International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

Optimizing IoT and IIoT Intrusion Detection: A Comparative Analysis of Ensemble and Deep Learning Techniques

Mohamed Koroma (MSc), Zidida Demekhaly Turay (MSc), Mohamed Syed Fofanah (PhD),
Maurice Sesay (PhD)
School of Technology, Department of Computer Science and Information Technology,
Njala University, Sierra Leone, West Africa.

Abstract - The use of Internet of Things (IoT) and Industrial Internet of Things (IIoT) has greatly enlarged the attack surface of the cyberspace, exposing the key systems to risks. This is a threat environment that requires smart intrusion detection systems (IDS) that overcome the shortcomings of traditional and signature security programmes.

Despite the potential of machine learning (ML) to be the best option of IDS, the comparison of the modern algorithms to the problem of IoT specifics is not yet conclusive. Neither a resource-limited hardware nor a high-accuracy ideal model has been uncovered to provide both high-accuracy and operational efficiency.

The proposed research is aimed at identifying the best ML model to use in the detection of IoT/IIoT intrusions. It aims to: i) evaluate seven different algorithms including tree-based ensemble and deep learning networks; iii) preprocess the Edge-IIoTset dataset with SMOTE to address the problem of class imbalance; iii) benchmark the performance involving standard metrics; and iii) discover the most important features in detection of threats.

The research is conducted in a systematic experimental manner with the use of the Edge-IIoTset dataset. Using SMOTE to preprocess, seven models are trained and assessed based on accuracy, precision, and F1-score: Logistic Regression, Random Forest, XGBoost, CatBoost, LightGBM, ANN and CNN. An analysis of the most performing features is then carried out. Gradient boosting models were the best models to use, with XGBoost having the highest accuracy (95.07%), precision (95.46%), and F1-score (95.14%). The TCP flags and application-layer data were found to be the major intrusion indicators analysed. XGBoost and LightGBM offered the most desirable compromise between the strength and power of detection and computational efficiency.

The research establishes the use of gradient boosting algorithms namely XGBoost and LightGBM are very effective in addressing IDS in IoT/IIoT setting, best combining high-performance, efficiency and interpretability. The findings provide practitioners with a clear evidence-based guideline on how to choose IDS models to provide them with effective cybersecurity without having to overstretch their limited processing resources. The next work should confirm these models in test-beds in real-time and investigate the construction of lightweight hybrid systems in new areas such as 5G and vehicular networks.

Keywords: Cybersecurity, Industrial IoT, Machine Learning Comparison, Anomaly Detection, XGBoost.

1. INTRODUCTION

The extensive adoption of the Internet of Things (IoT) and the Industrial Internet of Things (IIoT) has essentially redefined the structures of operations in domestic, commercial, and industrial spheres. These interconnections of sensors, actuators, and intelligent devices facilitate a high level of automation, real-time monitoring, and management in the form of analytics (Kikissagbe and Adda, 2024). This high rate of growth, however, has also resulted in a vast vulnerability environment. Cyber-attacks are often tempted to IIoT deployment, which is often part of the critical infrastructure such as smart energy grids and water purification plants. Such security breaches can bring more consequences than financial loss because they may cause a massive collapse of operations, system impairment, and severe threats to the common good (Ferrag et al., 2022; AlDosari, 2017). There are a number of inherent factors that contribute to security restrictions in the presence of IoT/IIoT. One of the central issues is the low cost and functionality nature of many machines, which due to their low cost, have limited computational power, memory, or power to support in-built defensive measures. Moreover, the high heterogeneity of hardware, communication protocols (MQTT, HTTP and Modbus TCP), and software versions creates a very complex ecosystem that is difficult to protect in a standardised way. A high percentage of outdated systems continue to run without security patches since they can be susceptible to the known

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

threats (Ni & Li, 2024). These vulnerabilities of the system are capitalised by using various intrusion mechanisms. Phishing campaigns, exploitation of software vulnerabilities and unsecured network port attacks are all common methods used by malicious actors to create an initial presence. After gaining access, they spread to other devices and servers in the network and instal malicious software, steal sensitive data, or execute ransomware activities. The changing and dynamic nature of threats also indicates the ineffectiveness of the traditional, fixed security boundaries. The compromise of one element in the IoT, such as a network camera or an environmental sensor, can become a critical point of attack by an attacker. This first tradeoff enables lateral movement without authorization through the network and may lead to the theft of confidential data, manipulation of industrial processes, or massive lack of service. This cascading failure model demonstrates the inadequacy of perimeter-based security and the high importance of the intelligent intrusion detection systems which are aimed at detecting malicious behaviour inside the network communications. Some of the initial access techniques used by attackers are targeted phishing and exploitation of unpatched software vulnerabilities. Once they gain access, they use strategies like lateral movement and credential theft to consolidate their access and finally implement their ultimate goals, and in most of the cases this is stealing data or ordering a ransom as demonstrated in figure 1.

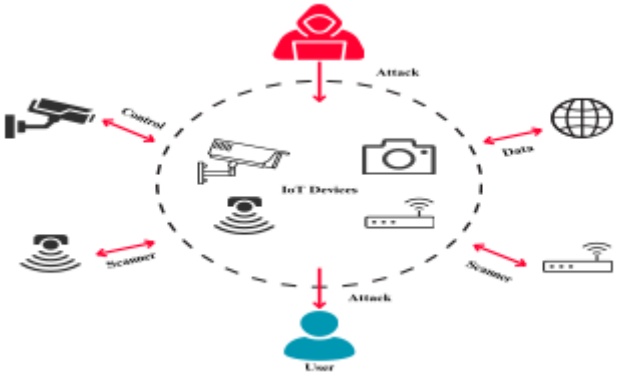

Figure 1: Cybersecurity threat on an IoT ecosystems.

Example of a malicious actor (red) that takes advantage of the vulnerabilities of cross-linked IoT devices (e.g., cameras, sensors, routers), which shows how lateral movement and system-wide compaction are possible in a network. The security of an IoT ecosystem is as safe as the least secure device. Single component that is not adequately safeguarded can become an easy target to an attacker, even to the sophisticated system. After being inside, an attacker may steal sensitive information, interfere with it or disturb major processes. The damage that such attacks can make is potentially severe, including loss of personal privacy or unauthorised tracking and disruption of services (both harmful to the IOT ecosystem). Protecting against these threats will demand the proactive security position. Encryption of data over the network, continuous surveillance of network traffic in order to identify suspicious activity, and hardware and software compatibility with the current security patches should not be excessive requests. These are the necessary measures to ensure the confidentiality and integrity of the data in the IoT systems. Attacks normally take advantage of a certain vulnerability. These include open network ports, unpatched software vulnerabilities, insider threats and exploitation of known security vulnerabilities which are represented.

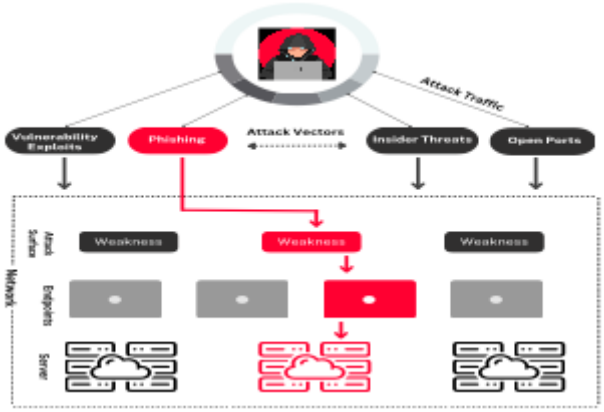

Figure 2: Cyber attackers exploit various attack vectors

Examples of typical paths of Cyber-attacks like phishing, exploit vulnerability and open port to gain access to the network, move laterally and access critical assets will demonstrate that layered defence is needed. in figure 2. The Internet of Things (IoT)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

network has a crucial element of security posture that is determined by its most vulnerable component. One poorly secured device can be used as an entry point by malicious actors that will destroy the integrity of the entire infrastructure. Once a hacker is allowed to access it, they are able to intercept and manipulate sensitive data or even shut down critical services which can be very disastrous in terms of individual privacy invasion, unauthorised spying and downtime. It is therefore important that a proactive and defensive security strategy is implemented. Basic security strategies such as the use of strong data encryption, ongoing analysis of network traffic to identify anomalies that may indicate a breach of network security, and regular application of security patches will be a minimum groundwork in maintaining data confidentiality and integrity of systems. In current cyber-intraudits, it is often based on the well-reported vulnerabilities, such as public network services, unpatched software vulnerabilities and insider access, as discussed in Figure 2. After the first compromise, the opponents usually utilise the lateral movement techniques, implementing options like credential misuse and targeted phishing to extend their influence. This facilitates the execution of malicious code, organised data mining, and the installation of ransomware at the network endpoints. The most common points of entry to such intrusions are persistent vulnerabilities that may be configuration mistakes or outdated software. As a result, the changing and dynamic nature of the modern cyber-threats provoke the need to have a security model that systematically handles vulnerabilities in all network layers. The scope and complexity of these attacks keep showing how ineffective the old signature-based intrusion detection systems are as they are frequently too sluggish and inadequate in their scope to handle the necessary level of protection. Contrary to the above, the use of Machine Learning (ML) offers a new approach to the development of a dynamic IDS. ML models can be used to analyse the complex network traffic patterns and differentiate between legitimate and potential attacks. Their capacity to handle high-dimensional data and adapt to new threats automatically through systems like ensembles methods or deep learning render them highly suitable technologies in the protection of complex IoT and Industrial IoT (IIoT) systems. Given this augmented potential of ML, there has been augmented enthusiasm in the creation of dependable intrusion detection strategies. This paper will make a just comparison of seven state of the art machine learning algorithms and their application in terms of detecting such an important security problem. The key contributions are the following: • To use a strong data preprocessing pipeline on the Edge-IIoTset data, critically solve the problem of severe class imbalance with the help of Synthetic Minority Over-sampling Technique (SMOTE). One task is to train, optimize, and evaluate a broad variety of models, such as a baseline logistic regression, tree based (Random Forest, XGBoost, CatBoost, LightGBM) and deep learning architectures (ANN, CNN) on identical conditions. • To perform a thorough analysis of feature importance, identification and explanation of the key network traffic measures used to make a model choice. The aim of this process is to enhance interpretation of the models and provide viable intelligence about security operation. • To establish an objective baseline of performance that demonstrates the high-performance level and functional efficiency of gradient boosting structures in the sphere of IoT/IIoT security. This will involve a discussion on applicability of these models in the real world in order to be implemented in edge computing environments where processing power and memory are usually constrained. The rest of this paper is organised on the following basis: Section 2 will be the review of the related research on ML-based IDS on IoT/IIoT and existing gaps in the research. Section 3 provides the details of the materials and methods: the dataset, the preprocessing processes, the description of the model, and the setting of the experiment. The results of the empirical evaluation are presented and discussed in section 4. And lastly, the paper ends with a conclusion in Section 5 that summarises the major findings, provides practical implications and recommendations as well as future research directions.

