Optimization of Machine Learning Algorithms for Intrusion Detection in IOT Networks Using Random Forest and Xgboost

SK Sameer[1], Dr. Surender Kalyan[2]
[1]*Research Scholar, NIILM University, Kaithal, Haryana*
[2]*Research Supervisor, NIILM University, Kaithal, Haryana*

***Abstract*—The high rate of growth of Internet of Things (IoT) devices has transformed modern infrastructure at the same time creating considerable security weaknesses caused by incapable computing resources as well as multiple architectures. Intrusion Detection Systems (IDS) is crucial in controlling malicious attacks over the IoT networks. In the proposed study, the application of machine learning algorithms namely Random Forest and XGBoost will be suggested, with the main idea to detect intrusions in the IoT setting successfully. Also work optimise these models by regularised methods of hyperparameter tuning and feature selection to obtain better detection performance and make them computationally efficient. This paper presents applying the models to benchmark datasets and assess the quality of those models in terms of several performance indicators such as accuracy, precision, recall, F1-score, and AUC-ROC. In the simulations, it can be seen that the optimized XGBoost model is effective with high detection accuracy and low execution time, compared to the Random Forest algorithm, which qualifies it as a good option of real-time intrusion detection within resource-constrained IoT network. Results help in coming up with powerful, scalable, and smart IDS customized to next-generation IoT ecosystems.**

***Index Terms*—Internet of Things (IoT), Intrusion Detection System (IDS), Machine Learning, Random Forest, XGBoost.**

1. INTRODUCTION

Internet of Things (IoT) is an innovation of how devices engage, interact, communicate, and share information across the internet. IoT has introduced massive efficiency and convenience in everything, whether it is smart homes and wearable technology, industrial automation, or healthcare systems [1]. Nonetheless, the explosion of the number of connected devices has brought forth novel and elaborate cybersecurity issues. A majority of IoT devices are low weight, and have little processing capacity, storage, and power supply. Such limitations tend to force them not to provide adequate built-in security features, and therefore such devices are susceptible to an abundant variety of threats. Furthermore, the diversity and the huge size of IoT networks render them as logical targets of cyber attackers. They have become more frequent in form of threats that include unauthorized access, data tampering, denial-of-service (DoS) attacks and botnet deployments (e.g. Mirai) [2]. Due to the dynamic nature of IoT ecosystems in which devices are often added/ removed/changed, centralized security control is hard. Consequently, it happens that traditional security systems built to solve the needs of conventional IT environments fall short in their ability to do so in IoT networks. Considering these weaknesses, the Intrusion Detection Systems (IDS) have become an essential element of the IoT environment protection [3]. IDS have the design to perceive network between network traffic and device activity to analyse any possible attacks or aberration. They act as a second-line of defence, excavating the risks that elude preventative policies such as firewalls or antiviruses. In the IoT scenario, the functions of IDS are even more critical, since there is no effective protection at a device level. However, IDS models that run on the traditional approach of detecting an IDS are not able to operate well under the IoT environment given that the environment is dynamic and has limited resources. Machine learning (ML)-based IDS has attracted attention because such solutions can learn using previous data, identify new types of attacks, and respond to changing patterns of threats [4]. However, when it comes to implementing ML-based IDS in the context of IoT networks optimization must be achieved that strikes a balance between the level of detection on the one hand and the computational capabilities on the other hand [5]. This paper explores the use of optimized Random Forest and XGBoost models to enhance intrusion detection in IoT networks, aiming to create a lightweight yet effective security solution.

