were run concurrently, a higher percentage of remaining battery power indicates lower resource consumption by that model. The experiment identified four models with high battery consumption: Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Naive Bayes (NB). Although LR and NB are less efficient, they are also notably power-intensive, with LR exhibiting the highest consumption.

The four resource factors analyzed are crucial for IoT solutions, as they ensure the system remains compact and suitable for deployment on a microcontroller. Initially, performance metrics highlighted seven top models: LightGBM, Random Forest (RF), Decision Tree (DT), Extra Trees (ET), Gradient Boosting Classifier (GBC), AdaBoost (ADA), and K-Nearest Neighbors (KNN). However, the study revealed that LightGBM, ET, and KNN could be excessively energy-consuming if used for IoT security tools. Consequently, given TinyML's requirements for fast, low-power, and resource-efficient algorithms, we recommend the following four models – RF, DT, GBC, and ADA – for vulnerability detection using the TON_IoT dataset.

VI. CONCLUSION

TinyML is an emerging technology that integrates machine learning into connected and autonomous devices. It offers advantages such as real-time data analysis, reduced data processing costs, enhanced local data security, and lower bandwidth requirements for remote server communication.

Nevertheless, optimizing machine learning models for TinyML remains a challenging task. It is crucial to establish further criteria for assessing models initially designed for conventional computing environments. This study aimed to evaluate machine learning models for network-based anomaly detection in IoT systems, specifically analyzing their performance on the TON_IoT dataset, identifying high-performance classifiers, and assessing resource utilization. The experiments revealed factors that could impact the stability of models when adapted for embedded devices, providing new insights into the evaluation of IoT security solutions. Although LightGBM emerged as the top-performing model in this study, our objectives suggest that four models – Random Forest (RF), Decision Tree (DT), Gradient Boosting Classifier (GBC), and AdaBoost (ADA) – are preferable due to their lower resource consumption.

Future research will focus on optimizing models by identifying the most relevant features using techniques such as Graph Neural Networks (GNN), Convolutional Neural Networks (CNN), or Recurrent Neural Networks (RNN), and employing hybrid sampling methods like SMOTEENN. Additionally, the study will aim to generalize the algorithms tested here by defining reference functionalities to facilitate a broader comparison of classifiers.

References

[1] B. Ankur and P. R. Udai, “Context-aware computing for IoT: History, applications and research challenges,” *Proceedings of the Second International Conference on Smart Energy and Communication*, January 2021, pp. 719-726. https://doi.org/10.1007/978-981-15-6707-0_70.

[2] L. Ravie, “New IoT RapperBot malware targeting Linux servers via SSH brute-forcing attack,” *The Hacker News*, 07 August 2022. [Online]. available at: https://thehackernews.com/2022/08/new-iot-rapperbot-malware-targeting.html.

[3] N. Paul, “Five most famous DDoS attacks and then some,” A10 Blog, *Network Security*, 4 May 2022. [Online]. Available at: https://www.a10networks.com/blog/5-most-famous-ddos-attacks/

[4] O. Amroussi, “Revisited: OWASP Top 10 Vulnerabilities 2022,” Vulnerability Management, 2024. https://vulcan.io/blog/owasp-top-10-vulnerabilities-2022-what-we-learned/.

[5] D. A. S. Lachit, “TinyML meets IoT: A comprehensive survey,” *Internet of things*, vol. 16, article 100461, 2021. https://doi.org/10.1016/j.iot.2021.100461.

[6] M. Nour, “A new distributed architecture for evaluating AI-based security systems at the edge: Network TONIoT datasets,” *Sustainable Cities and Society*, vol. 72, article 102994, 2021, https://doi.org/10.1016/j.scs.2021.102994.

[7] P. Marc-Oliver, A. François-Xavier, “All eyes on you: Distributed multi-dimensional iot microservice anomaly detection,” *Proceedings of the 2018 14th International Conference on Network and Service Management (CNSM)*, December 2018, pp. 72-80.

[8] M. Nour, “A systemic IoT-Fog-Cloud architecture for big-data analytics and cyber security systems: A review of fog computing,” arXiv preprint arXiv:1906.01055, 4 May 2019.

[9] K. V. Karthik, A. A. R. K. Nilofar, “Investigation on intrusion detection systems (IDSs) in IoT,” *International Journal of Emerging Trends in Engineering Research*, vol. 10, pp. 2347-3983, 2022. https://doi.org/10.30534/ijeter/2022/041032022.