2. LITERATURE REVIEW

Elaborating on the underlying security issues as described above, this section offers a methodical review of the academic sources dealing with machine learning-based Intrusion Detection Systems (IDS) adapted to IoT and IIoT contexts. The discussion is structured in such a way that it outlines the operational peculiarities of the IoT and IIoT networks. It then examines the structural frailty of these systems as such before embarking on a critical evaluation of the modern ML-driven security practises. The review concludes with the formulation of certain research gaps that the current study attempts to address. In this part, the review of the academic and commercial studies related to security issues in IoT and IIoT systems is performed with a specific focus on machine learning-driven intrusion detection systems and security systems. The creation of modern theoretical and practical innovations creates a basis of the further research. The review can be used to provide the context of the research aims and methodological focus of the current study and thus justify the need of adaptive and intelligent security solutions tailored to the realities of operation of modern IoT and IIoT ecosystems.

2.1. Distinguishing IoT and IIoT Operational Domains

Although they are sometimes used interchangeably, the Internet of Things (IoT) and the Industrial Internet of Things (IIoT) are two distinct spheres of operation with significantly different security concerns. The consumer-focused applications related to the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

IoT environment, including connected home appliances and wearable gadgets, are mostly characterized by usability and convenience as the key design objectives. On the other hand, IIoT systems are the foundations of the industrial processes, which include industrial control systems, automated production, and the administration of vital social amenities. In such situations, the requirement of the continuity of the operation, the safety of the personnel, and the integrity of the process are absolute. The consequences of a security breach indicate this essential divergence; an event in an IIoT network like the compromise of a public water system- can initiate a real-life physical damage, environmental pollution, or threats to human life, which significantly outweigh the privacy intrusions of data or the service outage that are common with consumer IoT breaches. Figure 3 gives a structural comparison of the Industrial Internet of Things (IIoT) to that of the Internet of Things (IoT), outlining the main fields of operations, technological basis, and main applications of both to industrial and consumer environments. This contrast can be used as an explanation of the different purposes and implementation situations that define each ecosystem.

Figure 3: IIoT (Industrial Internet of Things) and IoT (Internet of Things) space.

Theoretical analogy of the Industrial IoT (IIoT) and the Internet of Things (IoT) field, focusing on the related applications in the sphere of industrial automation, monitoring, and safety in contrast to consumer electronics, home automation and wearables. Although both the Internet of Things (IoT) and the Industrial Internet of Things (IIoT) involve the creation of interconnected environments, the areas of activity and priorities the two rely on vary greatly. The IIoT sphere, shown to the left side of the drawing, is mainly focused on sensors of industrial grade, automation controls and industrial process monitoring systems, quality assurance systems, and worker protection mechanisms in general. By comparison, the IoT field (depicted on the right) comprises consumer-focused IoT devices including home appliances, personal electronics, wearables, and home security systems. This illustration highlights the way in which the two paradigms promote intelligent ecosystems according to their respective contexts factory floors and utility networks to IIoT, and home contexts to IoT. There is a necessity to deal with the security weaknesses associated with these large networks and this requires advanced protection systems. Machine Learning (ML) and Artificial Intelligence (AI) have become essential instruments in the process of identifying, predicting, and responding to cyber-incursions of IoT and IIoT infrastructure. Using the huge data streams produced by the connected equipment, AI-based solutions can discover abnormal network activity, classify certain attack vectors, and even predict security violations before they manifest themselves into the full-fledged incidents. This is supported by the creation of dedicated datasets, e.g., the Edge-IIoTset which contains both normal and malicious network traffic examples that are useful in training and optimising detection models. However, implementation of these sophisticated analytics in the IoT/IIoT security faces significant challenges. The sheer amount and diversity of the data produced by the networks of devices pose major challenges in the sphere of data curation, data storage, and computational processing. To transform this raw data into usable intelligence on security, one will need exceptionally efficient algorithms and models that are computationally light. Moreover, the dynamic environment of cyber threats makes sure that the monolithic defensive policies become outdated in a very short time. As such the ML-based security solutions should be dynamic by nature and should be able to adapt to new attack patterns and widen the network perimeters. Lastly, it is essential to ensure model interpretability; the industry practitioners need clear and reliable systems to prove the need to use automated countermeasures (Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., and Janicke, H. (2022); Gueriani, A., Kheddar, H., and Mazari, A. C. (2024)).

2.2. IoT Systems Architectural Foundation

The architectural models transforming to provide the base of the IoT and IIoT systems are organised in different layers of interconnection that pose different security threats. The understanding of this multi-tier structure that is depicted in Figure 3 is a

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

precondition of determining possible points of intrusion and comprehending attack techniques. A standard architecture of such systems normally consists of four basic levels: • Perception Layer: The physical layer which is made up of sensors and actuators which acquires information about the environment. It is susceptible to physical attacks and nodejacking. • Network Layer: It is in charge of the transmission of data based on the protocols, such as Wi-Fi, Zigbee, and the Modbus TCP. This layer can be attacked by eavesdropping, spoofing, and DDoS attacks. • Processing Layer Processes data storage, aggregation, and analysis (usually in the cloud or at the edge). It is one of the best targets of data intrusions and tampering. Application Layer: Provides services to the end-user (e.g., smart home control). The threats in this case may result in actual loss of service or privacy. Processing protocol-level traffic data would provide the Network and Processing Layers security in identifying the presence of hostile behaviour that could be either of Perception layer device origination or directed toward the Application layer. The hierarchical structure that tries to depict the IoT and IIoT would be Figure 4 which demonstrates the way data flows through four major layers -i.e. (i) Precipitation Layer, (ii) Network Layer, (iii) Processing Layer and (iv) Application Layer. The architecture puts emphasis on the contribution of every layer towards the realisation of easy data collection, transmission, processing and exploitation in smart systems.

Figure 4: IoT layered architecture

The system has 4-layer model of IoT applications systems through: Perception (sensing) the Network (transmission of data), It is architecture Processing (Storage and processing of data) Application (applications and services to users). There are various layers that have various security challenges that must be resolved in order to protect them all. The most significant channel to provide user data captured by the IoT devices is the Network Layer. Such connectivity is enabled by a wide range of communication procedures, such as Wi-Fi, Bluetooth, and other wireless standards which are specific. These technologies can be used to to central infrastructures. Then, the Processing Layer is entered after the data is transmitted. facilitate a smooth flow of information on both localised networks of devices and cloud-based infrastructure. This layer enables the subsequent processing, storage and scale sensing of huge volumes of sensor data and device outputs by being able to deliver sensor data and device outputs reliably This layer sorts all the incoming information and stores and consolidates it in a manner that can allow execution of queries on-demand. To handle such huge databases, cloud computing, edge devices and the use of 'adaptive data analysis' tools are used on the raw data to transform the data into actionable data. This information can be used to make automatic decision making or other significant action. On the top of the hierarchy we get the Application Layer where the real life on ground services that can be developed are Health-care monitoring system, Home automation system and Industrial Automation System. The level is an example of how the IoT and IIoT data enrich a broad spectrum of operations and services: the hierarchical/nested one that is so prevalent in the modern world.

2.3. Related Work in Machine Learning for IoT/IIoT IDS

More recently, scholars have been keen in adapting IDS to the limited scale of the IoT. An example is the proposal of Hasan et al. (2025) to design an autoencoder comprising light weight feature learning to IIoT network with resources optimization in mind. On the same note, the resilience of ensemble classifiers was demonstrated by Priya et al. (2021) when detecting attacks on IIoT networks. These data point to the shift of modes of high accuracy at low cost of computation, which is needed to deploy edges. The last several years have been marked by the increase in the number of studies that have been done regarding the ML- based IDSs to the IoT/IIoT. In order to curb the usual threats of IoT/IIoT architectures, a lot of work has been undertaken in coming up with ML-based IDS. We outline first recent work which has been circulated by technical approach: Lightweight and Efficient Models: Lightweight and efficient models are one of the major spheres that have undergone some consideration as part of research

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

[37]. A light modish autoencoder of the unsupervised feature learning of the IIoT network was proposed by Hasan et al. (2025), and it was able to accurately identify an anomaly with low computing cost. The same genetic algorithm-optimised stack ensemble was used by Asif (2025) to create a high intrusion detection in the heterogeneous network setups through OSEN-IoT.

• Advanced and Hybrid Convolutional-Learning Models: Scholars have engaged in the study of complex architectural structures to learn complex spatiotemporal structures. Gueriani et al. (2024) developed an attention-based hybrid LSTM-CNN model, which can adapt to industrial internet of things (IIoT) data sequence and detect the network attack is improved because it uses the temporal/spatial dependence to classify the data. Zhukabayeva et al. (2025) integrated ML, as well as neural networks and edge computing to multiclassify different data sets.

• Ensemble and Gradient Boosting Methods: Ensemble became known to be accurate. Priya et al. (2021) demonstrated the usefulness of an ensemble classifier in the detection of APH on IIoT. Gradient boosting has been proved effective in the article by Surbhi et al. (2025) that has optimised the XGBoost hyperparameters through the dragon-fly algorithm to enhance detection in healthcare IoHT. Alharthi (2025) also created a weighted voting collective to the aerial vehicle and satellite-based systems.

• Interpretability and Robustness: In the context of sophisticated deep learning models, the interpretability of the ML model is crucial, particularly in high value industrial fields. This was what Nandanwar and Katarya (2025) focused on when they created an interpretable deep learning model over the intrusion detection systems in cyber-physical systems to enhance transparency and confidence.

2.4. Introduction of Ensemble Learning

when it comes to increased Detection Accuracy. Ensemble learning algorithms, and gradient boosting frameworks (XGBoost or LightGBM) have been winning ML competitions in the recent years besides finding use as practical ML system due to their ability to make prediction with efficiency (Bentéjac et al., 2021). Alharthi (2025) applied an adaptive weighted voting ensemble model to identify the threats in satellite systems in cybersecurity. The concept behind these models is to use a group of weak learners to form a strong and generalised predictor and is very effective against the various attack vectors present in IoT traffic.

2.5. Deep Learning in Network Traffic Analysis

Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have achieved impressive performance when it comes to processing spatial and sequential data, with Deep Learning (DL) in general. Gueriani et al. (2024) applied attention-based LSTM-CNN hybrid model in IIoT to detect adaptive cyber-attacks using network stream temporal relationships. They may, however, not perform effectively on tabular data such as network packet captures that is not dynamic, since they do not have the inductive biases of tabular data structure (Shyam et al., 2020).