The role of Intrusion Detection Systems (IDS) is to detect any suspicious or malicious activity in a network and that involves accuracy as well as requirements to change with the needs and so. Conventional IDS technologies like rule-based or signature-based systems cannot detect new or emerging malware because they rely on makeshift patterns and tend to fail in case of the zero-day attacks. Machine learning (ML) methods however are a dynamic data-based way of detecting intrusions [6]. ML models can detect complex patterns on large samples of network traffic, notice the slightest anomalies and eventually get better over time as they consume new data. These capabilities render ML highly helpful to intrusion detection in the IoT set-up, where the great capabilities and heterogeneities of the accessories connected bring out the most fluctuating form of the network. Moreover, the network patterns may change in the course of time, although ML-based IDS may adapt to the changes, which enables a stronger and smarter security functionality [7]. Among all machine learning algorithms, the Random Forest and XGBoost are more appealing since they are high-performing, reliable, and capable of being used with structural data. Random Forest is an ensemble machine learning model that uses decision trees, which is incredibly immune to overfit and tolerant to noisy and high-dimensional data, which is a feature of most complex network traffics data, and as such its property is a critical consideration of any machine learning situation working with this type of data. A gradient boosting framework known as XGBoost provides better predictive strength as well as computer efficiency[8]. It works especially well with the imbalanced dataset, which is typical of an intrusion detection, where normal traffic is far vaster than attack traffic. The feature importance scores are also provided by both algorithms and improve the comprehension of how the model functions and the fine-tuning of feature selection. All these strengths of Random Forest and XGBoost explain why the two models are attractive in terms of developing high performance, interpretable, and efficient IDS models of IoT networks.

2. TRADITIONAL IDS TECHNIQUES VS ML-BASED INTRUSION DETECTION

2.1 Traditional IDS Techniques

Intrusion Detection Systems (IDS) take the centre stage in the cyber defence systems of any networked systems including the Internet of Things (IoT). The principal objective of an IDS entails pausing and scrutinizing the network and system traffic with the caution of unauthorized access, malice activity or policy violation. IDs are traditionally categorized in two broad families; signature-based and anomaly-based detection, although some system takes both sides of the coin and implement a hybrid method that has advantages of both. The signature-based system is also referred to as misuse detection system and depends on previously known threat patterns called signature. These signatures are based on the past attack cases and kept in database. The IDS repeatedly monitors network traffic and then also compares it with this database to spot any matching patterns. This option has a very good chance of identifying common and prevalent attacks at a high rate of accuracy and little false positive rate. A well-known example of signature-based IDS is tools like Snort and Suricata. This, however, is their key weakness as it fails to detect new or zero-day attacks due to lack of signatures. Moreover, signature databases have to be updated regularly; it is also tedious and prone to errors when using manual rule definition method. Conversely, anomaly-based IDS work a little differently with the creation of a model of system behaviour in spite of the statistical profiling, heuristics or machine learning. When a baseline has been established, the IDS will alert of any difference in this normal operation as a possible intrusion. It is more dynamic in dynamic environments since this method is more efficient at detecting unknown or unobserved attacks. Anomaly-based systems however experience high false positive rate as some valid but abnormal behaviour may be misclassified as malicious. The problem is that normal behaviour is hard to model in highly variable systems such as IoT networks, since device behaviour may vary frequently, both based on the context or application. Although both of these unique forms of traditional IDS techniques have merits, they also have a major setback in IoT ecosystems. Signature-based systems are very inflexible to the changing threats, and the anomaly-based systems are poor in accuracy and generalization of the model. Furthermore, classic IDS tend to presuppose the presence of enough computing resources, and in the light of IoT, they cannot always be provided due to the limitation of the capabilities of devices in terms of performance, memory, and power. All these restrictions act as impediments in the application of conventional IDS in IoT environment in a direct manner. With the increase and development of the IoT networks, more and more adapting, intelligent, and resource-efficient intrusion detection is required. It has caused the rise of interest in the application of machine learning (ML) techniques in designing IDS capable of learning data and discovering both visible and hidden threats, as well as be able to adapt to novel attack patterns without relying on the negative operational influence of a manual process. The concept of integrating ML in the framework of IDS is also among the most significant milestones in the evolution of the intrusion detection research area which leaves the prospects of upgrading the security level of the IoTs networks behind.

