Journal of Intelligent Systems and Internet of Things
Vol. 13, No. 02, PP. 272-292, 2024
DOI: https://doi.org/10.54216/JISIoT.130222

An Ensemble Boosting Algorithm based Intrusion Detection System for Smart Internet of Things Environment

Rami Baazeem
MIS Department, University of Jeddah, Saudi Arabia
Emails: rbaazeem@uj.edu.sa

Abstract

An influx of smart spaces that are now connected to the IoT network has increased new forms of cyber threats; thus, a need for more effective IDS to deal with these complex cyber threats. Traditional security measures cannot solve the modern problem of protecting IoT devices as they are a complex and homogeneously distributed network. Advancements and development of Artificial intelligent (AI) and machine learning technologies have provided new hope to make more reliable IDS. Our study presents Particle Swarm Optimization integrated Light-Weight Gradient Boosting Machine, abbreviated as LGBM-PSO in which, the PSO algorithm is applied for hyper parameters optimization in the model training. Based on the ensemble methodology, a new model for network intrusion detection is proposed in this study to improve the accuracy of the technique proposed. As for the current study project, the "DS2OS" dataset was employed to execute the suggested task. All of the data obtained from the traces of the smart devices placed in a smart home environment are incorporated in this dataset. The IDS model comprises several stages, one of which comprises data preprocessing that entails data cleaning, normalization, and encoding of network traffic data. Feature selection and dimensionality reduction are used which leads to the optimization of the dataset in this case. The core of the model comprises four classifiers: The compared models are Decision Tree (DT), LGBM-PSO, Light Gradient Boost Machine (LGBM), and Extreme Gradient Boost (XGB). Each of these classifiers can be combined with a majority voting ensemble method to increase the reliability of the predictions. The suggested model's accuracy that is LGBM-PSO is the highest with a value of 99.89%. The corresponding figures for the training data are 99.79%. Stand on the testing data proving the efficiency and stability of the algorithm. The use of the ensemble approach is superior especially when using a complex model like LGBM-PSO in the field of intrusion detection. As a result, high accuracy, optimized time, and effective threat identification ensure that it is a useful tool in strengthening security in the different applications.

Received: October 30, 2023 Revised: March 17, 2024 Accepted: July 12, 2024

Keywords: CS; Cybersecurity; Artificial Intelligence; Internet of Things; Smart Environment; IDS; LGBM; SVM; KNN

1. Introduction

Thus, there was extensive networking and ease in the construction of smart environments due to the progression of IoT in the most exceptional way. Smart homes and cities, automated industries, and made their way into people's lives and became part of the interconnected complexities like the health care system and more IoT devices. This technological revolution has been reported to have a lot of positives; efficiency gain, quality improvements, and new concepts that never existed before [1]. For example, a smart home allows people to control lights, temperature and security among other features, the smart city on the other hand works to minimize traffic flow, and energy usage and improve safety [2]. In the industrial sector, the Internet of Things (IoT) makes it possible to anticipate

equipment failure and monitor devices in real-time, both of which greatly improve operational efficiency and save operating expenses. The role of connected devices in healthcare relies on improving people's lives by assisting remote care, customizing treatments, and managing scarce resources [3]. However, the increased use of this IoT technology has also led to a proportional rise in the cases of cybersecurity threats concerning these devices. The attractiveness of IoT devices – connectivity of devices [4], data exchange and often limited processing capabilities are the causes of IoT's vulnerabilities to cyber threats [5]. These vulnerabilities are of course magnified by the heterogeneity of most IoT networks, including different makes and models of IoT devices and platforms, different operating systems, and different levels of security on boarded the devices. Furthermore, security is difficult to establish across different connected devices that are situated in various locations and different contexts.

Traditional security solutions that have been developed to counter networks with less divergent and highly centralized architecture are not sufficient for coping with IoT networks [6]. These measures conventional measures are mostly perimeter-based with a signature-based detection method which is inadequate for an IoT environment that is dynamic and decentralized. Hence IoT devices have become a favourite for hackers who want to misuse the different loopholes in the devices for various purposes such as stealing data, gaining access to other authorized IoT devices, and even causing disruption of services. This is because conventional security solutions fail to adequately protect IoT networks; there is therefore need to come up with better and more advanced IDS [7]. These massive amounts of data need to be monitored and understood by these systems instantly, detect and recognize the existence of seemingly 'out of the ordinary' behaviours and activities potentially demonstrating possible threats, and take the necessary action immediately. More specifically, AI-based IDS [8] is a better idea for improving IoT security because IDS is an effective approach for detecting a wide range of attacks by monitoring the network traffic and providing timely alerts. These systems can learn from prior mishaps, and they will be able to evolve as the type of threats change, unlike the conventional methods, the detection capabilities are likely to be more accurate and efficient.

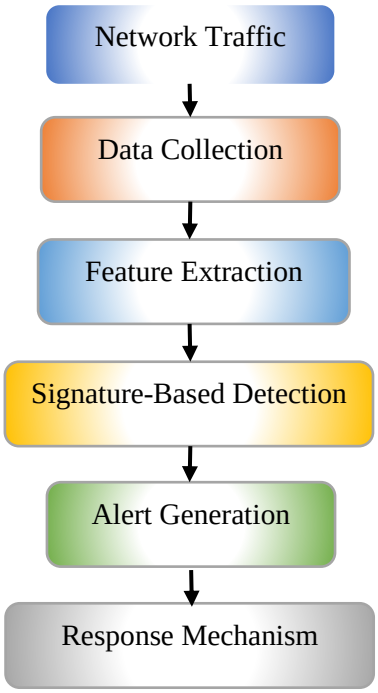


Figure 1. Traditional IDS Process Flowchart

Figure 1 represents the detailed process of a traditional IDS. It begins with Network Traffic, where data packets are exchanged across the network. These packets are captured in the Data Collection phase using packet sniffing tools that gather network traffic data for further analysis. In the Feature Extraction stage, key attributes such as IP addresses, port numbers, and protocol types are analysed and extracted from the collected data. These features are then compared against known attack signatures in the Signature-Based Detection phase to identify potential threats. If any suspicious activities are detected, the system moves to the Alert Generation phase, where alerts are generated to notify administrators of possible threats. The Response Mechanism phase involves taking appropriate actions to mitigate the detected threats, such as blocking traffic, logging events, or further notifying administrators to ensure network security.

Thus, a need to design a newer improved IDS in response to the current threat. All these IoT networks are on expansion, and they are continuously crawling into infrastructures deemed crucial for our daily functioning, which increases the degree and size of possible cyber threats. The popular approach of conventional IDS that focuses on standard set signatures, as well as rules to detect the threats, cannot meet the challenges of the neoteric and changing nature of contemporary cybercrimes [9]. These systems are suboptimal in understanding new threats as well as new classes emerging quickly in the threat space and consequently the probabilities of successful intrusions rise. IDS has been proven to be a critical tool in the advanced detection and threat control within a network; Artificial intelligence (AI) and machine learning (ML) technologies have allowed IDS to manage threats in a better and more effective way [10]. IDS applied by the use of artificial intelligence can be able to scan through large sets of data traffic in the network, recognize patterns as well as alert clients on extraordinary events that can be attributed to cyberattacks. Thus, this ability to learn and adapt is critical in the continually growing and changing IoT environment where devices and how they communicate with each other can vary greatly [11] [12]. This study is more concerned with the development of machine learning techniques using state-of-the-art deep learning algorithms to enhance IoT traffic anomalous behaviour detection using the LGBM model [11].

The motivation for this work derives from the increasing necessity of protecting the newly emerged and constantly developing Internet environments of Things. With connectivity being a core purpose of IoT devices, more dependencies arise in terms of privacy, safety and functionality that can be targeted by cyber-attacks as IoT devices find more uses in daily life. The usage of these devices has increased the exposed attack area that is occupied mainly by these devices as they remain at high risk from hackers because of poor security measures like improper user authentication and low encryption levels. Current traditional security approaches cannot adequately protect IoT networks because of the dynamicity of the IoT networks. Thus, this research will focus on discovering, evaluating and applying the modern approach encompassing the principles of AI and ML to create a flexible and predictive solution for modern and future security threats. IDS based on AI is capable of IS Zodiac learning from data, recognizing patterns, and improving over time to give far better protection than the existing conventional systems.

The purpose of this research is to encourage the development of optimal IoT networks that are secure enough to support the growing interconnection of society. In developing our IDS, we aim to have the capability to surpass current standards for IDSs by having short response times to threats and low rates of false positives. In this case, the focus is the improvement of IoT security to gain the necessary levels of public trust in the use of such technologies, proper usage and integration into different spheres of human life and the subsequent development of technology in general. The following are this study's principal contributions:

- The study introduces the LGBM-PSO model, leveraging Particle Swarm Optimization for hyper parameter tuning in a Light-Weight Gradient Boosting Machine framework. This approach significantly enhances the accuracy of detecting network intrusions in IoT environments, which are vulnerable to diverse and sophisticated cyber threats.
- By integrating the Decision Tree, LGBM-PSO, XGBoost, and LGBM classifiers through a majority voting ensemble method, the model enhances prediction reliability. This approach enables robust classification of network traffic into categories such as Benign and various types of Attacks, thereby bolstering cybersecurity defences in smart environments.
- The study provides a thorough evaluation of various performance metrics for intrusion detection. Through the use of criteria including recall, F1 score, specificity, accuracy, sensitivity, and precision, the study provides a thorough comparison of several models. The detailed analysis includes error metrics like MAE, MSE, and RMSE, offering a well-rounded assessment of model performance.
- The LGBM-PSO model demonstrates practical potential in real-world scenarios by effectively distinguishing between normal and malicious network behaviours.

The remainder of the document is structured as follows: The purpose of Section 2, the Literature Review, is to contrast artificial intelligence-based and conventional IDS methodologies. Section 3, referred to as Methodology, describes the data pre-processing, classifier training, and the experiment. The developed IDS's performance assessment and analysis are described in Section 4, Results and Discussion. Finally, Section 5, the Conclusion, presents the conclusions drawn from the study, the research contributions, and the potential research directions.

