Machine Learning-Based Histogram Boosting Approach for Attack Detection in IoT Networks

Muhammad Nouman, *Student Member, IEEE*, Nicola Cordeschi, *Member, IEEE*, Alessio Fascista, *Member, IEEE*
Department of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy
Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT)

***Abstract*—The attack detection is crucial in the Internet of Things (IoT) networks for preserving optimal network performance and system integrity. However, existing methods are insufficient to handle the large volume of available data, missing values, and high variance. Moreover, an integrated infrastructure is required to synchronize the various processes involved in attack detection. To address these challenges, we proposed an attack detection framework comprising three phases. The first phase addresses missing values, standardizes data, and maps attack classes based on shared features, reducing the dataset's complexity. In the second phase, oversampling and undersampling methods are sequentially employed to balance the dataset and reduce biases in highly imbalanced datasets. The third phase is a machine learning-based (ML) Histogram Gradient Boosting (HGB) model used to improve the prediction of attacks accurately and efficiently. This three-phase framework enhances ML models' performance and enables them to handle more complex classification tasks. Additionally, we investigate various ML and boosting models to increase the generalization capabilities of the classification. Simulation results conducted on actual IoT datasets indicate that the HGB model outperforms existing ML and boosting models, making it a suitable solution for industrial IoT applications.**

***Index Terms*—Machine Learning, Internet of Things, Cyberattacks.**

I. Introduction

In recent years, the IoT has rapidly grown and transformed the technological landscape. This evolution has invented new ways for humans to interact with technology. It enables various devices to communicate with one another and act based on shared information with minimal human intervention [1]. This process facilitates automation and the development of intelligent systems. IoT has a significant impact on industries such as smart home systems, medical devices, autonomous vehicles, and Industrial IoT (IIoT) [2]. However, security and reliability concerns arise as IoT networks grow and integrate with components. Addressing security and reliability concerns is crucial to ensuring reliable, resilient operation and protecting IoT systems.

IoT devices lack security mechanisms and are vulnerable to cyberattacks [3]. Adversaries exploit vulnerabilities to compromise data integrity and access controls. To effectively secure these devices, it is essential to implement advanced detection techniques and robust security protocols. Weak standards and ambiguous protocols increase the risks associated with IoT networks [4]. These weaknesses compromise the system's reliability and hinder large-scale IoT deployments from realizing their full potential. As IoT network traffic continues to increase, real-time threat detection becomes more complex. To address these challenges, intelligent detection methods have gained attention. Machine learning (ML) demonstrates strong potential in identifying previously unseen attack patterns [5]. However, ML performance depends on clean, balanced, and high-quality datasets. Imbalanced classes within a dataset introduce bias and degrade detection accuracy [6]. Furthermore, the dynamic nature of IoT traffic demands frequent dataset updates, which increase computational costs. The absence of standardized datasets further complicates model generalization across diverse environments.

To tackle security challenges in IoT systems, this study proposes an attack detection solution utilizing the CICIoT2023 dataset [7]. This dataset was chosen for its updated features and realistic attack scenarios from operational IoT networks. It provides comprehensive coverage of current threats relevant to practical environments. The main contribution of this study is organized into three distinct phases, as outlined below.

- The CICIoT2023 dataset Includes 33 different attack classes, resulting in a complex classification task that poses challenges for effective modelling and generalization. Many of these classes use the common features. In the first Phase, we preprocessed the dataset for ML models and mapped the 33 attak classes into 7 categories based on their common features.
- The CICIoT2023 is a multiclass dataset where instances are not distributed equally. The two attack classes have a significantly high number of instances, and the remaining are extremely low, including benign. This imbalance causes bias and increases the risk of the ML model's overfitting. In the second phase, we sequentially use an Adaptive Synthetic

(ADASYN) with Random Undersampling (RUS) to address the imbalance issue, which provides an equal instance for each class and avoids biases and overfitting toward one majority class. This approach effectively classifies the attacks and facilitates accurate modelling and generalization.
- Finally, the third phase employs the ML-based Histogram boosting model on CICIoT2023 to detect the different attacker and benign classes. The boosting model combines multiple weak learners and makes a strong prediction model, which reduces the risk of overfitting, while single ML models are prone to overfitting.