2.6. Recognised Research Gaps and the Originality of this Study

Despite the high advancement in the available literature, there are some gaps in the research that are still at the critical stage. The first drawback is the lack of comprehensive comparative studies that can scale to a large body of machine learning methods based on basic linear models, through to advanced gradient boosting and more complex deep learning architectures, through to reduced-dimensionality IoT-specific data sets in a uniform test platform. Most existing studies often focus on a small number of algorithms, and this hinders the development of conclusive findings about the relative performance of the algorithm. Moreover, whereas gradient boosting models, namely, XGBoost, CatBoost and LightGBM, have proven themselves to have an exemplary performance in other fields of machine learning, their use in the context of IoT security is relatively under-researched. Their ability especially in terms of computational efficiency and accuracy in handling disproportional network security data compared with those of deep learning has not been sufficiently determined. The current study aims to resolve these gaps with the help of a complex empirical analysis based on the professional Edge-IIoTset dataset. Through systematic analysis of seven various algorithms and the implementation of stringent feature signification study and performance efficiency indicators, the study lays a vital execution standard and determines the most efficient model choices in the actual intrusion detection application in IoT and IIot contexts.

3. RESEARCH METHODOLOGY

This research adapts a methodological, empirical study to design and evaluate an intrusion detection framework of IoT and IIoT ecosystems. The methodology of the research as presented in Figure 5 is based on a series of pipeline steps that involve the process of data collection, extensive preprocessing, comparative model verification, and the thorough process of performance evaluation. The steps that will be followed in ensuring methodological rigour and reproducibility of the outcomes of the experiments are described below.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

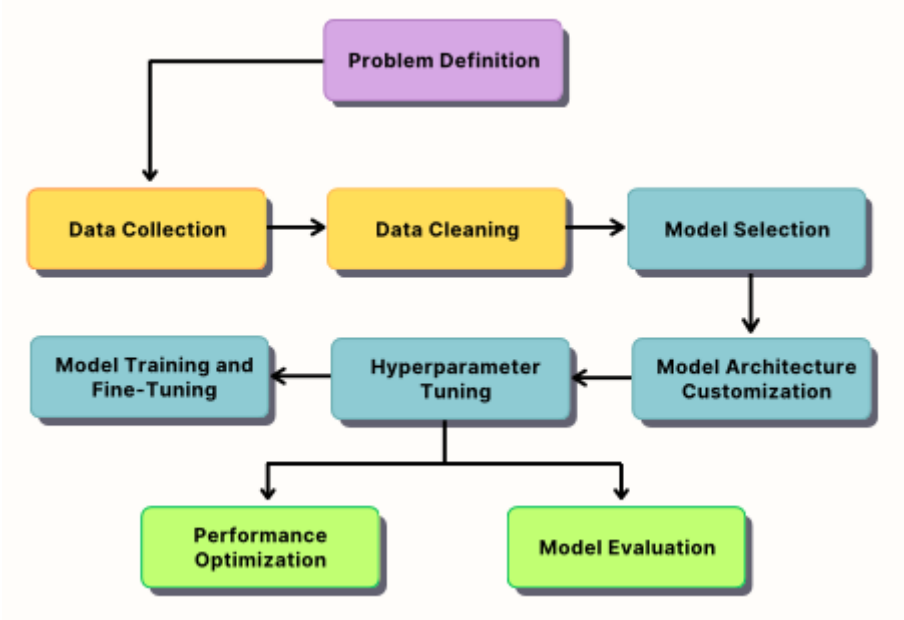

Figure 5: Research Methodology Workflow.

3.1. Description of Data

The Edge-IIoTset Cyber Security Data set. The dataset that will be used in this research is a publicly accessible cybersecurity dataset, the Edge-IIoTset (Ferrag et al., 2022), which is obtained in the Kaggle repository. The dataset was chosen as it fully simulates real IoT and IIoT network settings, with a wide variety of communication protocols, including MQTT, HTTP, Modbus TCP, ARP, and ICMP. The dataset contains the normal activity and attacks of DDoS, Scanning. etc., with 157,800 samples and 48 features. Every sample is marked with the categorical variable (Attack_type) and one binary variable (Attack_label), which makes it a supervised task. The dataset has high resolution protocol-dependent characteristics and attack patterns that the perfect benchmark to train and evaluate robust Intrusion Detection Systems (IDS) that are specific to smart systems.

3.2. Preprocessing Pipeline of Data

The preprocessing was applied in two phases with a structured framework to ensure that data is prepared and the model will be developed later. This strict process was required because the data of the IoT/IIoT network is heterogeneous and complex. The imposed pipeline was used to systematically transform the raw Edge-IIoTset information into an optimized format that can be successfully trained and evaluated in a model. It was intended as a multi-stage procedure to curb the pitfalls in data common to all the forms of analytical performance, a sequence of four sequential operations: exploratory data analysis, feature engineering, class distribution correction, and data normalisation. This methodology ensured that all the further machine learning processes were based on high-quality, relevant, and representative data.

3.2.1. Statistical Testing and Exploratory Data Analysis (EDA)

The data analysis began by a thorough Exploratory Data Analysis (EDA) to assess the data composition, quality and statistical characteristics of the dataset. The first preliminary step was the calculation of descriptive statistics of numeric variables to identify possible anomalies such as outliers and data entry errors. The main objective of this pre-test was to develop proper understanding of the data traits, therefore, predetermining the further preprocessing policies and minimising the possible risks of overfitting models or algorithmic bias. One of the key results of the analysis was that the skewness of the class distribution of the target classification, Attack_type, was statistically significant, which proved the previous studies concerning the issue of imbalance in security data (James et al., 2013). As shown in the first visualisation, the quantity of cases in the category of normal was dominant over others and the categories like those of DDoS, scanning and other attacks were over-represented. Consequently, a model that is trained on such skewed data would be biassed towards the majority class and result in low detection rates of crucial cyber-attacks that is unacceptable in a security application. To determine the predictive value of each numerical feature using continuous numeric values to differentiate between classes of attack, Analysis of Variance F-test was applied to each numerical feature versus the categorical target variable sword_HitCategories. The F-test is used to test the significance of the means of a feature between

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

classes (Normal, DDoS). The high F-statistics and p-value which was less than 2.3 retained to remain in the subsequent analysis since they indicated that it was strongly related to our target. This method builds a statistically sound foundation on the removal of non-predictive qualities which bring little value on the classification precision. The dimensionality reduction method enhances the computational performance of the feature and increases the generalizability of the model. The observations made after the analysis of the Exploratory Data Analysis, as shown in Table 1 created a comprehensive map of the exact corrective and transformative steps that were used in the later stages of the analytical process.

Table 1: Algorithm for Exploratory Data Analysis

Algorithm 3-1 of Exploratory Data Analysis (EDA)

Input: Dataset $\mathcal{D}$, significance threshold $\alpha = 0.05$
selected_features $\leftarrow$ [ ]
for each numeric feature f in $\mathcal{D}$ **do**
    Compute mean, median, standard deviation, min, max of f
    Perform one-sample t-test on f against population mean $= 0$
    if p-value $< \alpha$ **then**
        Append f to selected_features
    end if
end for
Generate bar plot of class distribution for `Attack_type`
Identify class imbalance visually and numerically
return selected_features

3.2.2. Feature Selection and Transformation

The methodology developed on the hinge of the exploratory analysis; here, the emphasis on the optimisation of the predictor set and the preparation of the data to be used by an algorithmic procedure was accomplished. The step is important in the creation of both computationally and analytically transparent models. The feature selection was directly based on statistical findings of the previous t-tests with numerical variables not showing statistical significance (p-value ≥ 0.05) being removed as non-discriminatory predictors. Also, redundant data rows such as Attack_label which had binary classification data that were already captured in multi-class Attack_type variable were eliminated to reduce the chances of multicollinearity and overfit the model. The categorical target variable, Attack type, should have been transformed into the number format to work with machine learning algorithms by turning the labels of the strings into the numbers, i.e., a normal attack should have been changed to a 1, and DDoS attack to a 2. This was achieved by encoding labels via designation of integer values to every category: Normal=0, DDoS=1, Scanning=2 and Other Attacks=3. The categorical structure was preserved by this transformation and allowed the processing of calculation. Table 1 is a graphical summary of the entire feature engineering process.

Table 2: Algorithm for Feature Selection and Transformation

Algorithm 3-2 of Feature Selection and Transformation

Input: Raw dataset $\mathcal{D}$
Remove redundant or statistically insignificant numeric columns from $\mathcal{D}$
for each row r in $\mathcal{D}$ **do**
    if r.label $=$ "Normal" **then** r.label $\leftarrow 0$
    else if r.label $=$ "DDoS" **then** r.label $\leftarrow 1$
    else if r.label $=$ "Scanning" **then** r.label $\leftarrow 2$
    else r.label $\leftarrow 3$ ▷ Other Attacks
end for
return Preprocessed dataset $\mathcal{D}$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

3.2.3. Handling Class Imbalance with SMOTE

The existence of a strong imbalance on classes as the first analysis revealed was corrected with the help of the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002). Unlike traditional oversampling, this algorithm adds fabricated data points instead of repeating the already existing ones. In case of an underrepresented class, SMOTE finds the k-nearest neighbours (a typical default of 5) of that particular example and creates new synthetic samples at the vectors in the feature space between that example and its neighbours. This methodologically broadens decision space in the minority groups and the model can build more generalised classification boundaries as opposed to memorising particular training data. An important methodological protection was also introduced, the use of SMOTE was applied to the training part only after the initial data division. Such a process helps to avoid data leakage, which is a situation when training information is damaged by the information of the test set and generates exaggerated performance measures. The intervention was effective in balancing all four categories of the attacks as the distribution represented in Figure 6 shows that it is balanced. As a result, all algorithms were exposed to the same number of examples of each type of classes in the learning process, which significantly increased their ability to recognise all types of attacks, especially the ones with low frequencies but high relevance in practise.