2. Literature Review

A wide range of approaches and conclusions are shown in the literature on IDS for IoT networks, highlighting both the advancements and existing challenges in the field. A unique intrusion detection system that uses machine learning (ML) to detect routing attacks against Routing Protocol for Low Power and Lossy Networks (RPL) is described in the article [13]. First, we simulate attacks on routing and capture traffic for many topologies. After processing the traffic, we provide sizable datasets for either two or more classes. For every assault, we choose a

subset of important features, and to train different classifiers to generate the IDS, we utilize this subset. LGBM-PSO had the shortest fitting time, according to the results of the 5-fold cross-validation trials, while DT, LGBM-PSO, KNN, and precision, accuracy, recall, and F1-score measures all yielded good results, with values more than 99%. On the other hand, the Multi-Layer Perception (MLP), NL, DT, and Deep Learning (DL) models' performance has noticeably declined.

In the work [14], we assess AI-powered intrusion detection systems for sensors that are wirelessly connected and keep an eye on vital applications. In particular, we provide a thorough examination of the techniques applied to machine learning-based invasive behaviour detection in the gathered traffic, reinforcement learning as well as deep learning. Through our simulations, we assess the suggested processes using the KDD'99 real assault dataset. Reinforcement learning as well as deep learning: The Q-learning-based IDS (Q-IDS) and the Clustered IDS based on Restricted Boltzmann Machine (RBC-IDS), and the Clustered Adaptively Supervised Hybrid IDS (ASCH-IDS). Furthermore, we demonstrate the effectiveness of several reinforcement learning techniques, like Temporal Difference Learning (TD) and State-Action-Reward-State-Action Learning (SARSA). Using simulations, we show that Q-IDS has a detection rate of about, whereas SARSA-IDS and TD-IDS have a detection rate of around.

The study [15] looks at cyber threat detection using artificial intelligence to safeguard today's digital ecosystems. Within the fields of mobile security network security, and security for the Internet of Things. The main objective is to assess ML-based classifiers and ensembles for anomaly-based malware detection and network intrusion detection, as well as how to combine those models. The presentation examines the challenges associated with integrating AI-enabled cybersecurity solutions into the corporate processes and IT infrastructures that are in place today, as well as solutions to those difficulties. To significantly enhance the resilience and security of our modern digital businesses, infrastructures, and ecosystems, the article concludes with research directions for the future.

The study [16] will offer a thorough synopsis of the problems related to IoT network security, including standard communication protocols for Internet of Things systems, examples of recent attacks on IoT technology and the use of AI in IoT security. For the first time, all of the key components of IoT security are examined and discussed collectively, along with possible AI-based solutions. Future research aimed at creating more secure IoT communication protocols and artificial intelligence (AI) solutions for IoT security and privacy management will greatly benefit from this study's comprehension of important points of view.

The research [17], in contrast to previous works, uses SVM for Vehicular ad hoc networks (VANET) intrusion detection. Numerous computational benefits come with the SVM structure, including irrelevance between algorithm complexity and sample dimension and specific direction at a finite sample. VANET intrusion detection is a combinatorial, nonconvex issue. Thus, three intelligence optimization approaches are applied to optimize the accuracy value of the SVM classifier. These optimization approaches include genetic algorithms (GA), particle swarm optimization (PSO), and ant colony optimization (ACO). Our findings show that GA performed better than alternative optimisation strategies.

The study [18] explores the possibilities, limitations, and advantages of ML and DL approaches for IoT device security. Several solutions investigate hardware-based techniques for secure offloading, and IoT infrastructure security using machine learning (ML)-based methods for malware detection, access control, and authentication. This review seeks to shed light on the merits of different strategies for tackling IoT security in a way that is genuinely efficient, adaptable, and seamless.

This paper [19] presents a comprehensive picture of "AI-driven cybersecurity," which, depending on these AI technologies, may be necessary for smart cybersecurity management and services. Compared to traditional security solutions, security intelligence modelling based on AI methodologies may make cybersecurity computing more automated and intelligent.

The methodology, key conclusions, and limitations of several research on IDS for IoT networks are compiled in Table 1. These investigations used a variety of ML and DL methods. Study [20] performed a systematic review from 2014 to 2021 and incorporated ML and DL approaches for IDSs and identified IDS weaknesses such as scalability, network delay, IoT device limitation in resources, and the importance of semi-supervised and reinforcement learning. [21] applied IoT traces and attack datasets to obtain high accuracy in identifying cyber-attacks while facing problems of small IoT resources and large computations. [22] which mainly focused on Deep Belief and Neural Networks, reported high precision of attack detection but they considered some problems such as location-dependent attacks such as ID cloning and spoofing as unknown. [23] used SVM, Decision Tree, Naive Bayes, ANN, and CNN for the anomaly detection that enhanced the threat detection process without expounding the study’s limitations. [24] employed CNN, LSTM, and GRU models and reported outstanding results on the Bot-IoT dataset with very few false alarms, but the authors noted that they faced difficulty in multi-class classification and keylogging protection. [25] presented a literature review for a hybrid IoT security model, and noted that there are still issues with the existing models and more research is needed to examine the security issues. [26] discussed

ML and DL with challenges including limited availability of good datasets and challenges in the development of near real-time IDSs. Last of all, [27] a database search was conducted to assess DL approaches for IoT anomaly detection but the search was conducted in journals only and excluded other types of publications.

Table 1: Main finding, Methodology & Limitations of related works

Ref.	Methodology	Main Finding	Limitations
[20]	Looked through databases (2014–2021) for articles utilizing ML and DL methods.	IoT network IDS robustness	Scalability, network latency, resource limits with IoT devices, need for semi-supervised and RL for IoT IDS.
[21]	Used IoT traces and attack traffic dataset, utilized packet header fields as features	High accuracy in detecting cyber attacks	IoT resource constraints, computational limitations
[22]	Applied Deep Belief Network and Deep Neural Network for high precision in attack detection	Achieved high precision, recall, and F1-scores across attacks	Limited in detecting location-dependent attacks like ID cloning, spoofing, sybil attacks
[23]	SVM, Decision Tree, Naive Bayes, ANN, CNN for anomaly detection in IoT networks	Improved threat identification and mitigation	Not explicitly stated
[24]	Employed CNN, LSTM, GRU for classification, trained on 80% of data and achieved 99.8% accuracy on the Bot-IoT dataset	High accuracy with minimal false alarms, especially for CNNs	Challenges in FNNs for multi-class classification, protection against keylogging
[25]	Conducted literature review on AI-based techniques, proposed hybrid framework for efficient IoT security model	Risk factor analysis proposed hybrid security model	Due to their limitations, current security models need to be further examined in terms of their problems and difficulties.
[26]	Reviewed ML and DL techniques in IoT IDS, emphasized design decisions, benefits, drawbacks, and detecting techniques.	Identified research challenges in IoT IDS	Lack of high-quality datasets, challenges in developing online and real-time IDSs
[27]	Automated search of databases, assessed study quality using 3-tier schema	Effective use of 7 DL techniques for IoT anomaly detection	Limited to journal articles, excludes books, magazines, conferences

The literature reveals some of the major gaps in the current research on IDS for IoT networks. The existing IDS solutions cannot handle scalability and real-time processing as the number of IoT devices increases and they have limited capabilities in terms of computation. The datasets used for training IDS models are not of high quality, and the existing models are not effective in identifying new and advanced attacks. False positive and negative rates are still high, and these issues need to be addressed to increase the effectiveness of the methods. New and more sophisticated AI techniques such as reinforcement learning are not well researched, and the inclusion of the IDS with AI into current IT systems is not easy. Also, there is a need to improve the models' performances in multi-class classification and identifying certain kinds of attacks. Filling these gaps is essential for improving IoT security and readiness for various threats.

3. Materials and Methods

With the use of IoTs, the present research seeks to create an intrusion detection model for smart settings. The DS2OS dataset is used in the suggested method for intrusion detection in Internet of Things systems, which includes 357,953 instances of network traffic with 13 features, separated into one benign class and seven assault classes. The approach begins with comprehensive data pre-processing involving data cleaning to remove irrelevant instances and fill missing values, data transformation to encode categorical features into numeric formats, and feature selection to find and keep the most pertinent characteristics for examination. The pre-processed data undergoes feature engineering to extract, scale, and optimize features, addressing the high-dimensional nature of the dataset. The approach uses some classification models such as the DT, XGB, and the LGBM as highlighted above, and the major LGBM-P model that combines LGBM with PSO techniques. The optimization used in the current work entails PSO as the parameter initialization, fitness evaluation using K-fold cross-validation, and the optimization of the particle position for the best parameter estimation of the model. The last model also applies the ensemble technique through the aggregation of classifiers where the final decision is made by a majority vote. The model's reliability and efficiency can be confirmed through cross-validation, whereby K-fold and Repeated Strata K-fold are the best. This approach aims to achieve high accuracy and low latency in detecting various types of network intrusions in IoT environments.

3.1 Dataset Description

We used the DS2OS dataset. The "DS2OS" dataset [28] was created by employing four simulated IoT sites with certain service kinds to collect traces in the IoT environment. This dataset combines benign (normal) and harmful (malicious) traffic. The "DS2OS" dataset catalogues a novel class of threats originating from real-world traffic in IoT systems. Conventional network traffic attacks serve as an inspiration for the attack's features. Based on their behaviour, the sets of recorded network traffic samples are labelled. Among all the records that are acquired, feasible samples (features) are chosen, and every record that is identical to a sample is chosen. Has been given a classification (normal or harmful). The essential information about the "DS2OS" dataset is shown in Table 2. Here, connected nodes have reported two kinds of behaviour: "normal" and "malicious." Attacks of several kinds impact nodes exhibiting malicious behaviour. Based on

Table 2: Essential information about the "DS2OS" dataset

Dataset Name	Total Feature	Total Number of Cases	Specific Features	Total Number of Classes	Anticipated Assaults
mainSimulationAccessTraces (DS2OS network Traffic)	13	357953	12	7 (attack classes) + 1(benign class)	DoSattackmalicious control, , data probing, malicious operation, scan, spying, wrongSetUp

The characteristics of the collection of records that were sent to the dataset, these assaults have been categorised. By using the training approach, Numerous prominent assaults have been recognized and categorized as "scan," "malicious operation," "data Probing," "DoS attack," "spying," "malicious Control," and "wrong setup."