This study extends prior boosting-based intrusion-detection approaches by integrating an adaptive ADASYN–RUS resampling mechanism with HGB to enhance learning stability on the highly imbalanced CICIoT2023 dataset, while recognizing that such resampling slightly modifies natural traffic characteristics.

II. Related Work

The rapid advancement of IoT devices has brought significant benefits across various industries, including improved operational efficiency, real-time automation, and predictive maintenance [8]. However, this expansion has also introduced serious challenges, particularly regarding the security of interconnected IoT systems. As the number of connected devices continues to grow, networks must be capable of detecting malicious nodes and abnormal activities. Traditional detection methods such as rule-based, heuristic, and signature-based approaches rely on statistical and pattern recognition techniques to identify threats [9]. While these techniques provide a basic level of protection, they are ineffective against the evolving and dynamic nature of modern IoT attacks. To overcome these limitations, ML models have gained increasing attention for their ability to learn from data, adapt to new attack patterns, and enhance detection accuracy [10]. ML techniques offer flexibility and adaptability in identifying emerging threats. However, they also face challenges such as high computational requirements, dependence on labeled data, and difficulties in obtaining accurate annotations for training.

The evolving nature of malicious attacks presents major challenges, underscoring the need for rapid and adaptive IoT threat mitigation. Numerous studies have applied ML and deep learning (DL) models for IoT attack detection using both conventional and modern datasets [11]. A two-step clustering method has been proposed to enhance classification accuracy through advanced preprocessing, though it remains highly dependent on these steps and faces scalability issues [12]. Traditional ML models such as Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbour (KNN) perform well on small datasets but struggle with large-scale IoT data due to increased complexity [13]. Conversely, DL models such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) effectively capture complex and sequential patterns [14]. However their high computational cost and data imbalance hinder real-time deployment in resource-constrained environments. Recent studies introducing collision-aware access strategies have shown potential to reduce latency and access overhead in dense IoT scenarios, improving network resilience and detection efficiency [15], [16]. Despite these advances, ML-based ensemble learning remains the most promising direction for scalable and effective IoT attack detection, though further research is needed to reduce computational demands and support real-time processing [17].

III. System Model

This section discusses the proposed model for IoT network traffic classification. The proposed model has many essential sub-modules, including Dataset Description, Data Preprocessing, Data Resampling, and HGB as show in Fig. 1.

A. Dataset Description

The CICIoT2023 dataset provides a robust foundation for developing and evaluating IoT security solutions. It contains traffic from 105 devices and includes 33 distinct attack classes, which in this study are consolidated into seven broader categories: Denial of Service (DoS), Distributed Denial of Service (DDoS), brute force, spoofing, reconnaissance, web-based, and Mirai attacks. The original class distribution, shown in Table I, highlights a pronounced imbalance across categories. The dataset captures real-time activity under diverse threats, reflecting the complex dynamics of IoT networks while balancing generalization and forensic detail through the seven-class consolidation.

B. Data Preprocessing

Data preprocessing is the first step, which ensures the dataset is in the proper format for ML models. The preprocessing steps include handling missing values, applying label encoding, normalizing features to a specific range, and, most importantly, performing attack mapping to facilitate the classification process [18]–[21]. The CICIoT2023 dataset contains 33 distinct attack classes, each representing a unique attack behavior. However, several of these classes share similar characteristics and can be grouped under broader categories. To simplify the classification process, we consolidated the 33 attack classes into 7 general categories based on shared features. This reduction decreases dataset complexity, enhances model efficiency, and improves classification performance. Furthermore, this approach increases the practical applicability of ML models for IoT network attack detection by ensuring compatibility with real-world threat diversity.