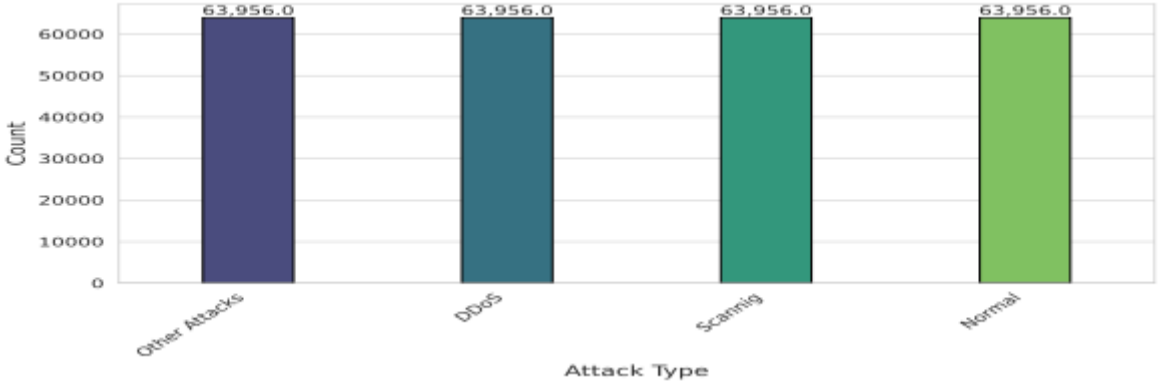

Figure 6: Class distribution of the training set after applying SMOTE.

3.2.4. Data Scaling and Normalization

Data scaling was the last preprocessing stage. The values in the dataset were of radically different scale; e.g., TCP sequence numbers (tcp.seq) can be large integer numbers, whereas protocol flags can be binary. Gradient based optimization machine learning algorithms (e.g., Logistic regression, Neural networks) or distance-based algorithms (e.g., SVMs, k-NN) are sensitive to feature size. Features of larger scale may over-represent the objective of the model, resulting in biassed learning. To reduce this, Min-Max Normalisation was used to normalise all the numerical features to a standard range [0, 1]. The transformation was done by the following formula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (3\text{-}1)$$

Where X is the original data value, and X max and X min are the largest and smallest data set values respectively. The formula standardises the value of X by dividing it within the range of values in data between 0 and 1. X is the original value, which is transformed into a normalised value, X. The reason why this scaling method was used is because the original distribution of the data is maintained but all features make equal contribution to the learning process of the model. Table 3 describes the algorithm of this step. The pipeline normalised the information, which implied it facilitated more consistent and quicker convergence in the model training process and avoided the impact of one particular feature having a disproportionate effect on the process only because of its magnitude.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

Table 3: Feature Scaling Algorithm using Min-Max Normalization.

Algorithm 3-4 of Feature Scaling using Min-Max Normalization

Input: Dataset $\mathcal{D}$

Initialize MinMaxScaler

for each numeric feature f in $\mathcal{D}$ **do**

  for each value v in f **do**

    $v_{\text{scaled}} \leftarrow \dfrac{v - \min(f)}{\max(f) - \min(f)}$

  end for

end for

Check $\mathcal{D}$ for missing values or anomalies after scaling

Store scaled dataset as `final_dataset` for model training

return `final_dataset`

To conclude, this carefully crafted four-step preprocessing pipeline, including EDA, feature engineering, imbalance correction, and normalisation played a critical role in transforming the raw and complex network data into a clean and balanced data and normalising it. This careful training was a strong basis to the successful and advantageous comparative analysis of the subsequent machine learning models.

3.3. Chosen Machine Learning models

Rationale and Set Up. It has a variety of seven machine learning models so as to give a comparative analysis.

3.3.1. Conventional Model

Logistic Regression. Logistic Regression (LR) was used as a benchmark since it is simple, computationally efficient, and highly interpretable (Hosmer et al., 2013). This linear nature is an important benchmark of more complex algorithms though. L2 regularisation was used to avoid overfitting in the model.

3.3.2. The Ensemble models based on Trees

Random Forest, XGBoost, CatBoost, LightGBM. The 4 progressive ensemble techniques were selected due to their effectiveness in managing structured and tabular data: • Random Forest (RF): It is a bagging algorithm, which builds a collection of decision trees and combines their results, eliminating variance and improving robustness (Breiman, 2001). • XGBoost (Extreme Gradient Boosting): It is a scalable gradient boosting model that has a high performance, fast and has a built-in regularisation of the model (Chen and Guestrin, 2016). • CatBoost: This is a gradient boosting algorithm that is capable of working with categorical data having minimal preprocess and eliminating prediction shift (Prokhorenkova et al., 2018). Light GBMC (Light Gradient Boosting Machine) High performance gradient boosting model that employs leaf-wise strategy of tree growth, speed, and memory efficiency (Ke et al., 2017). With all other parameters kept to default, these models were started with 100 estimators and offered a robust coverage of the state-of-the-art ensemble methods.

3.3.3. Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) Deep Learning Models.

In order to test the performance of deep learning with tabular network data, two architectures were used: Artificial Neural Network (ANN): An average feedforward network with three thicken layers, ReLU activation functions, and regularization by means of Dropout layers (rate=0.2). Multi-class classification was done in the output layer with the use of SoftMax activation. • Convolutional Neural Network (CNN): CNN is a 1D-CNN model with convolutional and max-pooling blocks, which were applied to investigate the power of the model to identify local patterns in the feature space, but is primarily applied to a spatial or sequential input (LeCun et al., 2015). The two models were optimised using the Adam optimizer and were trained with insular categorical cross-entropy loss.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

3.4. Experimental Setup

3.4.1. Data Splitting Strategy

An 80-20 stratified traintest split was used to partition the preprocessed data. This method maintained the initial classes in both subsets allowing a representative evaluation. All the preprocessing (SMOTE) steps were performed just to the training fold to avoid information leakage and make the test set remain a pure sample of real-world, imbalanced data.

3.4.2. Hyperparameter Customization and Model Training

To reproducibly test all experiments, a fixed random seed was applied (42). They used the tree-based models without hyperparameter optimization to get a baseline performance. The neural networks (ANN and CNN) were trained over 50 epochs in a batch size of 32 and 10 percent of the training was reserved to validate. In order to make probabilistic predictions reliable on models such as XGBoost, CalibratedClassifierCV was used to perform probability calibration.

3.4.3. Performance Assessment Dimensions

Area Under the Receiver Operating Characteristic Curve (AUC-ROC): The area of the curve under the ROC, when an average is made of all the classes. This metric is used to evaluate how the model differentiates the classes in all possible classification thresholds giving a sound measure of the performance which does not depend on the distribution of the classes. The evaluation strategy was chosen as a multi-faceted evaluation strategy with the following metrics, which were calculated using weighted averaging in order to consider the imbalance of classes:

• Proportion of the total correct predictions is called Accuracy. Precision: Capability of missing false positives.
• Recall (Sensitivity): Capability to find out all positive cases.
• F1-Score: Harmonic mean of the precision and the recall.
• Balanced Accuracy: Mean of recall on each of the classes, which is appropriate to use with imbalanced datasets. These measures offer a global picture of model performance, which is essential in estimating the performance of the IDS in the context of security where false alarms and missed detections have high costs (Powers, 2020).

4. DISCUSSION OF RESULTS

This part contains a detailed analysis of the seven machine learning models applied to intrusion detection on the Edge-IIoTset dataset. The analysis will look at the results of the individual models, give a comparative analysis of the results based on standard metrics, compute efficiency when using the models based on IoT deployment limitations, and explore the significance of features in model interpretability.

4.1. Individual Model Performance

4.1.1. Ensemble Learners: XGBoost, LightGBM, CatBoost and Random Forest

The tree-based ensemble models were found to be better in the classification of IoT/IIoT cyber-attacks. XGBoost was the highest performing algorithm with an accuracy of 95.07, precision of 95.46, recall of 95.07 and F1-score equal to 95.14. LightGBM was next in line with similar performance (accuracy: 95.02, F1-score: 95.10), and it demonstrates its effectiveness with large-scale data. Random Forest also showed good performance with the accuracy value of 94.24 with CatBoost performing more efficiently with the accuracy value of 93.94. The confusion matrices of such models (Figures 7 - 10) indicate the presence of an excellent class-wise discrimination, with little misclassification in attack categories. The power of the ensemble methods is that they are capable of capturing a complex interaction of features with many decision trees and are resistant to overfitting.