[10] N. Schizas, A. Karras, C. Karras, “TinyML for ultra-low power AI and large scale IoT deployments: A systematic review,” *Network Cost Reduction in Cloud and Fog Computing Environments*, vol. 14, issue 12, article 363, 2022. https://doi.org/10.3390/fi14120363.

[11] M. Inês, S. João, R. Patrícia, S. Simão et al, “Host-based IDS: A review and open issues of an anomaly detection system in IoT,” *Future Generation Computer Systems*, vol. 133, pp. 95–113, 2022. https://doi.org/10.1016/j.future.2022.03.001.

[12] A. Abdullah, M. Nour, T. Zahir, M. Abdun, A. Adnan, “TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems,” *IEEE Access*, pp. 2169-3536, 2020.

[13] M. Nour, “The TONIoT Datasets,” UNSW Canberra, 2021.

[14] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac and P. Faruki, “Network intrusion detection for IoT security based on learning techniques,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671-2701, 2019. https://doi.org/10.1109/COMST.2019.2896380.

[15] Y. L., L. D. Xu, “Internet of Things (IoT) cybersecurity research: A review of current research topics,” *IEEE Internet of Things Journal*, vol. 6, no. 12, pp. 2103-2115, 2018. https://doi.org/10.1109/JIOT.2018.2869847.

[16] C. Shanzhi, X. Hui, L. Dake, H. Bo, W. Hucheng, “A vision of IoT: Applications, challenges, and opportunities with China perspective,” *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349-359, 2014. https://doi.org/10.1109/JIOT.2014.2337336.

[17] J. Ashraf, M. Keshk, N. Moustafa, M. Abdel-Basset, H. Khurshid, A. D. Bakhshi, R. R. Mostafa, “IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities,” *Sustainable Cities and Society*, vol. 72, article 103041, 2021, https://doi.org/10.1016/j.scs.2021.103041https://doi.org/10.1016/j.scs.2021.103041.

[18] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, “ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets,” *IEEE Internet of Things Journal*, vol. 9, pp. 485-496, 31 May 2021. https://doi.org/10.1109/JIOT.2021.3085194.

[19] U. T. Maurras, C. Yousra, B. Aliou et C. Raja, “Etude comparative des méthodes de détection d’anomalies,” R*evue des Nouvelles Technologies de l’Information*, pp.1-13, 2020.

[20] B. Govindraj, “Guide to OWASP IoT top 10 for proactive security,” 11 May 2021. [Online]: available at: https://www.appsealing.com/owasp-iot-top-10/

[21] I. Mukherjee, N. K. Sahu, S. K. Sahana, “Simulation and modeling for anomaly detection in IoT network using machine learning,” *International Journal of Wireless Information Networks*, vol. 30, pp. 173–189, 2022. https://doi.org/10.1007/s10776-021-00542-7.

[22] NVD, “National vulnerability database,” Understanding Vulnerability Detail Pages, 20 September 2022, [Online]. available at: https://nvd.nist.gov/vuln

[23] S. Raza, L. Wallgren, T. Voigt, “SVELTE: Real-time intrusion detection in the Internet of Things,” Ad Hoc Networks, vol. 11, issue 8, pp. 2661-2674, 2013. https://doi.org/10.1016/j.adhoc.2013.04.014.

[24] B. Blinowski, P. Piotrowski, “CVE based classification of vulnerable IoT systems,” *Theory and Applications of Dependable Computer Systems*,vol. 1173, pp. 82–93, 2020. https://doi.org/10.1007/978-3-030-48256-5_9.

[25] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, V. Sivaraman, “Classifying IoT devices in smart environments using network traffic characteristics,” *IEEE Transactions on Mobile Computing*, vol. 18, pp. 1745-1759, 2019. https://doi.org/10.1109/TMC.2018.2866249.

[26] N. Koroniotis, N. Moustafa, E. Sitnikova, B. P. Turnbull, “Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset,” *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019. https://doi.org/10.1016/j.future.2019.05.041.

[27] A. Hamza, H. H. Gharakheili, T. A. Benson, V. Sivaraman, “Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity,” Proceedings of the 2019 ACM Symposium on SDN Research SOSR'19, April 2019, pp. 36–48. https://doi.org/10.1145/3314148.3314352.