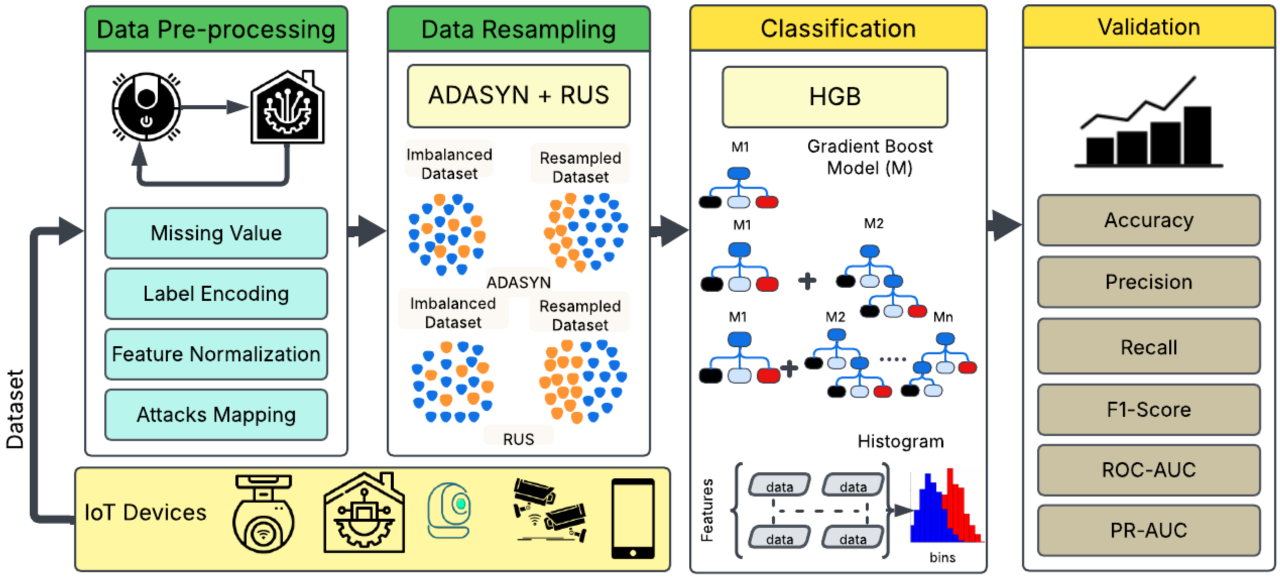

Fig. 1. Three-phase framework for IoT attack detection.

C. Data Resampling

Class imbalance is a major challenge in real-world datasets, where models tend to favor the majority class. To mitigate this, ADASYN and RUS were applied sequentially. ADASYN generates synthetic samples for minority classes by interpolating between samples with few same-class neighbors, improving generalization and reducing bias [22]. To ensure dataset reliability for small-sample classes, the ADASYN oversampling ratio was limited to prevent overfitting and maintain realistic traffic distributions. Subsequently, RUS removes excess samples from majority classes to achieve a balanced dataset [23]. Although RUS slightly alter the natural class distribution, this combined ADASYN–RUS strategy significantly enhances model fairness and stability across unseen samples. Table I summarizes the dataset composition before and after resampling, confirming improved class balance and equitable representation for all attack categories.

TABLE I
SAMPLE DISTRIBUTION AFTER RESAMPLING

Class	Original Instances	ADASYN	RUS
DDOS	47,729	47,729	47,002
DOS	11,120	47,002	47,002
Mirai	3,664	47,724	47,002
BENIGN	1,471	47,733	47,002
Recon	981	47,857	47,002
Spoofing	715	47,706	47,002
Web-based	25	47,724	47,002
Brute Force	18	47,730	47,002

D. Histogram Gradient Boosting

HGB is an optimized implementation of the Gradient Boosting (GB) algorithm that improves performance through histogram-based binning. Traditional GB techniques evaluate all possible split points for each feature, resulting in significant computational overhead. HGB overcomes this limitation by discretizing continuous feature values into a fixed number of intervals, substantially reducing candidate split points [24]. The first step of this process involves discretizing each continuous feature into quantile-based bins. In the standard implementation, each feature is divided into a predefined number of bins whose boundaries are derived from the empirical quantiles of the feature values. The smallest observed value defines the lower boundary of the first bin, and the largest value defines the upper boundary of the last bin, ensuring that all samples are represented. This quantile-based strategy automatically determines both bin width and boundary limits, preserving the feature distribution while accelerating model training. The GB framework within HGB iteratively adds decision trees to minimize the residual errors of preceding trees [25]. To optimize the loss function and update the model, HGB utilizes the gradient of the loss with respect to its predictions, as shown in Equation 1.