3.2 Data Preprocessing

The most important step in preparing a raw dataset for studies is data pre-processing. It is a crucial stage in every project when performance is to be increased [29]. In general, sufficient time and effort are dedicated to the preparation of data. Additionally, this procedure is necessary to provide a fitted labelled dataset that is intended to be developed to provide high-quality analytical findings while lowering complexity [30]. Models created Results using ML approaches can be obtained quickly and with excellent accuracy. The following actions are taken when the data is being pre-processed.

At first, there's a chance the dataset isn't in the right format. Data transformation and cleaning can improve its usefulness. If a dataset has repeated, redundant, extraneous, or incomplete data entries, processing it might be expensive and time-consuming. These entries are recognised and have the option of being deleted or having new values added. These values can be filled in and replaced in a variety of ways. Certain strategies are used to transform values of inconsistent data. Data that is missing, categorised, skewed, or transformed may handle data that is non-alphanumeric, either as a string or in both forms. The two methods for category encoding that work well are label-encoder and one-hot encoder. Data de-noising and filling in null or missing information are two essential phases in data cleaning.

Data cleaning: Relevant and appropriately suited data can be used to substitute noisy data or to remove it entirely. The dataset contains data that can be chosen based on suitability to replace noisy data. The data. The drop () function has been used in Figure 8, line no. 6, to eliminate unnecessary columns from the dataset.

3.3 Feature Engineering

High-dimensional datasets may complicate things further in terms of space and time. It is strongly advised to use the feature selection strategy to get over the complexity difficulties. The process of developing a new set of features based on previous features in accordance with project goals is known as feature engineering. [31]. The three main tasks of feature engineering are feature extraction, feature scaling, and feature relationship capture. Processing the entire amount of data that has been gathered takes too much time. As a result, among all the characteristics in a dataset, the most important features or attributes are chosen for study. During processing, features that are not helpful for the model might be excluded [32]. It is possible to remove entries that contain null or irrelevant values. The analysis's complexity is decreased throughout this stage. The majority of potential features are dynamically chosen from the complete dataset. In machine learning and predictive modelling, dimensionality reduction is important for data compression as it lowers computing time and storage requirements. It describes how to create a set of primary examples in order to reduce the number of attributes.

Feature selection: The basic idea behind feature choice is selecting the greatest significant characteristics (features) from the dataset and eliminating the less significant ones that have no bearing on the model's performance. If less helpful or irrelevant features are chosen, the model's performance may decline. Less overfitting (less redundancy also reduces noise), shorter training times (less complicated algorithms), and higher model accuracy (better results by minimizing misleading data) may all be achieved with optimal feature selection. Numerous techniques are available for feature selection, such as single-variable selection, feature significance, as well as a correlation matrix and heat map.

Scaling of features: Out of the thirteen attributes, twelve have been employed in activities that include target variables. Replacement of unexpected values (including "false," "true," "none," and string-type values) with the best float values is required. While the "label Encoder ()" method assigns an integer value of "Y," the "OneHotEncoder" approach modifies the category column of "X" in Label Encoding." Due to its 'iloc ()' technique, any column in the dataset may be handled using the Pandas module. It makes it possible to choose values from a certain dataset row and column. There are eight classes: one is the benign or normal class, while the other seven are different forms of attacks. Several sub-parameters must be adjusted to achieve the best classification.

3.4 Classification Algorithms

Numerous techniques for predictive machine learning and decision-tree-based classification exist, and they can be categorised as individual or ensemble classifiers. Three linearly separable ensemble classifiers and a single class classifier will be the subjects of a simulation by the writers in this work. Among the ensemble classifiers are SVM, KNN, and LGBM. A classification method known as an ensemble learning classifier integrates several base models to produce an optimal single classification model.

3.4.1 Decision Tree

Another ML algorithm is the decision tree. It breaks down the data into progressively smaller nodes and is a tree structure classifier, as its name suggests. It is made up of two components: decision leaves and nodes. The decision's results are leaves. It may be used for regression and classification issues. Entropy quantifies the degree of uncertainty or impurity in data samples. It is calculable as:

$$H(S) = \sum_{y \in X} p(y) \log 2 \frac{1}{p(y)} \quad (1)$$

The entropy of information gain $IG(S, A)$ for a set S is altered in Eq. 2, specifically for feature A. The features on which to separate his nodes to come closer to the target variable's prediction are determined using entropy and IG. It also indicates the end of splitting.

$$IG(S, A) = H(S) - \sum_{i=0}^{n} P(y) \times H(y) \quad (2)$$

DT is often constructed using the ID3 and C4.5 algorithms.

Figure 2 shows the overview of the Decision Tree model.

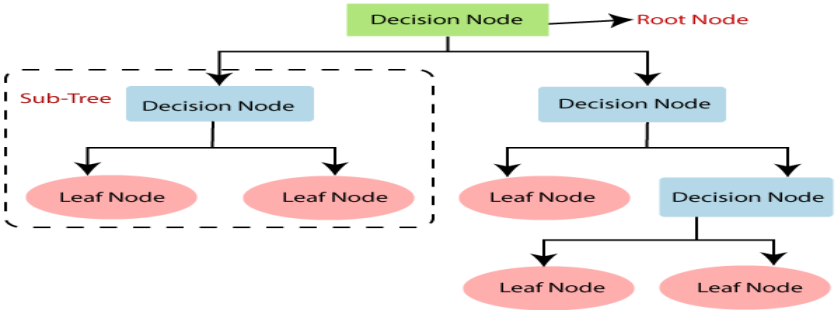


Figure 2. Overview of the Decision Tree Model

3.4.2 Gradient Boosting Machine

The gradient boosting machine, or GBM, is a technique for gradually improving inaccuracy. Freidman [17] created GBM and identified, $y = \eta(s(t)$ as the approximation of the functional dependence. How stable the categorization process is is explained by the loss function. The loss function should be lowered in the direction of the gradient to enhance the classification model. The GBM classifier was used to train the classifier in this investigation. $M = \{m_p, p = 1,2, \ldots \ldots, N\}$ represents the training data set, where N is the total number of patients and reflects the feature vector of chosen characteristics. In this paradigm, the loss function $\psi(y, \eta)$ is represented as

$$\hat{\eta}(s(t)) = \hat{y} = \arg min \psi(y, \eta) \qquad (3)$$

The function estimation,$\hat{y} = \sum_{i=1}^{M} \hat{y}_l$ is parametrizes with $\hat{y}_l$ it provides a lift. The approach known as greedy was developed, estimating. $\hat{y}_k = \hat{y}_{k-1} + \Delta_k . \xi(\overline{s(t)}, \theta_k)$ at every recursion, where the decision tree-like base learner, $\xi(\overline{s(t_l)}, \theta)$

$$(\Delta_k \theta_k) = arg\ min_{\Delta,\theta} \sum_{i=1}^{N} \psi\left(y^{(1)}, \hat{\eta}_{k-1}\right) + \Delta . \xi(\overline{s(t_l)}, \theta) \qquad (4)$$

Given that optimization poses a challenge to both the general loss function and the base learner, Together with the perceived data, Friedman suggested a unique function $\xi\left(\overline{s(t_l)}\right), \theta$ that is the closest to being parallel to the negative gradient, whereby the traditional least square minimization method is applied to the optimization procedure. In Table 3, the GBM algorithm is displayed.

Table 3: Machine method for gradient enhancement

An algorithm for gradient-boosting machines
Data: statistical features $\{(\overline{s(t_l)}$,n observed data features T-F$\}$ feature Process: Determine the number of iterations M for the loss function $\psi(y, \eta)$and the base learner classifier $\xi(\overline{s(t)}, \theta)$. Build the predictive classifier. $\hat{\eta}\left(s(t)\right)$ for $\overline{s(t)}$ Initialize $\widehat{\eta_0} = arg\ min_{\Delta_k} \sum_{i=1}^{N} \psi(\overline{s(t_l)}, \Delta_k)$ for $m\epsilon\{1,2, \ldots, M\}$ Determine the gradient's negative value. $\zeta_k(s(t))$ Add a new basic learning feature. $\xi(\overline{s(t)}, \theta_k)$ Determine which Δ_k Gradient descent step size is optimal for generating a tree classifier. $\Delta_k = arg\ min_{\Delta\theta} \sum_{i=1}^{N} \left(y^{(i)}, \eta_{k-1}\left(\widehat{s(t_l)}\right)\right) + \Delta . \xi(\overline{s(t_l)}, \theta_k)$ Update function $\eta_k = \Delta_k \zeta_k(s(t))$ and the GBM classifier $\eta\left(\overline{s(t_l)}\right) = \eta_k + \eta_{k-1}$ end for. Return $\eta(\overline{s(t_l)})$;

3.4.3 Extreme Gradient Boosting

Extreme Gradient Boosting, or XGBoost, is a well-liked and effective gradient-boosting method that has garnered a lot of traction in both real-world applications and machine-learning contests. Developed by Tianqi Chen, XGBoost builds on the principles of gradient boosting and introduces a range of enhancements aimed at improving performance, scalability, and flexibility. This section delves into the key features, architecture, and advantages of XGBoost. Some of the key features of XGBoost are as follows:

- Regularization: To avoid overfitting, XGBoost incorporates L1 (Lasso) and L2 (Ridge) regularization. This regularization term helps in smoothing the final learned weights to avoid overly complex models.
- Parallel Processing: Unlike traditional GBM, XGBoost can perform parallel computation, greatly accelerating the training procedure. This is accomplished by building decision trees in parallel.
- Tree Pruning: XGBoost employs a novel tree pruning technique called "max depth pruning" rather than the traditional pre-pruning approach. This enables the algorithm to bring out trees to the required depth and then cut backwards to arrive at the right structure.
- Handling Missing Values: It should also be noted that XGBoost is a method that works very well with missing values. It has the capability of discovering its strategies for dealing with missing data during the training phase and is capable of tackling sparse data.
- Weighted Quantile Sketch: It is used by XGBoost to handle weighted data points and come up with the right split points for new decision trees. Cross Validation: Additionally, XGBoost has built-in cross-validation tools that make it simple for users to evaluate the model's performance and modify the hyper parameters.