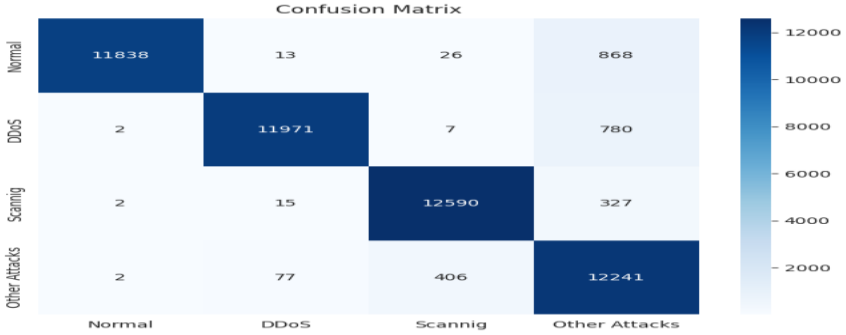

Figure 7: Confusion matrix of XGBoost

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

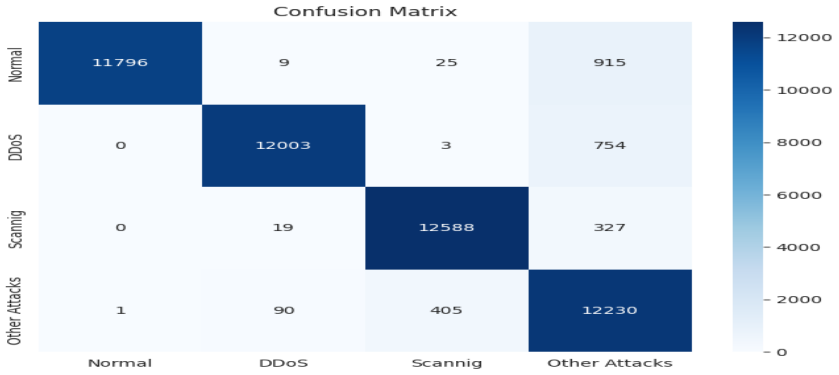

Figure 8: Confusion matrix of LightGBM

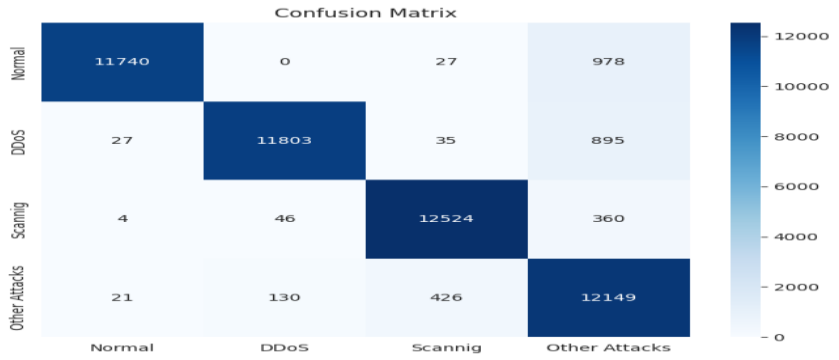

Figure 9: Confusion matrix of CatBoost

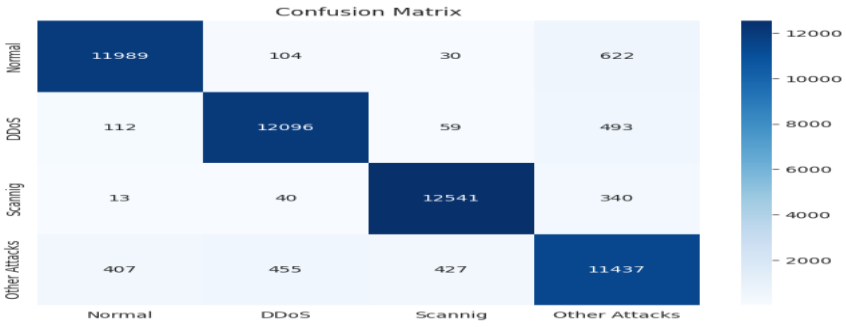

Figure 10: Confusion matrix of Random Forest

4.1.2. Deep Learning Models: ANN and CNN

The deep architecture approaches had moderate performance as compared to the ensemble approaches. The Artificial Neural Network (ANN) attained an accuracy of 81.76 percent and precision and F1-score values of 86.20 percent and 82.03 percent respectively. The Convolutional Neural Network (CNN) showed reduced performance with 75.43 percent accuracy and 76.49 percent F 1 score. Their confusion matrices (Figures 11-12) reveal that there was a lot of misclassifications between the cases of "Normal" traffic and "Other Attacks" which implies that it is difficult to learn the discriminative features of the tabular network data with the help of deep structures that are designed with the spatial or sequential patterns. Although ANN demonstrated a fairly good performance in learning non-linear relationships, both deep learning models performed poorly compared to tree-based ensembles in this particular task.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

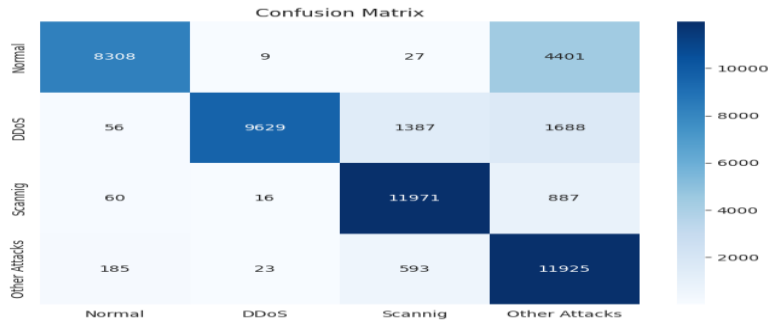

Figure 11: Confusion matrix of Artificial Neural Network

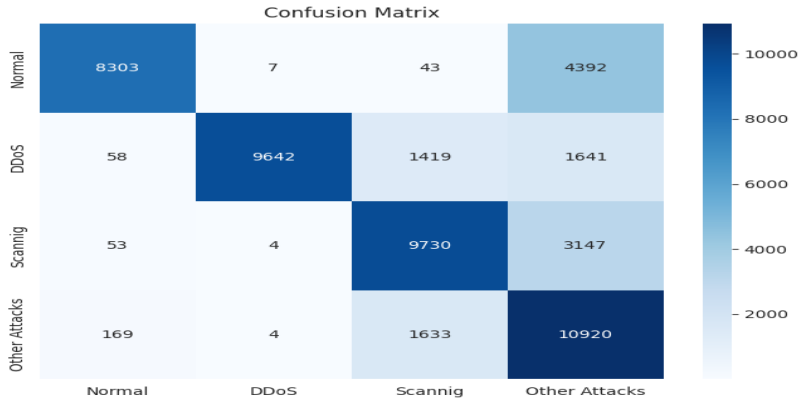

Figure 12: Confusion matrix of Convolutional Neural Network

4.1.3. Baseline Model: Logistic Regression

As predicted, the lowest performance exhibited by all models was the Logistic Regression which has an accuracy of 65.20, 70.56 precision, and 65.81 F1-score. The confusion matrix (Figure 13) shows high misclassification on all classes of the labels especially between the labels of normal and other attacks (5,400 times) and on the attack classes. The performance of this model validates the fact that linear models are insufficient to characterise the non-linear relationships of data in an IoT network traffic to perform advanced intrusion detection tasks.

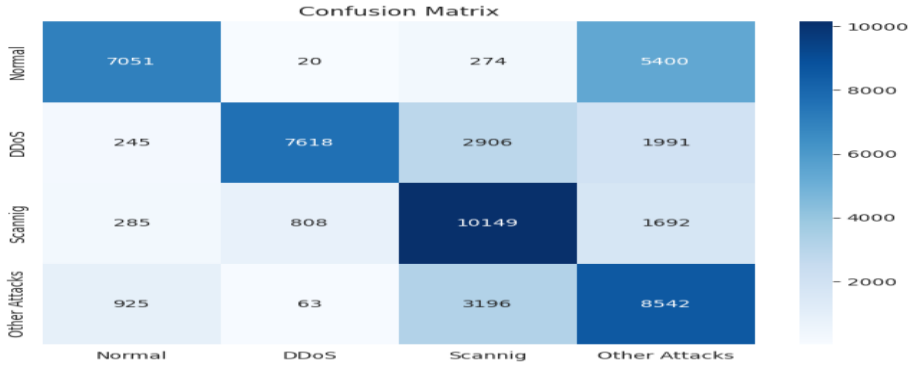

Figure 13: Confusion matrix of Logistic Regression

4.2. Comparative Analysis Based on Standard Metrics

A multi-faceted analysis was performed to make a comparison of the seven models in all aspects. The findings that are summarised in Table 1 in terms of six key performance measures indicate that there is a clear and consistent hierarchy in performance. This discussion has given an overview of the ability of each model to identify the intrusions with accuracy and control false positives and false negatives.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

Table 1. Comparative Model Performance on the Edge-IIoTset Test Set

Model	Accuracy	Balanced Accuracy	Precision	Recall	F1-Score	AUC-ROC (Macro)
XGBoost	95.07%	95.06%	95.46%	95.07%	95.14%	0.99
LightGBM	95.02%	95.01%	95.41%	95.02%	95.10%	0.99
CatBoost	94.24%	94.23%	94.65%	94.24%	94.32%	0.98
Random Forest	93.94%	93.92%	94.38%	93.94%	94.05%	0.98
ANN	81.76%	81.75%	86.20%	81.76%	82.03%	0.90
CNN	75.43%	75.41%	81.69%	75.43%	76.49%	0.85
Logistic Regression	65.20%	65.18%	70.56%	65.20%	65.81%	0.75

Note: AUC-ROC (Area Under the Receiver Operating Characteristic Curve) scores are calculated using a macro-average to account for class imbalance.

In the first rank, there are the gradient boosting models, namely the XGBoost, LightGBM, and CatBoost, which have shown the best results in all measures. XGBoost became the most successful algorithm as it scored the highest in accuracy (95.07%), precision (95.46%), and F1-score (95.14%). The almost equal values on accuracy and balanced accuracy of these models imply their ability to deal with the test set that is not well balanced is also strong. More importantly, their almost flawless macro-average AUC-ROC scores (0.98 and above) indicate an outstanding capacity to differentiate among all classes in all potential classification thresholds. This is one of the strengths of an IDS in which the price of an attack going undetected (false negative) is high. Random Forest has also provided a good performance but it was a bit less than the gradient boosting variants. The difference between the performance of the ensemble approaches and the deep learning models (ANN and CNN) is overwhelming. Though the ANN was also moderate, its measures are significantly lower, and its AUC-ROC of 0.90 indicates a lower overall separability among classes. The poor performance of the CNN emphasises further the architectural incompatibility with tabular data. The baseline Logistic Regression model verified its insufficiency to this complex task, as the lowest scores are obtained across the board, and the AUC-ROC is only 0.75, showing the ability of the model to achieve the performance almost identical to that of random guessing in certain classes. To offer a visual comparison of these findings, the values of Table 1 are plotted in the form of graphs in Figures 4.8 -4.12. The uniformity of the performance hierarchy based on all metrics can be observed through these numbers, which further support the advantage of the use of tree-based ensemble techniques in this task of intrusion detection.

4.2.1. Accuracy and Balanced Accuracy Comparison

The comparison of the accuracy of the models (Figure 14) shows the evident performance hierarchy. Ensemble methods were always in the lead with the top member being XGBoost (95.07%), LightGBM (95.02%), CatBoost (94.24%), and Random Forest (93.94). This was also true of the balanced accuracy parameters (Figure 15), which showed that these models were not affected by the initial imbalance in the data set across all classes. The middle range was occupied by the deep learning models of ANN (81.76) and CNN (75.43), with the Logistic Regression coming second (65.20). The fact that the difference between the standard and the balanced accuracy of the ensemble methods is minimal signifies that they are effective in the sense that they can be used to deal with class imbalance.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