$$L(F) = \sum_{i=1}^{n} \ell(y_i, F(x_i)) \tag{1}$$

where $F(x_i)$ is the predicted value for sample i, and $\ell(y_i, F(x_i))$ is the loss function, often chosen as squared error, $(y_i - F(x_i))^2$. The model updates its predictions by adjusting subsequent weak learners to minimize the negative gradient of the loss function, as given in Equation 2:

$$r_i^{(m)} = -\left.\frac{\partial \ell(y_i, F(x_i))}{\partial F(x_i)}\right|_{F=F^{(m-1)}} \quad (2)$$

where $r_i^{(m)}$ is the residual for the m-th iteration. Each new model is trained to predict these residuals. The final prediction is the weighted sum of all weak learners, as shown in Equation 3:

$$F_M(x) = F_0(x) + \sum_{m=1}^{M} \eta\, f_m(x) \quad (3)$$

where $F_M(x)$ is the final prediction after M boosting rounds, $f_m(x)$ is the weak learner at iteration m, and η is the learning rate controlling each tree's contribution to the final prediction. The final prediction in HGB is obtained as a weighted sum of all weak learners. The model efficiently handles large, high-dimensional datasets while requiring less memory and computational cost than conventional Gradient Boosting methods. This makes HGB particularly suitable for real-time IoT traffic classification, where both speed and accuracy are critical.

IV. Simulation Results and their Discussion

Simulations are performed on the CICIoT2023 dataset to examine the performance of ML models. Python is used within the Google Colaboratory environment to implement the three-phase framework drawn in Fig. 1. The CICIoT2023 dataset is publicly available, and we used the file named "Merged52" [7]. The dataset is divided into two parts: 70% of the dataset is used for training, and the remaining 30% is used for ML model testing. The performance of multiple ML models was examined, including Naive Bayes (NB), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Stochastic Gradient Descent (SGD), Logistic Regression (LR), Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XGB), Categorical Boosting (CatBoost), and Light Gradient Boosting (LGB). These models were evaluated across several metrics, such as accuracy, precision, recall, F1 score metrics. The performance of all evaluated models is presented in Table II.

TABLE II
Performance comparison of ML models on the CICIoT2023 dataset.

Model	Accuracy	Precision	Recall	F1 Score
SGD	0.6264	0.6304	0.6264	0.6157
LR	0.6529	0.6551	0.6529	0.6445
NB	0.5098	0.6286	0.5098	0.4389
LDA	0.6069	0.6231	0.6069	0.6006
QDA	0.5652	0.6322	0.5652	0.5150
AdaB	0.2815	0.4878	0.2815	0.2364
XGB	0.8859	0.8901	0.8859	0.8848
HGB	0.9375	0.9419	0.9375	0.9368
CatB	0.8496	0.8539	0.8496	0.8470
LGB	0.9259	0.9315	0.9259	0.9253

In Fig. 2 represents the error metrics which include False Negative Rate (FNR), False Discovery Rate (FDR), False Positive Rate (FPR), and False Omission Rate (FoR), are compared across various ML models. HGB and LGB perform exceptionally well. Their low FPR and FoR show that they successfully reduce false positives and negatives and ensure precise identification of attack classes without excessive misclassification. AdaB has significantly higher FPR and FoR values compared to other models, indicating that it is prone to making many false positive errors. The NB model also has a relatively high FDR, continually misclassifying benign cases as attackers and generating false alerts. While the SGD and LR models perform moderately well across various criteria. These findings emphasize the importance of minimizing both false positives and false negatives to ensure accurate and effective attack detection in IoT networks.