3.4.4 Light Gradient Boosting Machine

Light GBDT or Gradient Boosting Decision Trees utilizing histograms aiming at reducing the usage of the machine memory and the time of execution while improving the performance of the model. It was noted that LGBM is significantly better optimal when compared to a few other boosting ensemble decision tree techniques that are currently in use [33]. It is a much enhanced, more dispersed, potent, and quicker learning algorithm. Large data flow can be effectively handled by LGBM [34]. LGBM employs a pre-sorted approach for generating the superlative split and a histogram-based algorithm for decision tree learning, similar to other more boosting techniques [35]. Exclusive feature bundling and gradient-based one-side sampling (GOSS) are two techniques used by LGBM, two new methods. (EFB). To identify a split value by dividing the data samples, GOSS down samples the instances based on the gradient sizes.

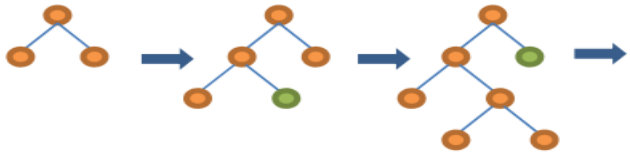


Figure 3. Leaf-wise tree growth.

Big gradient samples are the model's main emphasis, as tiny gradient samples are removed. Big gradient samples lack proper training, However, well-trained samples have moderate gradients. The accuracy of this procedure is higher than that of uniform random sampling. However, the drawback of conventional histogram-based algorithms is addressed by EFB. While conventional decision trees grow level-wise (depth-first), the LGBM method grows leaf-wise (best-first). Figure 3 depicts the decision tree's growth leaf by leaf. The diagram's green nodes show where tree growth is occurring. For leaf-wise expansion, the greatest delta value should be chosen. ML issues including decision-making, regression, and classification can all be solved with LGBM [36]. Compared to level-wise tree growth, leaf-wise tree growth minimises loss or mistakes.

3.4.5 The proposed LGBM-PSO

In this paper, we provide a unique method for creating an enhanced intrusion detection system by fusing LGBM with PSO, referred to as the LGBM-PSO model. LGBM is a sophisticated gradient-boosting technique that excels in efficiency and effectiveness for large-scale data tasks. Its key features include leaf-wise tree growth, which enhances model performance by focusing on leaves with the greatest potential for improving the model's accuracy, and histogram-based decision trees, which optimize computational resources and speed up the learning process. To improve the performance of LGBM, the LGBM-PSO model integrates PSO, which is a metaheuristic optimization algorithm based on the social behaviour of a flock of birds. PSO works by having a population of possible solutions or particles; where each particle is a potential set of hyper parameters for LGBM. These particles sample the hyper parameter space using iterative movement, which results from their own best solution and the swarm's best solution. The essence of this process is to find the best hyper parameters for use in LGBM. The general working of the LGBM-PSO model mainly focuses on the incorporation of the feature of high learning from the LGBM with the improved optimization method of PSO. The enhancement of LGBM's hyper parameters using PSO delivers the model a better performance in identifying many types of network attacks. The final model is then developed with the optimized parameters to create an efficient IDS for dealing with several security threats. Besides improving the detection performance, the combination of these two techniques also guarantees that the model is effective and competent in practice.

3.5 PSO-based Optimization

Based on how birds behave as predators in a small search area, Particle swarm optimization (PSO) [37] belongs to the category of evolutionary computation. Every particle in PSO is a solution for the objective problem; it represents a search for a bird in its natural environment.

You may think of each particle i as a mixture of two characteristics. The position and velocity attributes, which are represented by the following vectors, respectively: the position vector. $X_i = x_i^1, x_i^2, \dots, x_i^D$ and the velocity vector $V_i = [v_i^1, v_i^2, \dots, v_i^D]$ where D denotes the search space's dimensions. Because each particle's beginning velocity and starting position were initially assigned at random using the PSO algorithm, the particle swarm's first journey across the search space is chaotic. As the iterative process moves on, each particle will communicate the information found during the search to other members of the population, causing the population as a whole to evolve towards the best possible solution to the goal issue. Particle I's optimal placement (p_{best}) and the population's overall optimal position (g_{best}), during the iteration, two additional vectors kept in the algorithm, are helpful to this mechanism. According to Eqs. (5) and (6), respectively, every iteration will give every particle a new position and velocity depending on its prior position and the two previously indicated optimum positions:

$$v_{id}^{t+1} = wv_{id}^{t} + c_1 r_1 (p_{best\ id}^{t} - X_{id}^{t}) c_2 r_2 (g_{best\ id}^{t} X_{id}^{t}) \quad (5)$$

$$X_{id}^{t+1} = X_{id}^{t} + v_{id}^{t} \quad (6)$$

Where t denotes the current iteration, w is the inertia weight, the acceleration coefficients c_1 and c_2 determine the maximum distance a particle may travel in one repetition. And for the d^{th}dimension, at each update, two independent random numbers, $r_1$and r_2, are created in the interval [0, 1]. The particle's current location and velocity are denoted by X_{id} and v_{id} respectively.

Table 4: Setting the optimization algorithm's parameter size

Parameter	**Value**
Maximum iteration count;	100
Number of particles	10,20,30,40,50,100,150,200
Inertia weight, w	$w_{min} = 0.4; w_{max} = 0.9$
Maximum particle velocity:	6
n_estimators	Lower bound=20; Upper bound=250
Min_sample_split	Lower bound=3; Upper bound=200 Lower bound=2; Upper bound=100

First, as shown in Table 4, establish the model's starting values for the parameters that are required. Next, produce a beginning particle population of a specific size. This population will serve as the basis for next iterations. Eight PSO-GBDT models were created in this work, each with a distinct particle count determined by the optimization technique, by the notion of preventing unintentional mistakes throughout the experiment. Each possible solution (particle) must be evaluated for quality, and the fitness function must be included. It has long been known that this technique may greatly increase a model's reliability and extract as much useful information as possible from the sparse input, this study needs cross-validation, or CV. In this study, the hybrid model's fitness function was determined to be the average accuracy attained during fivefold cross-validation. An input sample set can be divided into five groups using a fivefold CV; the four remaining groups are then utilized as training sets, with each group serving as a test set alternatively. The hybrid model produces the best answer when the termination condition is met, or when there are 100 iterations.

3.6 The proposed Ensemble Model

The ensemble model depicted in Figure 4 is designed for intrusion detection, comprising several integral components. Firstly, data pre-processing and normalization involve cleaning, transforming, and normalizing raw data to ensure consistent input for subsequent stages. This step is crucial for maintaining data integrity and uniformity. Next, feature selection and dimensionality reduction extract relevant features from the dataset, reducing its dimensionality to improve computational efficiency and model performance. The fundamental of this model is at the classifiers training component where four classifiers are developed on labelled training data that include; DT, XGB, LGBM, and LGBM-PSO. Every classifier tends to recognize patterns associated with non-

nerve-racking behaviours or other types of assaults. Majority voting, in the evaluation and decision-making stage, is the method used to combine the predictions of these classifiers. This ensemble technique increases the efficiency and reliability of the model and, therefore, makes the final identification of the input as being either benign or an attack more accurate. Last of all, inputs are described as belonging to certain categories like 'Benign, 'Attack 1,' 'Attack 2' and 'Attack n' Some of the insights that can be elicited include the kind of intrusions that have been detected. Ensemble can be effective in security applications, as it performs better in terms of detection and is more accurate and less sensitive to variations as a result of using multiple classifiers.