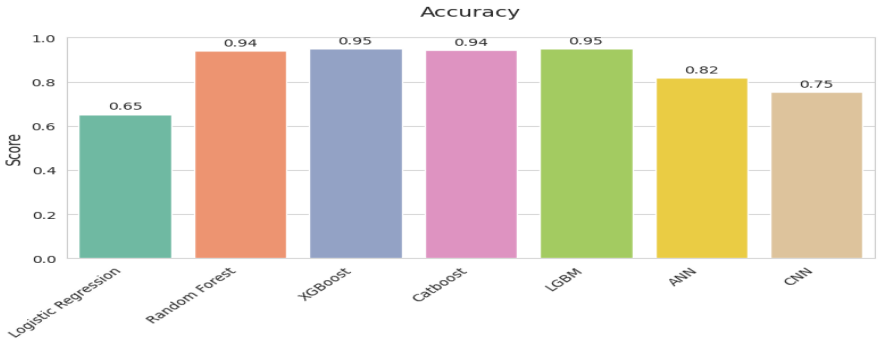

Figure 14: Accuracy Comparison

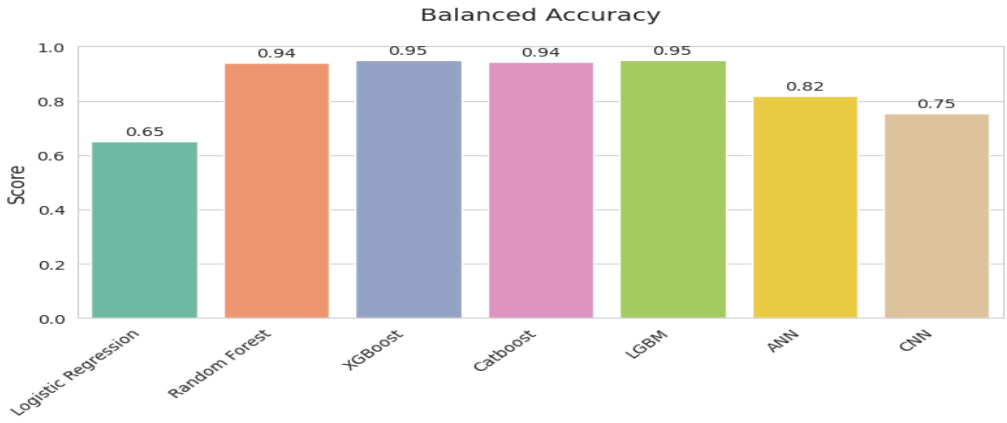

Figure 15: Balanced Accuracy Comparison

4.2.2. Precision, Recall, and F1-Score Analysis

The ensemble models represented a higher level of performance in all the measures of evaluation as shown in Table 1. The best precision score of XGBoost was 95.46% which indicates its high sensitivity to reduce false positives which is an important feature in any operational IDS to avoid alert fatigue. The small difference between the accuracy and balanced accuracy scores of the top models verify the strength of the models to resist the class imbalanced presented by the dataset. This is also confirmed by their near-optimal macro-average AUC-ROC scores (XGBoost: 0.99, LightGBM: 0.99) which indicates an exceptional ability not only to be able to classify instances at a default threshold with accuracy, but also to rank the instances with accuracy throughout the entire range of decision thresholds, effectively distinguishing all types of attacks before regular traffic.

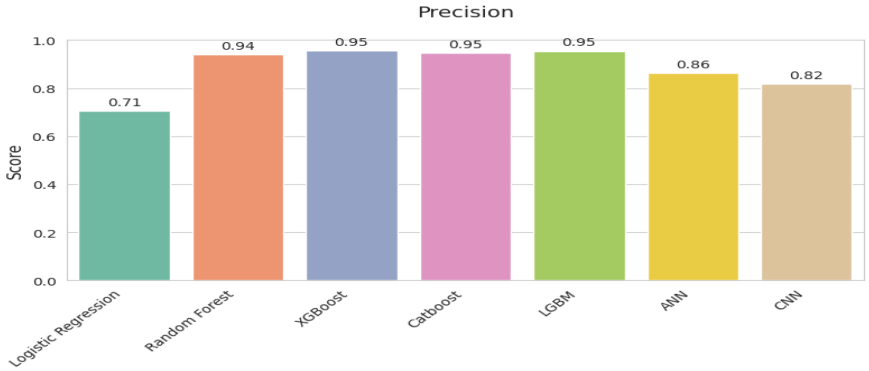

Figure 16: Precision Comparison

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

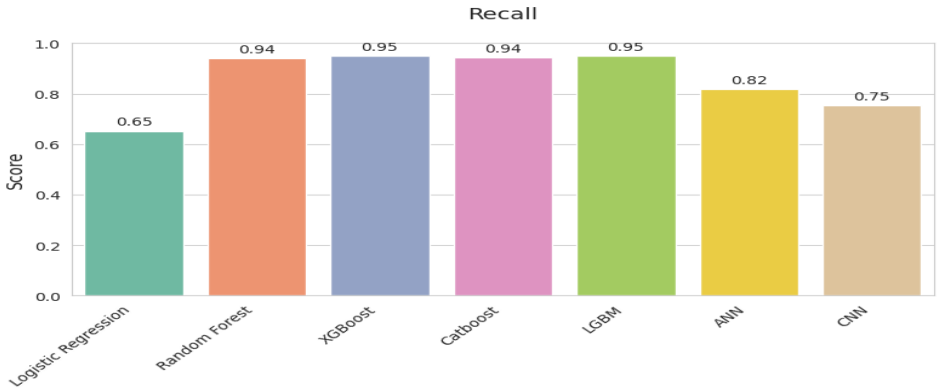

Figure 17: Recall Comparison

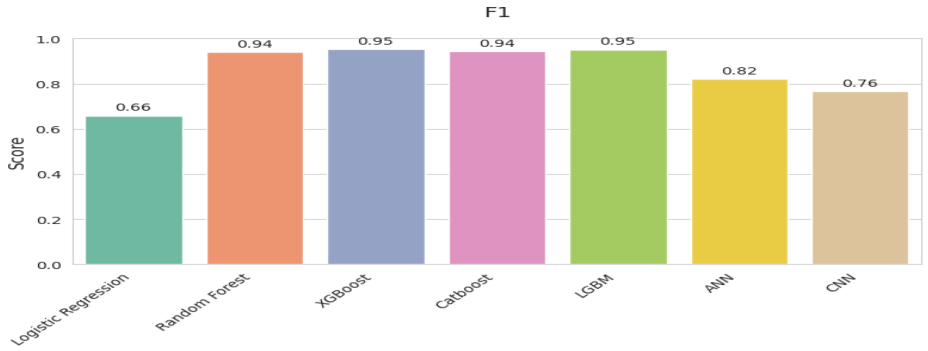

Figure 18: F1-Score Comparison

4.3. Analysis of Computational Efficiency for IoT Deployment

Table 2 critically analyses the computational efficiency, considering its metrics that are important in the real-world implementation of the IoT. Logistic Regression showed the lowest inference time (0.000656 ms/sample/sample) and lowest memory footprint (2.99 KB) which is appropriate in extremely resource-limited devices, but has insufficient detection accuracy to be used in security systems. XGBoost was the top-performing model with the most competitive balance (0.020749 ms/sample) and a reasonable amount of memory usage (1256.85 KB). LightGBM competed in terms of training time (13.18 seconds) although it was slightly less accurate than XGBoost. Random Forest was precise, but consumed too much memory (218174.92 KB), which made it too practical to use on an average IoT instrument. The deep learning models, specifically CNN, demonstrated unrealistically long training times (3465.33 seconds) and inference times, which do not allow them to be used in edge deployment real-time.

Table 2. Computational Efficiency Analysis for IoT Deployment

Model	Inference Time (ms/sample)	Memory Footprint (KB)	Training Time (s)
Logistic Regression	0.000656	2.99	16.27
XGBoost	0.020749	1256.85	18.51
LightGBM	0.064983	1725.07	13.18
CatBoost	0.009942	3101.96	199.12
Random Forest	0.024320	218174.92	34.90

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

Model	Inference Time (ms/sample)	Memory Footprint (KB)	Training Time (s)
ANN	0.052988	85.50	863.33
CNN	0.093879	518.43	3465.33

4.4. Feature Importance Analysis for Model Interpretability

The analysis of feature importance (Figures 19 - 22) showed the similarity in patterns among the models of the ensembles, discovering the important indicators of malicious activity. The most important predictors were low-level TCP features (tcp.flags.ack, tcp.seq, tcp.ack), which occurred in all XGBoost, LightGBM, and CatBoost models. Attributes of application-layer protocols (mqtt.hdrflags, http.content length, arp.opcode) were also in the top list, which underscores the importance of multi-protocol analysis of IoT security. Their ability to be consistent with key characteristics by various models justifies why they are important in distinguishing patterns of attack and normal network behaviour. This interpretability gives actionable information to security practitioners, which indicates which parameters of networks are supposed to be emphasised in the monitoring systems.