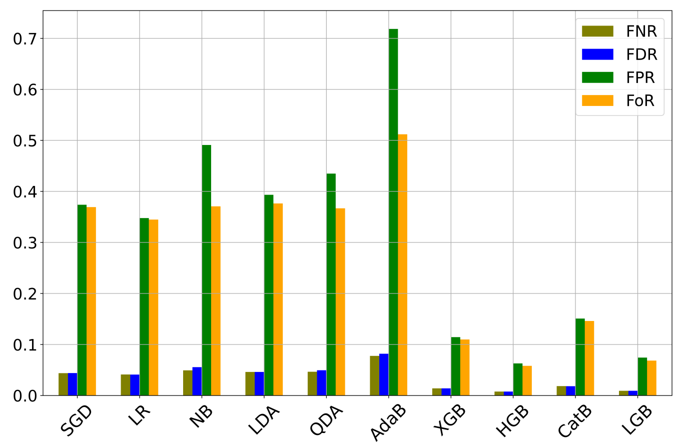

Fig. 2. Comparison of error metrics across different ML models.

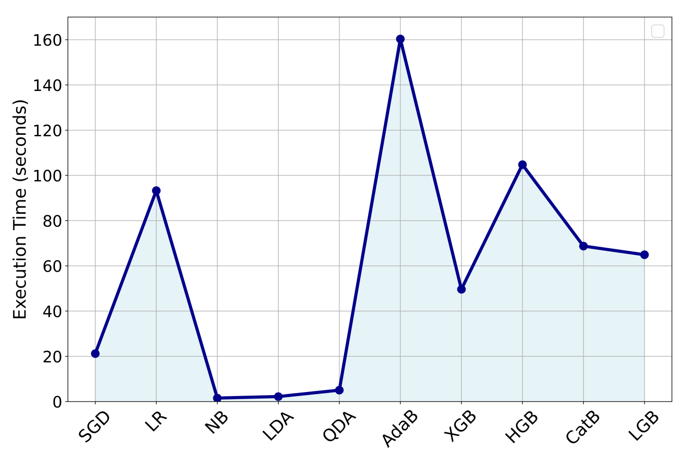

Fig. 3. Execution time comparison of different ML models.

Time efficiency is essential in real-world applications that require rapid processing, especially in resource-constrained environments. Fig. 3 illustrates the execution times of each ML model, which shows that AdaB has the longest execution time of the other models. The LR, SGD, and NB maintain shorter execution times. However, their prediction results are not satisfying, which means the models are not learning and predicting well. In contrast, XGB, HGB, CatB, and LGB depict moderate execution times and obtain satisfying results on unseen data. Understanding these variations in execution time is crucial when selecting the appropriate model. It is important to balance computational resources, time efficiency, and model performance according to the application's specific needs. Such comparisons help select models that deliver strong performance and meet the practical requirements for deployment in resource-constrained environments.

V. Conclusion and Future Work

Securing IoT networks is essential due to their numerous weaknesses. ML models help to identify malicious behaviours within network flows. However, these models still struggle to identify malicious behaviour due to training on inadequate data. To address these challenges, this study presents a three-phase framework for IoT attack detection that includes preprocessing and mapping 33 attack classes into 7 broader categories, mitigating class imbalance to improve model generalization, and employing the HGB model to enhance detection accuracy. The proposed framework integrates preprocessing, resampling, and classification into a unified pipeline and demonstrates strong performance on real-world datasets. Although the HGB model achieved the highest detection accuracy, it incurred increased computational complexity. Future research will investigate bagging and stacking approaches combined with deep learning models to design lightweight and scalable solutions for practical IoT deployments.

References

[1] Riva, G. (Ed.). (2005). Ambient intelligence: the evolution of technology, communication and cognition towards the future of human-computer interaction (Vol. 6). IOS press.

[2] Chataut, R., Phoummalayvane, A., & Akl, R. (2023). Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0. Sensors, 23(16), 7194.

[3] Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., & Kim, K. J. (2019). A survey of deep learning-based network anomaly detection. Cluster Computing, 22(Suppl 1), 949-961.

[4] Alotaibi, A., & Rassam, M. A. (2023). Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense. Future Internet, 15(2), 62.

[5] Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in big data. Journal of Big Data, 5(1), 1-30.

[6] Pattar, S., Buyya, R., Venugopal, K. R., Iyengar, S. S., & Patnaik, L. M. (2018). Searching for the IoT resources: Fundamentals, requirements, comprehensive review, and future directions. IEEE Communications Surveys & Tutorials, 20(3), 2101-2132. .