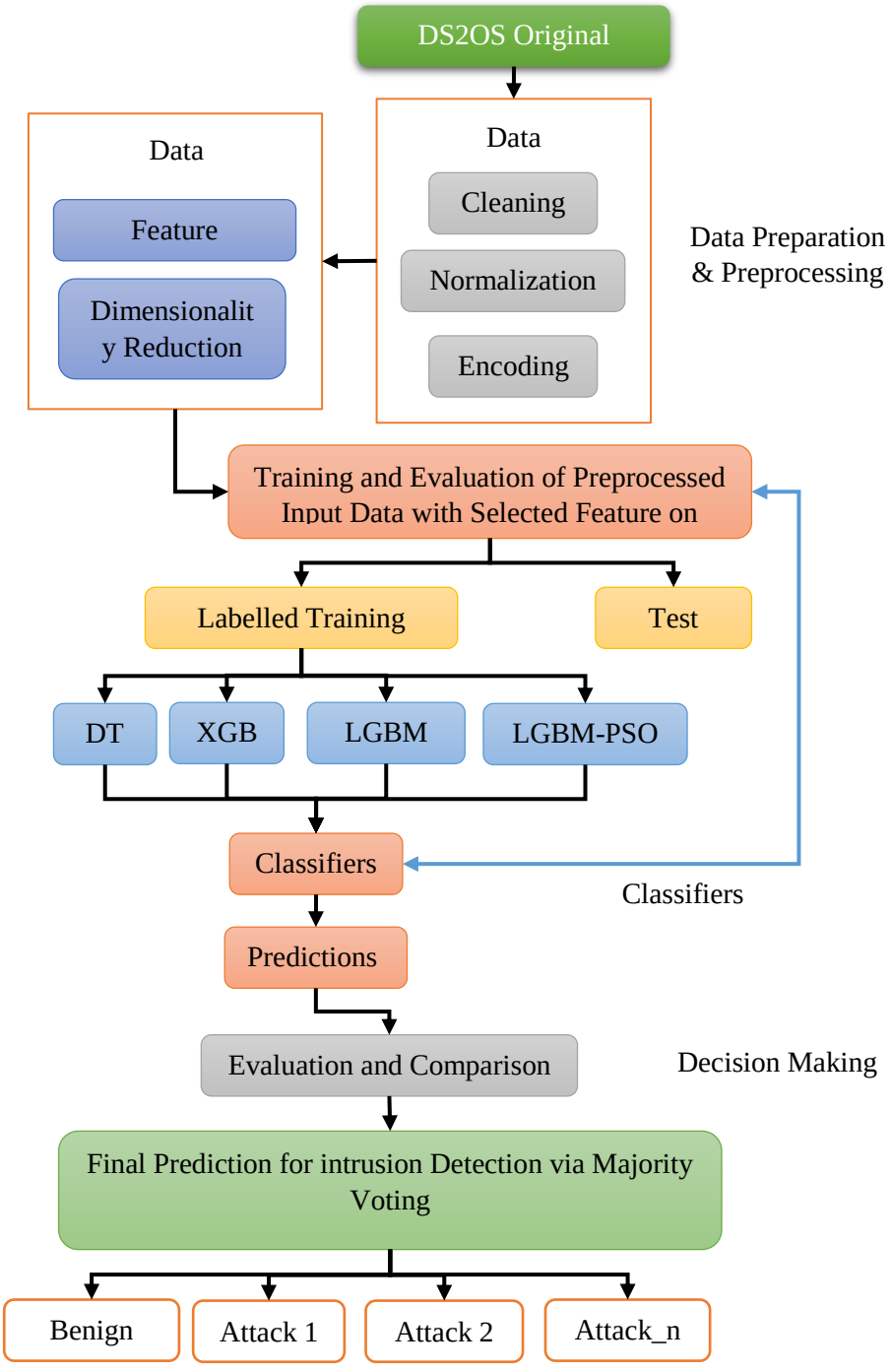


Figure 4. The proposed Ensemble-based intrusion detection model.

3.7 Model Training

The verifiability of the model can be as follows: Nevertheless, the best approaches include the stratified K-fold cross-validation and the simple K-fold cross-validation. The input dataset is divided into k groups or partitions of

equally sized samples in what is known as the k-fold cross-validation. However, this approach sometimes makes the model noisy while splitting the large dataset. Increasing the k-value reduces the noise level. “RepeatedK-fold” "K-fold repeated" Applying the k-folds n times with various random states is how CV works. This is how the “RepeatedStratifiedK-fold”, which is a kind of CV, works: the data is rearranged in such a way that each fold is representative of the whole dataset. Instead of dividing randomly, it divides in a stratified manner. It is the most efficient approach to managing fluctuations and prejudice.

This skeletal overview of the proposed model should imply how one can validate the model using certain validation techniques, out of which the most reliable techniques include the stratified K-fold cross-validation and the other K-fold cross-validation. K-fold cross-validation divides the input set into k equal folds also known as subsets also known as iterations apart from the training set. In k-fold cross-validation, fold is regarded as the validation set and the remaining k-1 folds as the training set. This leaves the other k-folds of the data and uses each of them only once as the validation data. This kind of technique enables the user to get a good different estimate of the model through validation as well as training of each of the data points. Nonetheless, the use of K-fold cross-validation can add some level of noise when it is used with huge datasets mainly because of the randomness in the allocation of folds. To avoid this noise, one can be advised to raise k-value which implies that each set is split into smaller subsets and hence a more stable estimate of performance is attained. To advance the validation of the model, Repeated K-fold cross-validation can be used. This technique is performed n number of times by using different random seeds each time for the K-fold process. This kind of repetition aids in providing a more accurate and reliable estimate of the model's performance above the average. A slightly better algorithm is known as Repeated Stratified K-fold cross-validation, especially used when dealing with the volatility and bias in datasets with unbalanced classes. In this method, the given database is divided in a way that each fold has the same proportion of classes that is present in the whole large database. This stratified sampling is repeated n times to ensure a broad assessment of the model’s performance at different data splits while maintaining the class distribution. Thus, by applying these cross-validation techniques, it is possible to make sure that the model is tested thoroughly and improved for the best performance. These methods not only give an accurate and strong validation of the model performance but also in tuning and biassing the model to get better results when the model gets challenged with new data.

4. Experimental Results

The selected dataset has been used to test the specified model, namely “DS2OS”. To qualify the suggested model profoundly and to partition data into relevant categories, statistical classification is employed either on a subset of data or the data as a whole. First, split ratios for the training and testing sets were 80% and 20%. The train-to-test data ratio has been determined to be 70% and 30%, respectively, in a subsequent round. Still, there wasn't much of a difference in the outcomes.

The simulation on Google Colab was performed using Python 3.3 and TensorFlow v2.13.0. The system configuration included an 11th Gen Intel® Core™ i5 processor, 16 GB of memory, and 145 GB of storage, operating on Windows 11. For graphical processing, the setup utilized an NVIDIA GTX 750 GPU.

This section compares several ML classifiers in great detail. The model primarily focuses on prediction and classification using gradient-boosting methods. To evaluate the algorithms, a proper ratio of training and test data is used, and the resulting numbers represent several performance measures (accuracy score, speed (runtime), and error determination). By achieving the ideal values for the hyper-parameters, the results are achieved for each parametric performance. The following measures have been used to assess the prediction performance.

4.1 Evaluation Metrics

The efficacy of the proposed intrusion detection model is evaluated based on many factors. The results of the proposed model are examined at the analysis step. The ensemble classifier for the suggested IDS model is assessed using accuracy metrics on the "DS2OS" dataset, and the effectiveness of time, Rates of false positives (FPR) and true positives (TPR). Regarding the real findings, TPR and FPR are associated with the results that are categorized and misclassified.

4.1.1 Accuracy

The accuracy of the harmful and normal indexes is shown in this study as a percentage. Equation 7 appears at the bottom of the next page, contains a mathematical formula that can be used to calculate accuracy.

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

4.1.2 Sensitivity or Recall

The proportion of attack samples that have been accurately identified as malicious out of all attack samples is known as sensitivity or recall. The mathematical formula for determining sensitivity is provided in Equation 8.

$$Recall\ or\ Sensitivity = \frac{TP}{TP+FN} \quad (8)$$

4.1.3 Specificity

Specificity is defined as the percentage of all benign samples that have been correctly classified as benign. Equation 9 provides the mathematical formula for calculating sensitivity.

$$Specificity = \frac{TN}{TN+FP} \quad (9)$$

4.1.4 Precision

The percentage of correctly detected attack samples among all samples that have been classified as attacks is known as precision. Equation 10 provides the precision formula in mathematics.

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

4.1.5 F1 score

The harmonic mean of recall and accuracy is the F1 score, sometimes referred to as the F-measure. The formula for computing the F1 score is given by Equation 11.

$$F1\ score = 2 \times \frac{Recall \times Precision}{Precision + Recall} \quad (11)$$

In this case, True Positive denotes the percentage of malicious nodes that are appropriately categorized as dangerous. The number of benign nodes that are accurately classified as benign is represented by True Negative. The amount of hostile nodes that are incorrectly categorized as benign is known as False Negative (FN). False Positive (FP) refers to the quantity of innocuous nodes that are incorrectly categorized as malicious.

4.1.6 ROC-AUC (TPR VS. FPR)

The true positive rate (TPR) and false positive rate (FPR) indicate the rates at which events are discovered. Another name for TPR is sensitivity and recall. Equations 12 and 13 provide the formulas that are used to calculate TPR and FPR, respectively. While FPR shows the false alarm rate, TPR represents the true outcomes.

$$Sensitivity\ or\ TPR = \frac{TP}{TP+FN} \quad (12)$$

$$FPR = \frac{FP}{TN+FP} \quad (13)$$

4.2 ROC-AUC Results

Figure 5 illustrates the ROC-AUC scores of four models: DT, XGB, LGBM, and LGBM-PSO. The DT model has the lowest score at 68.98%, indicating poor performance compared to the others. The XGB and LGBM models both demonstrate excellent performance with scores of 99.95% and 99.92%, respectively. The LGBM-PSO model achieves the highest score at 99.98%, showing that optimization enhances performance. This graph highlights the significant performance differences, emphasizing the superiority of advanced models and optimization techniques in classification tasks.

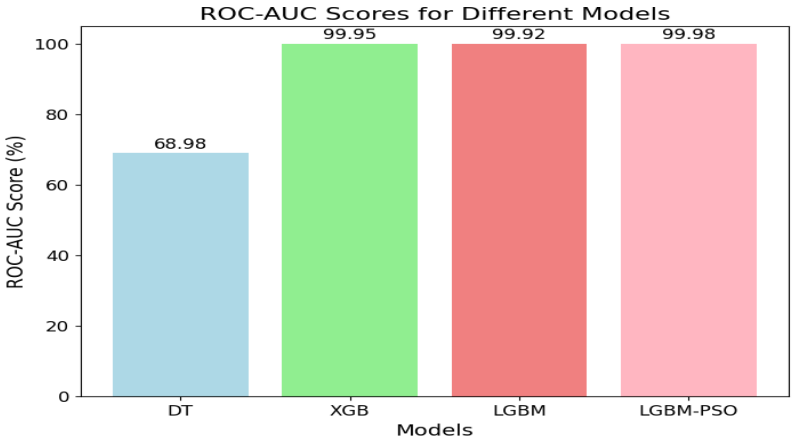


Figure 5. ROC-AUC Scores for Different Models

The TPR and FPR results for DT, LGBM-PSO, In Table 4, XGB and LGBM are displayed. The "mainSimulationAccessTraces.csv" dataset was examined. A good TPR should be close to 1, while a good FPR should be close to 0.