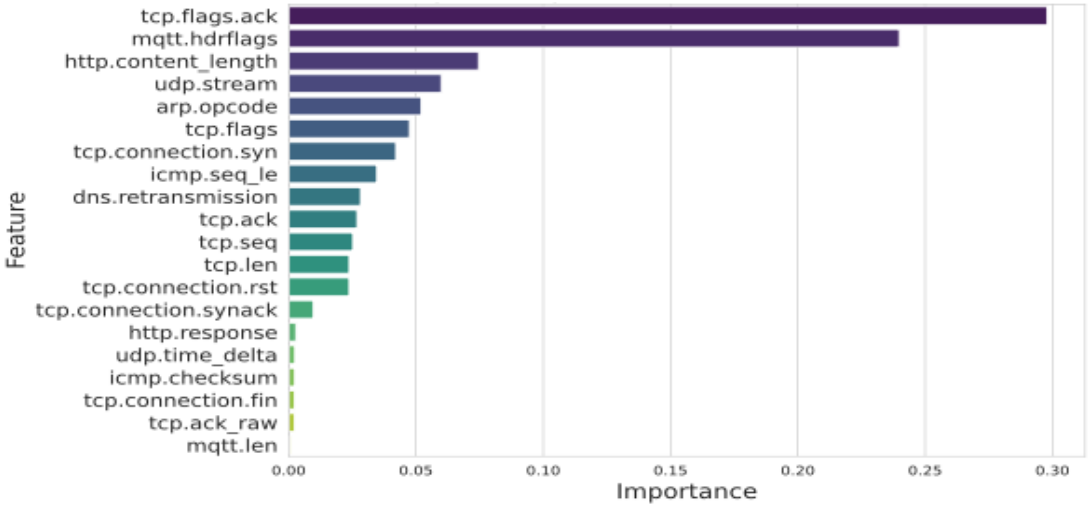

Figure 19: Feature importance chart of XGBoost

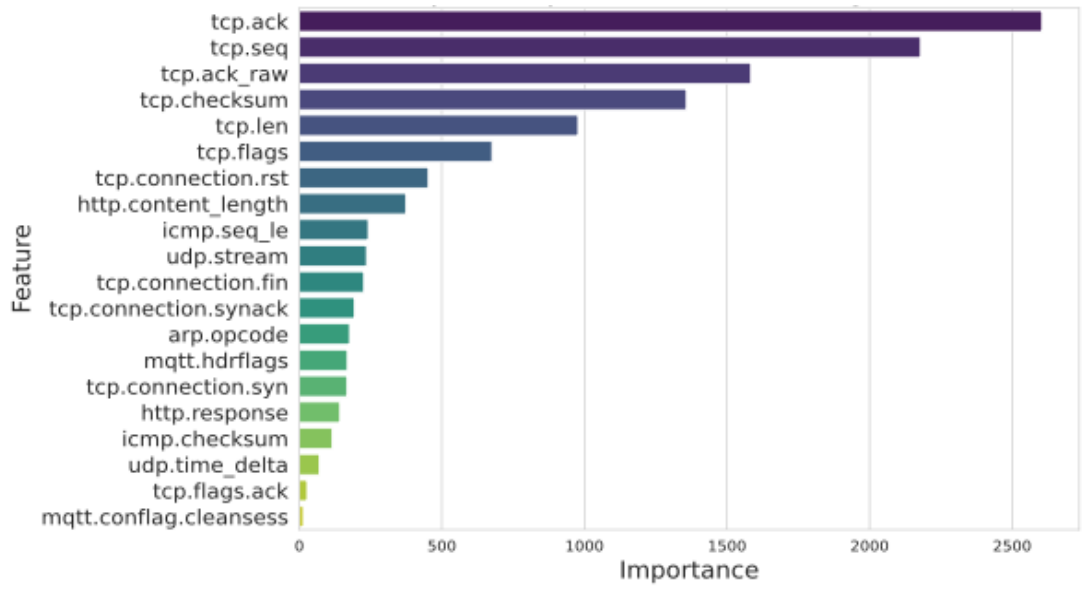

Figure 20: Feature importance chart of LightGBM

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

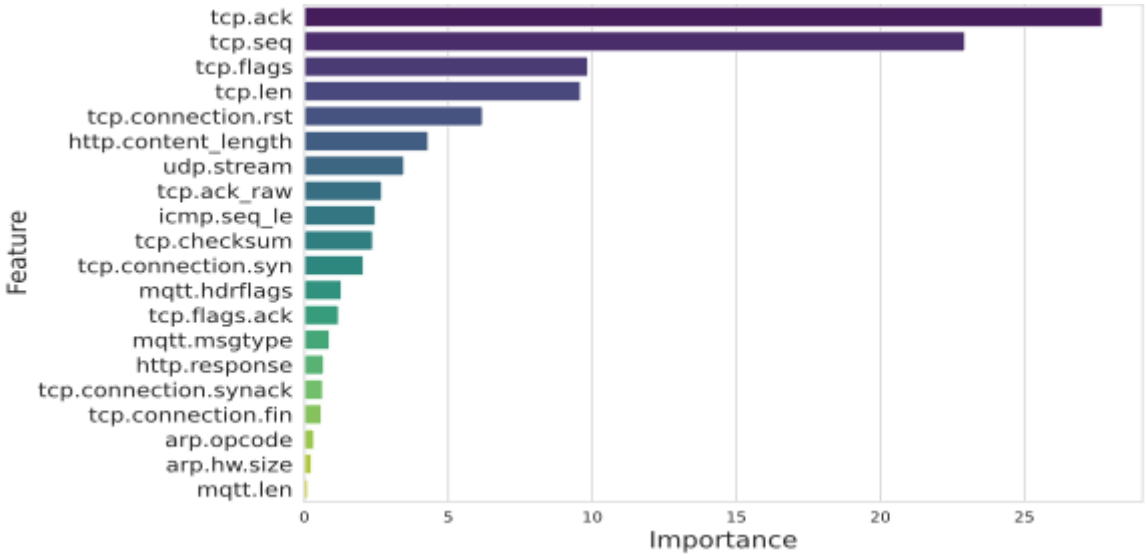

Figure 21: Feature importance chart of CatBoost

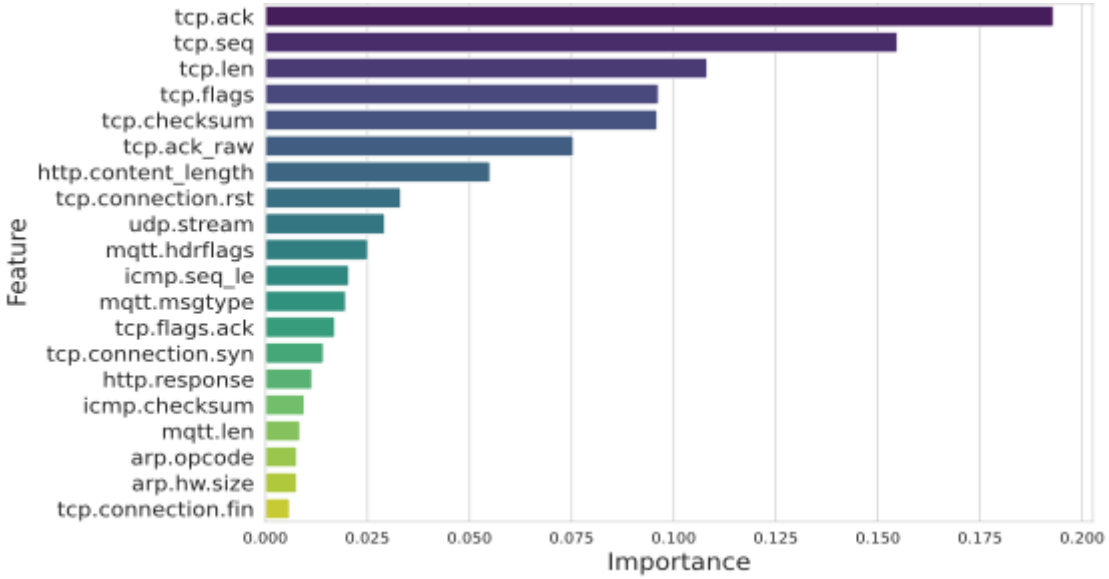

Figure 22: Feature importance chart of Random Forest

All the results show that the gradient boosting algorithms, especially XGBoost, give the most suitable balance of detection accuracy, computational efficiency, and interpretability to the IoT/IIoT intrusion detection system. Their high effectiveness in all the assessment areas makes them the most appropriate solution that can be used to ensure that resource-limited edge environments are secured against changing cyber threats.

4.6. Gradient Boosting Models: Superiority of Gradient Boosting

The general effectiveness of gradient boosting models (XGBoost, LightGBM, CatBoost) as compared to other algorithms can be explained by the fact that they have architectural benefits to tabular data analysis. Gradient boosting is superior compared to deep learning models which need huge amounts of data to discover an interpretation of features (learned in a pure model-building step), whereas structured data is exploited (learned in an additive model-building step) (Chen & Guestrin, 2016). The trees in the ensemble are used in succession and each tree aims to mitigate the mistakes of the earlier trees and this way the model is able to learn and capture complex, non-linear relationships between features of the network traffic and attack signatures. The dimensionality of Edge-IIoTset data (48 features of various protocols, including MQTT, HTTP, Modbus TCP, etc.) suggests that the landscape of feature interactivity is better modelled by gradient boosting than by other algorithms. Regularisation mechanisms

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

(L1 and L2 in XGBoost) inherent to the algorithm are a highly important tool that helps prevent overfitting, which at the same time is crucial in the context of addressing the problem of noisy, heterogeneous data that real-world IoT context entails. Moreover, the natural capabilities of these models to process missing data values and outliers are ideal with the real-life issues of network traffic information gathering. Such a combination of algorithmic properties was the reason why XGBoost has the best results (95.07% accuracy, 95.14% F1-score), which makes gradient boosting the most successful paradigm in the context of IoT/IIoT intrusion detection on the tabular network data.

4.7 When Deep Learning Models Failed on Tabular Network Data

The poor performance of the deep learning design the Artificial Neural Network (ANN) and specifically the Convolutional Neural Network (CNN) may be largely explained by the architectural mismatch with the characteristics of tabular network data. Deep learning is particularly good at problems with strong local correlations and hierarchies in feature representations i.e. spatial coherence in images (processed by CNNs) or temporal dependencies in sequences (processed by RNNs). Table data, such as the network traffic characteristics in the present study, however, do not have this implied locality; it is not more or more significantly related to a TCP sequence number than it is to an HTTP content-length field. The CNN convolutional philtres which are set to identify the local patterns therefore failed to identify meaningful spatial hierarchies, which is why the CNN is less effective than tree-based models, which are also tailored to make optimal splits across all features. But the tabular network data does not have this locality the correlation between the 15th feature (e.g., http.content_length) and the 16th feature (e.g., tcp.flags.ack) is not necessarily higher than their correlation with the 40th feature. The CNNs convolutional layers that are constructed to identify local patterns with the help of sliding philtres could not find any significant spatial hierarchies within the feature vector. Although the ANN proved superior to the gradient boosting owing to its fully connected structure to incorporate interactions across the globe, it nonetheless performed poorly as compared to the gradient boosting. This performance disparity is arguably attributable to a number of factors: (i) deep networks are generally more sensitive to hyperparameter optimization; (ii) deep networks are generally more sensitive to large datasets in order to make inferences; and (iii) tree-based models are generally more robust to feature scaling and missing values. These results indicate that when using tabular data of IoT security, the investment in the complex deep learning architecture might not pay off in the same proportion to the highly-tuned ensemble algorithms.

4.8. Elucidative Features on IoT/IIoT Intrusion Detection

Feature Importance Insights. The patterns of feature importance analysis of all the high-performing models presented information that offers practical intelligence to cybersecurity practitioners. The prevalence of TCP-level features (tcp.flags.ack, tcp.seq, tcp.ack) among XGBoost, LightGBM and CatBoost suggests that the interactions between basic transport-layer communication patterns are extremely discriminative in detecting attacks. This is in line with documented attack strategies such as DDoS attacks that tend to use TCP flags and sequence numbers to saturate resources and scanning that produce atypical SYN packet sequences and ACK packet sequences. The high input of application-layer features, especially mqtt.hdrflags and htp.content_length was also important. This highlights the fact that the IoT/IIoT security cannot be based only on the network-related metrics but must include the protocol-related semantics. The MQTT protocol, which is common in IoT messaging, seems especially susceptible to header manipulation attacks, and abnormal HTTP content lengths could be evidence of data exfiltration. This similarity in the ranking of feature importance in several models gives credence to the validity of the results and indicates that security monitoring system must focus on these indicators to detect the presence of threats in real-time.

4.9 Striking the Right Balance between Accuracy and Efficiency: Implications to a Real-World Deployment.

The trade-offs analysis (Section 4.3) indicates that the computational efficiency of this architecture was a critical trade-off to take into account in order to be used in practise in resource-constrained environments of the IoT. Although XGBoost has the best accuracy, LightGBM provided a strong competitor, having the same performance (95.02% accuracy) and shorter training time (13.18 seconds vs. 18.51 seconds). LightGBM is especially appropriate to situations where it is necessary to recalculate the model or use it on the edge with limited computing power. Random Forest has a very high memory demand (218 MB), thus making it unfeasible in most edge deployments, even though its accuracy (93.94%) is great. Equally, the deep learning models (ANN: 863s, CNN: 3465s) by virtue of their long training time are not suitable to the dynamic IoT world where quick adaptation to new threat is necessary. These results indicate that gradient boosting solutions with reduced resource use (especially XGBoost and LightGBM) are the optimal tradeoff between detection and operation in real-world privacy-protected IoT security applications.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

4.10 Limitations and Generalizability

Although the presented research offers a very strong argument supporting the dominance of the gradient boosting approach in IoT IDS, one should admit multiple limitations. To begin with, the results are inherent to the properties of the Edge-IIoTset data set. The performance can be different when used on datasets with different network protocols, more recent attack vectors or other data distributions. Secondly, the study was carried out in a non-controlling offline environment. It fails to tackle the major issues of real time continuous learning and inference in a dynamic network environment where concept drift is always an issue. Lastly, although computational efficiency has been considered in theory, the models have not been implemented on a real resource-constrained platform such as microcontrollers. The actual performance and energy usage of such extreme edge conditions is an open issue and is a vital field to subject to validation in future.