[7] University of New Brunswick, "IoT Dataset 2023," *University of New Brunswick*, 2023. [Online]. Available: https://www.unb.ca/cic/datasets/iotdataset-2023.html (Last accessed: April 29, 2025).

[8] Molaei, F., Rahimi, E., Siavoshi, H., Afrouz, S. G., & Tenorio, V. (2020). A comprehensive review on internet of things (IoT) and its implications in the mining industry. American Journal of Engineering and Applied Sciences, 13(3), 499-515.

[9] Chinnasamy, R., & Subramanian, M. (2023). Detection of Malicious Activities by Smart Signature-Based IDS. In Artificial Intelligence for Intrusion Detection Systems (pp. 63-78). Chapman and Hall/CRC.

[10] Sarker, I. H., Khan, A. I., Abushark, Y. B., & Alsolami, F. (2023). Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. Mobile Networks and Applications, 28(1), 296-312.

[11] Gudivada, V., Apon, A., & Ding, J. (2017). Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. International Journal on Advances in Software, 10(1), 1-20.

[12] Gheni, H. Q., & Al-Yaseen, W. L. (2024). Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 9, 100673.

[13] Boateng, E. Y., Otoo, J., & Abaye, D. A. (2020). Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: A review. Journal of Data Analysis and Information Processing, 8(4), 341-357.

[14] Millham, R., Agbehadji, I. E., & Yang, H. (2020). Parameter tuning onto recurrent neural network and long short-term memory (RNN-LSTM) network for feature selection in classification of high-dimensional bioinformatics datasets. In Bio-inspired algorithms for data streaming and visualization, big data management, and fog computing (pp. 21-42). Singapore: Springer Singapore.

[15] Cordeschi, N., De Rango, F., & Tropea, M. (2021). Exploiting an optimal delay-collision tradeoff in CSMA-based high-dense wireless systems. IEEE/ACM Transactions on Networking, 29(5), 2353-2366.

[16] Cordeschi, N., Zhuang, W., Tafazolli, R., & Gao, Y. (2023). Optimal random access strategies for trigger-based multiple-packet reception channels. IEEE Transactions on Mobile Computing, 23(3), 2303-2320.

[17] Abu Al-Haija, Q., & Al-Dala'ien, M. A. (2022). ELBA-IoT: An ensemble learning model for botnet attack detection in IoT networks. Journal of Sensor and Actuator Networks, 11(1), 18.

[18] Li, C. (2019). Preprocessing methods and pipelines of data mining: An overview. arXiv preprint arXiv:1906.08510.

[19] Number Analytics, "Mean Imputation Techniques: Handling Missing Data," *Number Analytics Blog*, 2025. [Online]. Available: https://www.numberanalytics.com/blog/mean-imputation-techniques-handling-missing-data (Last accessed: April 29, 2025).

[20] Bolikulov, F., Nasimov, R., Rashidov, A., Akhmedov, F., & Young-Im, C. (2024). Effective methods of categorical data encoding for artificial intelligence algorithms. Mathematics, 12(16), 2553.

[21] Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. Applied Soft Computing, 97, 105524.

[22] Dey, I., & Pratap, V. (2023, March). A comparative study of SMOTE, borderline-SMOTE, and ADASYN oversampling techniques using different classifiers. In 2023 3rd international conference on smart data intelligence (ICSMDI) (pp. 294-302). IEEE.

[23] Tahir, M. A., Kittler, J., & Yan, F. (2012). Inverse random under sampling for class imbalance problem and its application to multi-label classification. Pattern Recognition, 45(10), 3738-3750.

[24] Nouman, M., Qasim, U., Nasir, H., Almasoud, A., Imran, M., & Javaid, N. (2023). Malicious node detection using machine learning and distributed data storage using blockchain in WSNs. IEEe Access, 11, 6106-6121.

[25] Terven, J., Cordova-Esparza, D. M., Romero-González, J. A., Ramírez-Pedraza, A., & Chávez-Urbiola, E. A. (2025). A comprehensive survey of loss functions and metrics in deep learning. Artificial Intelligence Review, 58(7), 195.