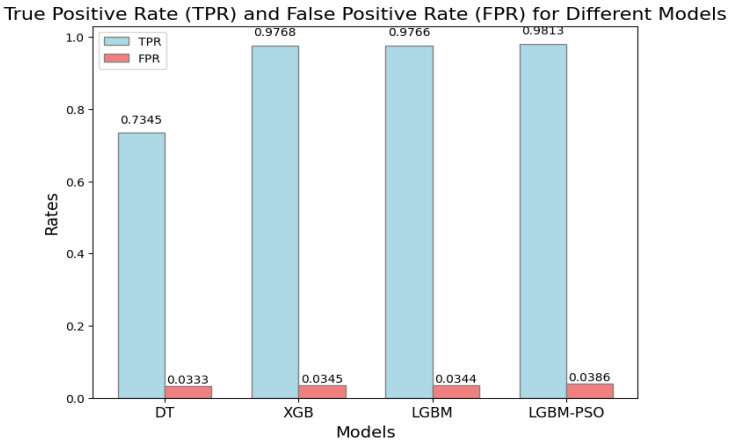
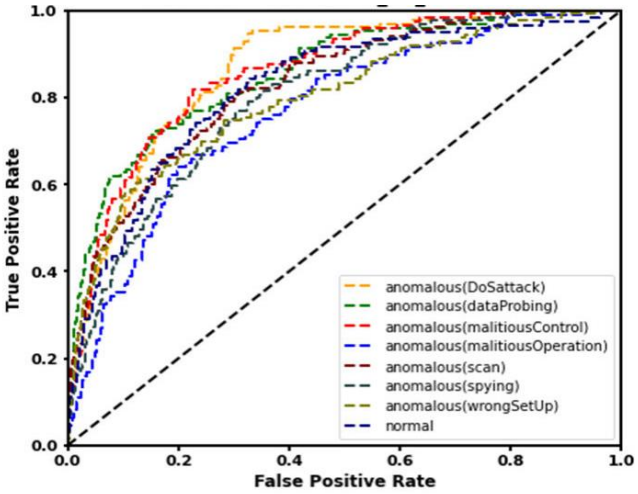


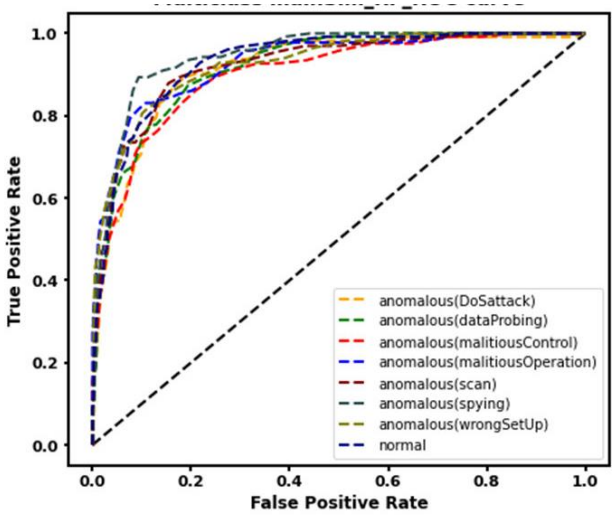
Figure 6. TPR and FPR for Different Models

Figure 6 compares the TPR and FPR for four machine learning models: DT, XGB, LGBM, and LGBM-PSO. The TPR indicates the proportion of actual positives correctly identified, with the LGBM-PSO model achieving the highest TPR of 0.9813, followed closely by XGB and LGBM with scores of 0.9768 and 0.9766, respectively. The DT model has a lower TPR of 0.7345. In terms of FPR, all models exhibit low values, with DT having the lowest at 0.0333 and LGBM-PSO the highest at 0.0386.

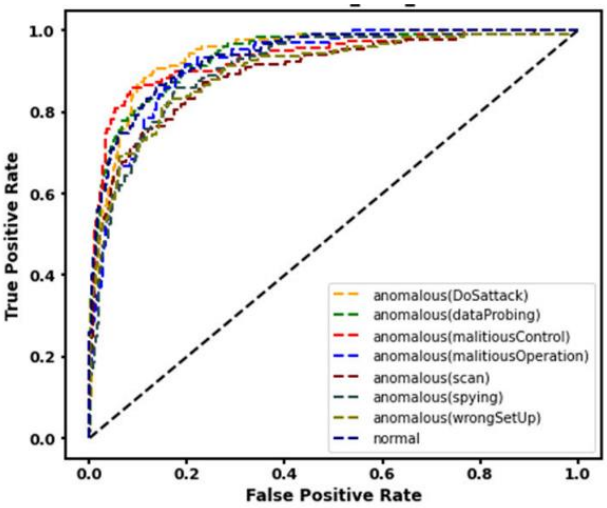
Figure 7 is divided into four sections that show the ROC scores of the DT, LGBM-PSO, XGB, and LGBM classifiers as graphs. Here, multiclass classification using logistic regression is shown in Figure 7(a). Figure 7(b) illustrates the multiclass classification using the LGBM-PSO approach. Figure 7(c), displays the multiclass categorization using XGB. Whereas the multiclass classification using LGBM is shown in Figure 7(d). Seven assault classes and one normal class are predicted by these models. The curves in the aforementioned figures extend past the AUC limit. Other network traffic patterns include "Dos attack," "data probing," "malicious control," "malicious operation," "scan," "spying," and "wrong Setup" attack behaviours in addition to "normal" behaviour. When it comes to accuracy and error measures, XGB and LGBM are the best-performing models. Additionally, the LGBM classifier's TPR and FPR are rather acceptable. The LGBM will also be contrasted with a few additional gradient-boosting ensemble techniques for strong validation.


True Positive Rate
False Positive Rate
anomalous(DoSattack)
anomalous(dataProbing)
anomalous(malitiousControl)
anomalous(malitiousOperation)
anomalous(scan)
anomalous(spying)
anomalous(wrongSetUp)
normal

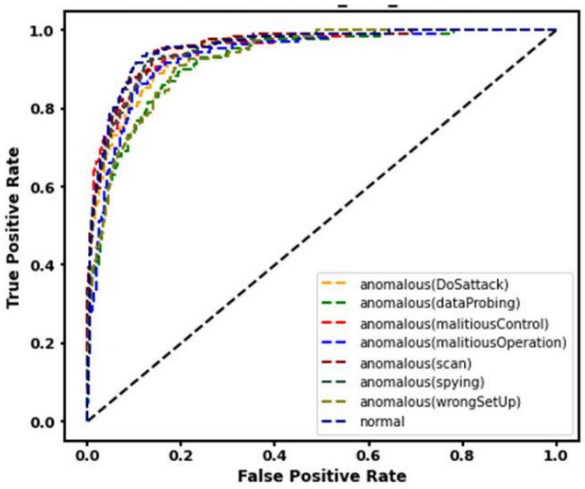
(a)


True Positive Rate
False Positive Rate
anomalous(DoSattack)
anomalous(dataProbing)
anomalous(malitiousControl)
anomalous(malitiousOperation)
anomalous(scan)
anomalous(spying)
anomalous(wrongSetUp)
normal

(b)


True Positive Rate
False Positive Rate
anomalous(DoSattack)
anomalous(dataProbing)
anomalous(malitiousControl)
anomalous(malitiousOperation)
anomalous(scan)
anomalous(spying)
anomalous(wrongSetUp)
normal

(c)



(d)

Figure 7. Classification Models' Graphical Representations of TPR and FPR: (a) Using logistic regression for multiclass categorization; (b) Random Forest categorization with several classes; (c) Multiclass classification with XGBoost; (d) Several classes are classified using Light Gradient Boost Machine.

4.3 Classification Accuracy

To create the best possible prediction model, the most crucial factors to take into account are the test and train accuracy scores. Utilizing n samples for the training set and n-1 samples for the testing set, using the "DS2OS" dataset, the classification accuracy score was determined.

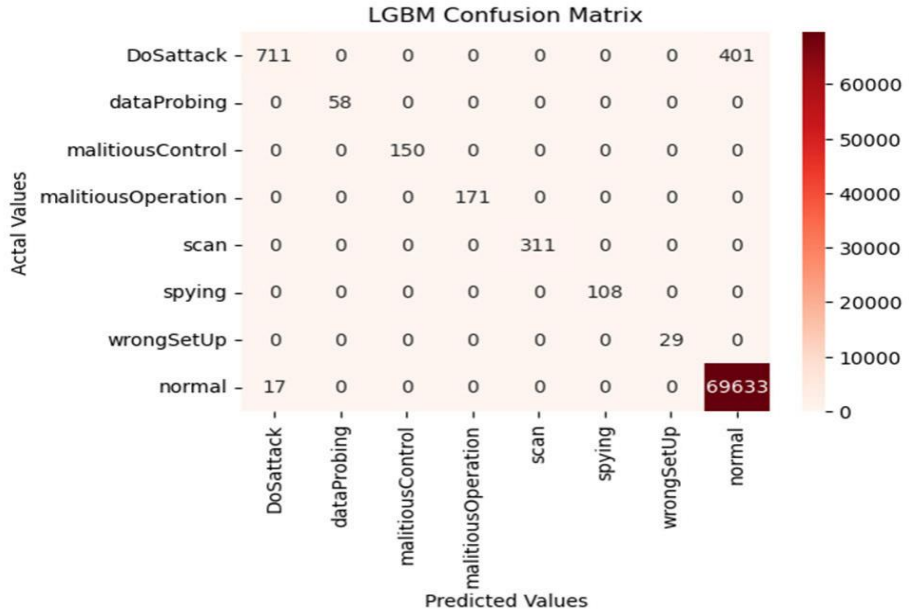


Figure 8. Confusion matrix for LGBM-PSO.

A confusion matrix can provide a clearer illustration of the threat detection rate classification. The confusion matrix, which is used to evaluate the classification's accuracy, is shown in Figure 8. The total number of observations has been divided between the number of values that were discovered and those that were predicted. It displays the locations of the suggested model's faults.

Table 5 displays the test and train accuracy values for the DT, LGBM-PSO, XGB, and LGBM classifiers. You can adjust the train data ratio to get the best outcomes. The train and test data are often assumed to be 80% and 20%, respectively. Here, XGB and LGBM have good values of 99.99% for both train and test accuracy. Consequently, With LGBM, favourable train and test accuracy values of 99.92% are achieved.