[28] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie and M. Palaniswami, “Labelled data collection for anomaly detection in wireless sensor networks,” *Proceedings of the 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, February 2011, pp. 269-274. https://doi.org/10.1109/ISSNIP.2010.5706782,

[29] O. Brun, Y. Yin, E. Gelenbe, “Deep learning with dense random neural network for detecting attacks against IoT-connected home environments,” *Procedia Computer Science*, vol. 134, pp. 458-463, 2018. https://doi.org/10.1016/j.procs.2018.07.183.

[30] Ö. A. Aslan, R. Samet, “A comprehensive review on malware detection approaches,” *IEEE Access*, vol. 8, pp. 6249-6271, 2020. https://doi.org/10.1109/ACCESS.2019.2963724.

[31] Amit, A. Dhingra, V. Sindhu, A. Sangwan, “A comprehensive review of DDoS attack, types and mitigation techniques in the Internet of Things network,” *International Journal for Modern Trends in Science and Technology*, vol. 8, pp. 72-79, 2022.

[32] C. Kolias, G. Kambourakis, A. Stavrou, S. Gritzalis, “Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 11, pp. 184-208, 2015. https://doi.org/10.1109/COMST.2015.2402161.

[33] Q.-D. Ngo, H.-T. Nguyen, V.-H. Le, D.-H. Nguyen, “A survey of IoT malware and detection methods based on static features,” *ICT Express*, vol. 6, n. 4, pp. 280-286, 2020. https://doi.org/10.1016/j.icte.2020.04.005.

[34] A. Cano, D. T. Nguyen, S. Ventura, K. Cios, “ur-CAIM: improved CAIM discretization for unbalanced and balanced data,” *Soft Computing*, vol. 6, p. 173–188, 2014. https://doi.org/10.1007/s00500-014-1488-1.

[35] C. Jun et C. Chi, “Design of complex event-processing IDS in Internet of Things,” *Proceedings of the 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*, April 2014, pp. 226-229, https://doi.org/10.1109/ICMTMA.2014.57.

***JEAN MARIE KUATE FOTSO** received a master's degree in distributed system and environment at the Faculty of Sciences of the University of Ngaoundéré since 2018, and since 2023 he has been enrolled in the same university where he works on the cybersecurity of objects connected.*

Since 2019, he has been recruited to the Ministry of Scientific Research and Innovation/National Committee for Technology Development.

Holder of several publications, he focuses his work on Artificial Intelligence, the Internet of Things and cybersecurity

***FRANKLIN TCHAKOUNTÉ** is a researcher with over 5 years of experience in cybersecurity and data science with a strong background in distributed systems. He obtained his Master of Science in Computer Engineering from the University of Ngaoundéré and then his PhD in Mobile Security from the University of Bremen. He is the author of books, book chapters, and over fourteen research articles in the field of cybersecurity and distributed systems. Pr.Dr-Ing. Franklin Tchakounté is a reviewer in IEEE journals as well as related conferences. He holds professional certifications in the field of network administration and has participated as a (senior) member of the ACM and the University Without Borders (UWB). It is dedicated to being a leader in cybersecurity in Africa.*

***Ismael ABBO** holds a master's degree in Master in Computer Engineering: Systems and Software in Distributed Environments from the University of Ngaoundéré on the theme Resource Management and Feature Optimization in the Detection of Attacks in IoT. He focuses his work on Artificial Intelligence, blockchain and the Internet of Things.*

***Naomi DASSI TCHOMTÉ** is a computer science Lecturer at the University of Ngaoundéré in Cameroon. She obtained her master's degree in Systems and Software in Distributed Environment in 2014 and her PhD in computer science in 2021 in the same University where she teaches. She has been a fellow of ACM-W in 2017, TWAS in 2018 and European Mathematical Society Simons for Africa in 2023. She is interested in Artificial Intelligence in general and in particular in Case-Based Reasoning and its Applications.*

***Claude FACHKHA** obtained his Master's in information systems security engineering and a Ph.D. in Electrical and Computer Engineering from Concordia University Canada in 2010 and 2015, respectively. he is a recipient of the prestigious Fonds de Recherche du Quebec – Nature et Technologies (FQRNT) award from Canada. He served as Technical Editor for the IEEE Communication Magazine. He has many publications. He is currently an assistant professor at the College of Engineering and IT, University of Dubai.*