4.11 Resolving Class Imbalance: The Effect of SMOTE on the Model Performance

The SMOTE strategic use in the preprocessing stage was also essential in attaining strong performance on all the types of attacks. The original data was highly imbalanced with the normal traffic overwhelming the attack types a realistic situation in operational networks where malicious behaviour is rare in nature. Lack of balancing would have skewed the models to predict the majority class, which would have low detection rates of critical attacks. The good results of SMOTE are demonstrated by the high values of balanced accuracy of the best models (XGBoost: 95.06, LightGBM: 95.01) which were not much different in comparison with their regular accuracy scores. It means that the models were trained to be able to identify all types of attacks in a more or less equal amount, not only common ones. This balanced performance is also supported by the confusion matrices, which indicate that there are high detection performances of all four classes. This effective management of class imbalance proves that SMOTE is a vital part of the preprocessing pipeline when the task of applications in the sphere of cybersecurity is to detect very rare but critical events. Overall, the findings are very thorough in covering the objectives of the research, since they define the gradient boosting as the most efficient method to use in IoT/IIoT intrusion detection, the understanding of critical detection properties, and the practical recommendations of the tradeoff between accuracy and computational capacity in a real-world deployment.

5 CONCLUSION

The research conducted a systematic study to determine the effectiveness of different machine learning tools and deep learning tools in optimising intrusion detection in IoT and IIoT network.

The experimental outcomes prove the existence of considerable hierarchy of the model performance. XGBoost and its variants came out as the best models, with maximum accuracy (95.07), precision (95.46), as well as F1-score (95.14). LightGBM and CatBoost gave similar and high-performance results, and also the results of the Random Forest were strong, albeit, less efficient. Deep learning models (ANN and CNN), in turn, demonstrated average performance, which is indicative of a lack of congruence between the tabular data of the network. The base-line Logistic Regression model performed very poorly, which proves the insufficiency of the linear models in this complicated activity. In addition to raw accuracy, computational efficiency analysis showed that XGBoost and LightGBM were the best in providing a performance/resource ratio that is not too high and too low respectively, the latter being a vital factor in the deployment of IoT. More so, the feature importance analysis showed repeatedly that TCP-level features (e.g., tcp.flags.ack, tcp.seq) and application-layer indicators (e.g., mqtt.hdrflags, http.content_length) are the most significant ones as they indicate malicious activity.

This study fulfilled its main objective of designing and testing a data-based model of detecting intrusions in IoT/IIoT. The particular objectives were achieved in the following manner: • Strong Data Preprocessing: A full pipeline was followed, with the application of SMOTE used to successfully balance the classes, and Min-Max scaling used to have a stable model training.

• *Evaluation of Algorithms:* Seven models in total were stringently compared and it was found that gradient boosting ensembles outperformed deep learning and conventional algorithms clearly. • Feature Importance Analysis: The research discovered and explained the significance of important features, which also brought transparency and practical results in terms of prioritising security monitoring endeavours.

• *Comparison of Model Performance*: Multi-metric comparison of models proved that ensemble methods, in particular, XGBoost, are the best to apply in the field of cybersecurity in a real environment. To conclude in regards to the topic of intrusion detection systems based on the tabular network traffic data of the IoT/IIoT setting, gradient boosting models are the existing state-of-the-art which unites high precision, efficiency of operations, and interpretability.

6 IMPLICATIONS OF THE RESEARCH

The results have far-reaching implications on an academic research and industrial practise.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

Theoretical Implication: The study makes a contribution to the machine learning community by empirically supporting the fact that tree-based ensembles are better approaches to structured, tabular data in the cybersecurity domain than deep learning architectures. It proves the fact that more complicated architectures such as CNNs do not necessarily have an edge and the significance of data structure to model architecture matching.

Practical Implications: This study is a proven blueprint to cybersecurity practitioners and architects of IoT systems. Innovations in edge detectors based on models such as XGBoost or LightGBM can be used to enable organisations to install efficient and high-accuracy intrusion detection mechanisms. The critical features identified (e.g., TCP flags, MQTT headers) provide a direct behavioural guideline on how the network monitoring and security protocol designs can be optimised to allow a more focused and more resourced security posture.

6.1. Limitations of the Study

Although extensive, this study has a number of limitations to be viewed. To begin with, the results are conditioned by the properties of the Edge-IIoTset dataset; the results might be different in case of other sets with different protocols, attack vectors, or data distributions. Second, the experiment was analytical and carried out in an offline setting and not in a dynamic network environment. Third, although the discussion was made on computational efficiency, the experiments were not conducted on actual implementation on extremely resource-constrained IoT hardware (e.g. microcontrollers), where memory and power constraints may further limit what models can be chosen.

6.2. Future Research Instructions

In accordance with the conclusions and limitations, the following are some of the potential directions of future work.

6.2.1. Adding Temporal Analysis to RNNs/LSTMs.

Network intrusions can be in form of sequences of events in time. Future research ought to investigate Recurrent Neural Networks (RNNs), specifically, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, to capture temporal variation of network traffic flows in a network. This would do a great job in detecting slow-burning attacks and multi-stage intrusion that are hard to detect on single packet snapshots.

6.2.2. Creating Hybrid and Lightweight Edge Deployment Model.

It has a lot of potential in the creation of hybrid models that would build the merits of various architectures. An example of this is the case of an autoencoder, where unsupervised learning of features on raw traffic data is done, and then the result is input to a lightweight gradient boosting model. Moreover, investigations into the method of compressing models (including pruning, quantization, and knowledge distillation) to produce ultra-lightweight models of the highest-performing models, including XGBoost, are needed to be deployed on the smallest edge devices.

FUNDING

There is no funding provided during and after this research work.

COMPETING INTERESTS

Authors declared no competing interests exist during and after this research work.

REFERENCES

[1] AlDosari, F. (2017). Security and privacy challenges in cyber-physical systems. *Journal of Information Security, 8*(4), 285–295. https://doi.org/10.4236/jis.2017.84019

[2] Alharthi, R. (2025). Weighted Voting Ensemble Model Integrated with IoT for Detecting Security Threats in Satellite Systems and Aerial Vehicles. *Journal of Computer and Communications, 13*(2), 250–281. https://doi.org/10.4236/jcc.2025.132015

[3] Asif, S. (2025). OSEN-IoT: An optimized stack ensemble network with genetic algorithm for robust intrusion detection in heterogeneous IoT networks. *Expert Systems with Applications, 276*, 127183. https://doi.org/10.1016/j.eswa.2025.127183

[4] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review, 54*, 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

[5] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

[6] Buja, A. (2023). A Model Proposal of Cybersecurity for the IIoT: Enhancing IIoT Cybersecurity through Machine Learning and Deep Learning Techniques. *Advances in Artificial Intelligence and Machine Learning, 4*(3), 140. https://aimlscicom.org/journals/index.php/AIMLSCICOM/article/view/40

[7] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

[8] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

[9] El-Sofany, H., et al. (2024). Using machine learning algorithms to enhance IoT system security. *Scientific Reports, 14*(1), 12077. https://doi.org/10.1038/s41598-024-62684-x

[10] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access, 10*, 40281-40306. https://doi.org/10.1109/ACCESS.2022.3165809

[11] Gueriani, A., Kheddar, H., & Mazari, A. C. (2024). Adaptive Cyber-Attack Detection in IIoT Using Attention-Based LSTM-CNN Models. In *2024 International Conference on Telecommunications and Intelligent Systems (ICTIS)*. IEEE. https://doi.org/10.1109/ICTIS61804.2024.10541120

[12] Hasan, T., Hossain, A., Ansari, M. Q., & Syed, T. H. (2025). Enhanced Intrusion Detection in IIoT Networks: A Lightweight Approach with Autoencoder-Based Feature Learning. *arXiv preprint arXiv:2501.15266*. https://arxiv.org/abs/2501.15266

[13] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.

[14] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*.

[15] Kikissagbe, B. R., & Adda, M. (2024). Machine learning-based intrusion detection methods in IoT systems: A comprehensive review. *Electronics, 13*(18), 3601. https://doi.org/10.3390/electronics13183601

[16] Kumudham, R., & Shakir, M. (2024). Enhancing brix value prediction in strawberries using machine learning: A fusion of physiochemical and color-based features for improved sweetness assessment. *Malaysian Journal of Computer Science, 37*(2), 107–123.

[17] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444.

[18] Nandanwar, H., & Katarya, R. (2025). Securing Industry 5.0: An explainable deep learning model for intrusion detection in cyber-physical systems. *Computers and Electrical Engineering, 123*, 110161. https://doi.org/10.1016/j.compeleceng.2025.110161

[19] Ni, C., & Li, S. C. (2024). Machine learning enabled industrial IoT security: Challenges, trends and solutions. *Journal of Industrial Information Integration, 38*, 100549. https://doi.org/10.1016/j.jii.2024.100549

[20] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

[21] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

[22] Priya, V., Thaseen, I. S., Gadekallu, T. R., Aboudaif, M. K., & Nasr, E. A. (2021). Robust attack detection approach for IIoT using ensemble classifier. *arXiv preprint arXiv:2102.01515*. https://arxiv.org/abs/2102.01515

[23] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems, 31*.

[24] Shyam, R., et al. (2020). Competitive analysis of the top gradient boosting machine learning algorithms. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE. https://doi.org/10.1109/ICACCCN51052.2020.9362846

[25] Surbhi, N., Chauhan, N. R., & Dahiya, N. (2025). Optimizing XGBoost hyperparameters using the dragonfly algorithm for enhanced cyber-attack detection in the internet of healthcare things (IoHT). *Cluster Computing, 28*(4), 230. https://doi.org/10.1007/s10586-025-04518-x

[26] Zhukabayeva, T., Ahmad, Z., Karabayev, N., Baumuratova, D., & Ali, M. (2025). An Intrusion Detection System for Multiclass Classification Across Multiple Datasets in Industrial IoT Using Machine Learning and Neural Networks Integrated with Edge Computing. In *Data, Information and Computing Science* (pp. 98-110). IOS Press. https://doi.org/10.3233/FAIA250008