Table 5: Classification accuracy score (in percent).

Algorithm	DT	XGB	LGBM	LGBM-PSO
Train	96.35	98.99	99.12	99.89
Test	96.18	98.99	99.12	99.79

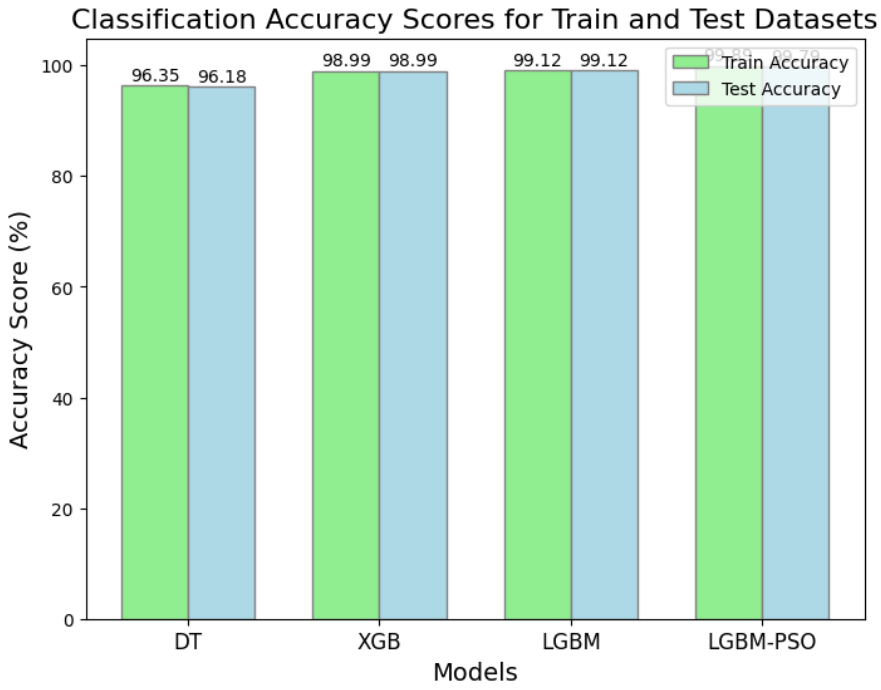


Figure 9. Classification Accuracy Scores for Train and Test Datasets across Different Models

Four machine learning models' categorization accuracy ratings for training and testing datasets are shown in Figure 9: DT, XGB, LGBM, and LGBM-PSO. The graph shows that the DT model has the lowest accuracy for both training and testing, with scores of 96.35% and 96.18%, respectively. The XGB and LGBM models demonstrate high and consistent accuracy, with both achieving 98.99% for training and 99.12% for testing. The proposed LGBM-PSO model demonstrates the best fitting, achieving a very high training accuracy of 99.89% and testing accuracy of 99.79%, the advanced models are in terms of high accuracy that is obtained in both training as well as in test runs of the models whereby LGBM-PSO has proven to be the best.

4.4 Error-Rate Performance

The DT, LGBM-PSO, XGB, and LGBM error-rate assessments are summarized in Table 6. The disparities between expected and actual values that can be expressed as mean absolute error (MAE), The mean squared error (MSE) is the average of the squared differences between the predicted and actual values, this is the absolute differences between the expected and actual values divided by the root mean square error (RMSE), this represents the MSE's average value. These characteristics form the basis of the error-rate analysis; a lower error rate is seen as improving prediction accuracy. The model, which gives the least error value, is most suitable to the data. In this case, it shows that the minimum and equal error rates were obtained for both the XGB and LGBM classifiers.

Table 6: Error rate.

Algorithm	DT	XGB	LGBM	LGBM-PSO
Train	96.35	98.99	99.12	99.89
Test	96.18	98.99	99.12	99.79

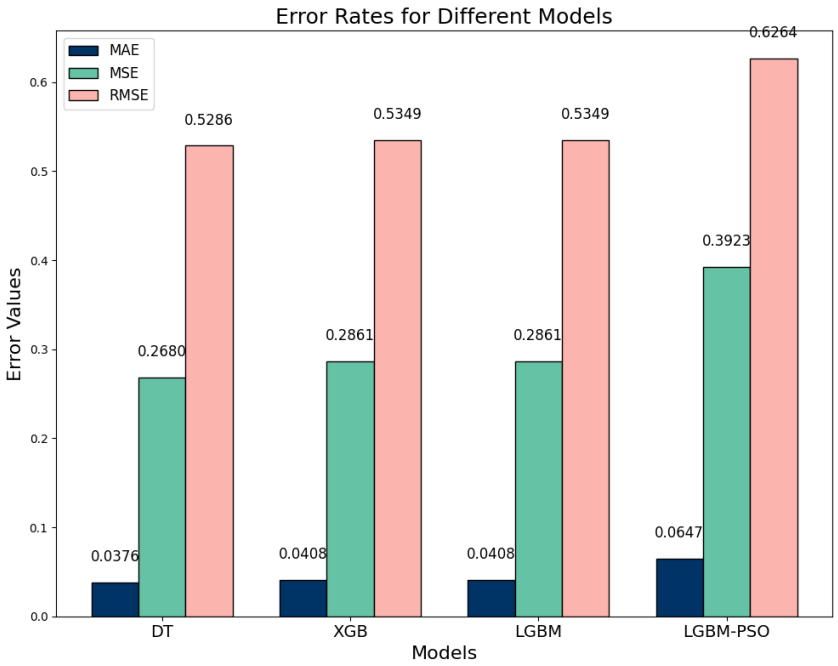


Figure 10. Error Rates (MAE, MSE, RMSE) for Different Models

Figure 10 illustrates the MAE, MSE, and RMSE for four machine-learning models: DT, XGB, LGBM, and LGBM-PSO are the four involved models. The Ma examination shows that the Decision Tree has the least MAE at 0.0376 but slightly higher MSE and RMSE are 0.2680 and 0. 5286, respectively. It can be seen that the XGB and LGBM have the same error metrics errors and both have MAE of 0.0408, MSE of 0.2861, and the RMSE of 0.5349. Speaking of the errors, it should be noted that the LGBM-PSO model has the highest values of the MAE, equal to 0.0647 is the average value with an MSE of 0.3923 and an RMSE of 0.6264. The graph above effectively presents the errors thus we see that while the DT model has the lowest MAE, it does worst in MSE and RMSE. On the other hand, the LGBM-PSO even has the highest error metrics despite all the optimization work that has been done.

4.5 Run Time Performance

Efficient speed or run-time performance is another crucial criterion in choosing a classifier for model development. The intrusion detection system presented in this research achieves superior run-time performance when compared to comparable works of literature. LGBM's accuracy value is likewise excellent. But occasionally, alternative algorithms exhibit 100% accuracy, which might result in the overfitting problem. "DoSattack" (DoS), "wrongSetUp" "data probing" (Probe), "malitiousOperation" (MO), "malitiousControl" (MC), "spying" (Spy), "normal" "scan," (WS), and one class are among the seven malicious classes found in the dataset under examination. The main benefit of the suggested approach is that it sheds light on how long the suggested IDS would take to compute when the classification and prediction method is applied. The LGBM-PSO model executes a method in 6.465 seconds.

4.6 Threat Prediction and Detection Rate

The actual findings may be subtracted from the anticipated (P) results to get the threat detection rate for each class. The actual detection rate (AD) could not match the expected results. Several data samples were chosen to train and test the dataset. One regular class and seven threat classes in total. The percentage of actual detection (AD) and the average threat prediction (P) findings for each class are displayed in Table 7. The DT, LGBM-PSO, XGB, and LGBM algorithms were used to calculate the accuracy, f1-score, recall and support for each class. The detection rates of all attacks determined by LGBM and XGB are nearly identical.

Table 7: Detection rate (in percent).

	DT		XGB		LGBM		LGBM-PSO	
	P%	D%	P%	D%	P%	D%	P%	D%
Dos	99%	68%	99%	89%	100%	97%	100%	98%
probe	99%	89%	100%	99%	100%	100%	100%	100%
MC	100%	100%	100%	99%	100%	100%	100%	100%
MO	99%	96%	99%	97%	99%	100%	99%	100%
Scan	99%	98%	99%	98%	99%	100%	99%	100%
Spy	100%	100%	100%	100%	100%	100%	100%	100%
WS	100%	100%	100%	100%	100%	100%	100%	100%
Normal	100%	99%	100%	99%	100%	99%	100%	99%

The detection rates attained by XGB and LGBM range from 99% to 100%. It is 99% in the majority of instances. In certain cases, DT has not been very successful in identifying dangers. But LGBM-PSO has outperformed DT thus far. XGB and LGBM perform better overall than the others. For XGB and LGBM, Both the true negative and true positive rates are elevated. These algorithms also have an extremely low false alarm rate (FAR). A threat may increase in value if it is present but not recognised or anticipated.

4.7 Overall Performance of Proposed IDS

The efficiency of the recommended IDS is shown in Table 8 concerning precision, accuracy, F1 score, recall, and support. A total of 357941 samples were gathered, and 71589 of them were chosen to be used in the training and testing of the proposed model. In more than 99% of cases, the accuracy is mediocre.

Table 8: Overall performance of proposed IDS.

	DoS	**Probe**	**MC**	**MO**	**Scan**	**Spy**	**WS**	**Normal**	**Accuracy**	**Average**
Precision	98.0	100.0	100.0	100.0	100.0	100.0	100.0	99.0		
Recall	64.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0		
F1-score	77.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.0	99.0
Support	1112	58	150	171	311	108	29	69650	71589	71589

The f1-score, recall, and accuracy by class are shown in Table 9. The appropriately classified attack samples are what matters in this case. Recall is the proportion of attack samples among all attack samples that have been accurately identified as malicious. The matching f1-score, recall, and accuracy rate for the DoS assault are 64, 77, and 98. Nonetheless, for typical class and other assaults, the f1-score, recall, precision, and f1-score are all 100. Figure 12 shows the obliteration of intrusion detection rate using f1-score, recall, and accuracy. The threat detection rate classification is displayed here as a percentage. Encouragement indicates the quantity of samples that genuinely corroborate the finding.

5. Conclusion

In this paper, we examined how suitable advanced machine-learning techniques are for enhancing IDS in Internet of Things environments. Our research demonstrated that incorporating ensemble methods, the LGBM-PSO significantly improves the accuracy and reliability of IDS models compared to traditional approaches. The LGBM-PSO model achieved exceptional performance with a ROC-AUC score of 99.98%, outperforming other algorithms like DT, XGBoost, and LGBM in terms of both detection accuracy and error metrics. Our extensive evaluation

using the "DS2OS" dataset revealed that the LGBM-PSO model excels in identifying network traffic anomalies and maintaining a low false positive rate while demonstrating high true positive rates across various attack scenarios. These findings support the value of using Capital-Ma cutting-edge machine learning approaches to address IoT network security concerns, and hence, IDS setting out a solid groundwork for future innovations. Based on the work presented in this book, multiple directions for further research on IDS technologies for IoT are presented in the following sections to address novel issues in IoT networks. Another line of research might be dynamic IDS systems that can improve their performance through updates according to the new threat information and attack characteristics. Researchers can look at utilizing fresh optimization methods and compounded models that incorporate various machine learning approaches to enhance detection efficiency as well as minimize specific false alarms.

Funding: "This research received no external funding"

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

References

[1] Schwab, K., 2017. The fourth industrial revolution. Crown Currency.

[2] Aliero, M.S., Qureshi, K.N., Pasha, M.F. and Jeon, G., 2021. Smart Home Energy Management Systems in Internet of Things networks for green cities demands and services. Environmental Technology & Innovation, 22, p.101443.

[3] Zeadally, S. and Bello, O., 2021. Harnessing the power of Internet of Things based connectivity to improve healthcare. Internet of Things, 14, p.100074.

[4] Andreev, S., Galinina, O., Pyattaev, A., Gerasimenko, M., Tirronen, T., Torsner, J., Sachs, J., Dohler, M. and Koucheryavy, Y., 2015. Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap. IEEE Communications Magazine, 53(9), pp.32-40.

[5] Praveen, G. Pandian, D. F., C. "IntelliCare: Integrating IoT and Machine Learning for Remote Patient Monitoring in Healthcare: A Comprehensive Framework," Journal of Journal of Cognitive Human-Computer Interaction, vol. 7, no. 2, pp. 50-59, 2024. DOI: https://doi.org/10.54216/JCHCI.070205

[6] Ali, M.S., Vecchio, M., Pincheira, M., Dolui, K., Antonelli, F. and Rehmani, M.H., 2018. Applications of blockchains in the Internet of Things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 21(2), pp.1676-1717.

[7] Khraisat, A. and Alazab, A., 2021. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4, pp.1-27.

[8] Choi, I., Lee, J., Kwon, T., Kim, K., Choi, Y. and Song, J., 2021, August. An easy-to-use framework to build and operate AI-based intrusion detection for in-situ monitoring. In 2021 16th Asia Joint Conference on Information Security (AsiaJCIS) (pp. 1-8). IEEE.

[9] Aziz, A. Mirzaliev, S. Maqsudjon, Y. "Enhancing Malware Detection in Cybersecurity through Optimized Machine Learning Technique," Journal of International Journal of Advances in Applied Computational Intelligence, vol. 4, no. 2, pp. 26-32, 2023. DOI: https://doi.org/10.54216/IJAACI.040203

[10] panthi, V. Kumar, A. "Enhancing Healthcare Monitoring through the Integration of IoT Networks and Machine Learning," Journal of International Journal of Wireless and Ad Hoc Communication, vol. 7, no. 1, pp. 28-39, 2023. DOI: https://doi.org/10.54216/IJWAC.070103

[11] Osman, M., He, J., Mokbal, F.M.M., Zhu, N. and Qureshi, S., 2021. ML-LGBM: A machine learning model based on light gradient boosting machine for the detection of version number attacks in RPL-based networks. IEEE Access, 9, pp.83654-83665.

[12] Okey, O.D., Maidin, S.S., Adasme, P., Lopes Rosa, R., Saadi, M., Carrillo Melgarejo, D. and Zegarra Rodríguez, D., 2022. BoostedEnML: Efficient technique for detecting cyberattacks in IoT systems using boosted ensemble machine learning. Sensors, 22(19), p.7409.

[13] Medjek, F., Tandjaoui, D., Djedjig, N. and Romdhani, I., 2021. Fault-tolerant AI-driven intrusion detection system for the internet of things. International Journal of Critical Infrastructure Protection, 34, p.100436.

[14] Otoum, S., Kantarci, B. and Mouftah, H., 2021. A comparative study of AI-based intrusion detection techniques in critical infrastructures. ACM Transactions on Internet Technology (TOIT), 21(4), pp.1-22.

[15] Schmitt, M., 2023. Securing the Digital World: Protecting smart infrastructures and digital industries with Artificial Intelligence (AI)-enabled malware and intrusion detection. Journal of Industrial Information Integration, 36, p.100520.

[16] Abed, A.K. and Anupam, A., 2023. Review of security issues in Internet of Things and artificial intelligence-driven solutions. Security and Privacy, 6(3), p. e285.

[17] Alsarhan, A., Alauthman, M., Alshdaifat, E.A., Al-Ghuwairi, A.R. and Al-Dubai, A., 2023. Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks. Journal of Ambient Intelligence and Humanized Computing, 14(5), pp.6113-6122.
[18] Kornaros, G., 2022. Hardware-assisted machine learning in resource-constrained IoT environments for security: review and future prospective. IEEE Access, 10, pp.58603-58622.
[19] Sarker, I.H., Furhad, M.H. and Nowrozy, R., 2021. Ai-driven cybersecurity: an overview, security intelligence modeling and research directions. SN Computer Science, 2(3), p.173.
[20] Thomas, L. and Bhat, S., 2021. Machine learning and deep learning techniques for IoT-based intrusion detection systems: A literature review. International Journal of Management, Technology and Social Sciences (IJMTS), 6(2), pp.296-314.
[21] Ge, M., Syed, N.F., Fu, X., Baig, Z. and Robles-Kelly, A., 2021. Towards a deep learning-driven intrusion detection approach for Internet of Things. Computer Networks, 186, p.107784.
[22] Thamilarasu, G. and Chawla, S., 2019. Towards deep-learning-driven intrusion detection for the internet of things. Sensors, 19(9), p.1977.
[23] Naithani, K., 2019. AI-based Intrusion Detection System for Internet of Things (IoT) Networks. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 10(2), pp.1095-1100.
[24] Banaamah, A.M. and Ahmad, I., 2022. Intrusion detection in IoT using deep learning. Sensors, 22(21), p.8417.
[25] Jayalaxmi, P.L.S., Saha, R., Kumar, G., Conti, M. and Kim, T.H., 2022. Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey. IEEE Access, 10, pp.121173-121192.
[26] Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W. and Wahab, A., 2020. A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. Electronics, 9(7), p.1177.
[27] Alsoufi, M.A., Razak, S., Siraj, M.M., Nafea, I., Ghaleb, F.A., Saeed, F. and Nasser, M., 2021. Anomaly-based intrusion detection systems in IoT using deep learning: A systematic literature review. Applied Sciences, 11(18), p.8383.
[28] (2018). DS2OS Traffic Traces, IoT Traffic Traces Gathered in the DS2OS IoT Environment. Accessed: Jul. 28, 2022. [Online]. Available: https://www.kaggle.com/francoisxa/ds2ostraffictraces
[29] Y. K. Anupama, S. Amutha, and D. R. R. Babu, ‘‘Exploring efficient preprocessing techniques for breast cancer diagnosis,’’ in Futuristic Communication and Network Technologies (Lecture Notes in Electrical Engineering), vol. 792. Singapore: Springer, 2021, pp. 855–864.
[30] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, ‘‘Data preprocessing for supervised learning,’’ Int. J. Comput. Sci., vol. 1, no. 2, pp. 111–117, Jan. 2006.
[31] T. Rawat and V. Khemchandani, ‘‘Feature engineering (FE) tools and techniques for better classification performance,’’ Int. J. Innov. Eng. Technol., vol. 8, no. 2, pp. 169–179, 2017.
[32] S. Ullah, J. Ahmad, M. A. Khan, E. H. Alkhammash, M. Hadjouni, Y. Y. Ghadi, F. Saeed, and N. Pitropakis, ‘‘A new intrusion detection system for the Internet of Things via deep convolutional neural network and feature engineering,’’ Sensors, vol. 22, no. 10, p. 3607, May 2022.
[33] S. Seth, G. Singh, and K. K. Chahal, ‘‘A novel time efficient learning-based approach for smart intrusion detection system,’’ J. Big Data, vol. 8, no. 1, pp. 1–28, Dec. 2021.
[34] D. Jin, Y. Lu, J. Qin, Z. Cheng, and Z. Mao, ‘‘SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism,’’ Comput. Secur. vol. 97, pp. 1–17, Oct. 2020.
[35] Md. K. Islam, P. Hridi, Md. S. Hossain, and H. S. Narman, ‘‘Network anomaly detection using LightGBM: A gradient boosting classifier,’’ in Proc. 30th Int. Telecommun. Netw. Appl. Conf. (ITNAC), Melbourne, VIC, Australia, Nov. 2020, pp. 1–7.
[36] D. Rani, N. S. Gill, P. Gulia, and J. M. Chatterjee, ‘‘an ensemble-based multiclass classifier for intrusion detection using Internet of Things,’’ Comput. Intell. Neurosis. vol. 2022, pp. 1–16, May 2022.
[37] De Almeida, B.S.G. and Leite, V.C., 2019. Particle swarm optimization: A powerful technique for solving engineering problems. Swarm intelligence-recent advances, new perspectives and applications, pp.31-51.