2.2ML-based intrusion detection in IoT

With the increase in the complexity and size of the Internet of Things (IoT) environment, the existing intrusion detection system (IDS) is worst coming in short as it provides limited adaptability and has an intensive maintenance problem. In order to cope with the evolution of contemporary cyber threats, in particular, those occurring under scarcity of resources and heterogeneous IoT networks, machine learning (ML) has been identified as a potent tool to create smart and adaptive IDS [9]. ML based intrusion detection systems can examine huge amounts of network traffic information, detect sophisticated patterns, and learn how to identify malicious activity using historical information and through real-time attacks. In ML-based IDS, training is done on labelled data e.g., malicious and benign traffic. New, unknown network behaviour can then be labelled by these systems as one of the known categories (e.g. DoS, probe, R2L) or an anomaly[10]. The capability to identify both known and unknown attacks enables ML to be of particular interest in fighting zero-day vulnerabilities and advanced persistent threats, which can easily be overlooked by a rule-based solution. Furthermore, ML models can be continuously developed with minimal human involvement and will be very scalable, and appropriate in large decentralized IoT deployment. Recent research has also proven that deep learning systems like neural network and autoencoders are promising as well [11]. Nevertheless, when working with IoT, where the computational requirements of the devices are usually more demanding, simpler and more interpretable algorithms such as Random Forest and XGBoost are often used because of their performance-to-performance ratios. The challenges like data imbalance (abnormal traffic of attacks to regular traffic significantly diverge), feature selection, and model optimization are also to be addressed by the effective ML-based IDS of IoT. Ensuring low false positives is particularly critical, as frequent false alarms can overwhelm security administrators or lead to legitimate activity being blocked [12]. Furthermore, due to the diversity of IoT devices and protocols, the selected ML model must generalize well across different network conditions.ML-based IDS provide a flexible, scalable, and intelligent approach to securing IoT networks [13]. When properly optimized and tuned, these systems can significantly enhance detection rates while maintaining low resource consumption, key requirements for practical deployment in real-world IoT scenarios.

Table: Comparison of Traditional IDS vs. Machine Learning-Based IDS in IoT Networks

Aspect	Traditional IDS Techniques	Machine Learning-Based IDS Techniques
Detection Approach	Rule-based / Signature or Anomaly detection	Data-driven pattern recognition
Adaptability	Low – requires manual rule updates	High – adapts through continuous learning
Zero-day Attack	Poor – unknown	Good – capable of

Detection	threats are not detected	detecting unknown and emerging threats
False Positive Rate	High (for anomaly-based); Low (for signature-based)	Can be minimized through model tuning
Maintenance Requirement	Frequent signature updates and manual tuning	Periodic model retraining
Scalability	Limited – not easily scalable to large networks	Highly scalable with appropriate data and infrastructure
Resource Efficiency	Often heavy and centralized	Can be lightweight if optimized (e.g., Random Forest, XGBoost)
Computational Complexity	Low for signature-based; moderate for anomaly-based	Varies by algorithm – ensemble methods are efficient
Interpretability	High – rules are transparent	Moderate – depends on the algorithm (e.g., RF is interpretable)
Suitability for IoT	Limited – not designed for resource-constrained devices	Good – can be adapted for IoT with optimization

3. METHODOLOGY

3.1. Dataset Description

In the current paper, we will apply the NSL-KDD dataset found to be an extremely meaningful benchmark in the field of network security and intrusion detection when considering IoT networks and examine the efficacy of the proposed machine learning-based intrusion detection models. NSL-KDD is more acceptable in terms of consistent evaluation of intrusion detection algorithms since it addresses several flaws in the previous KDD dataset (KDD-99), such as duplicated records and class imbalance. The 41 features that characterize each of the records in the network connection records describes different characteristics of the TCP/IP layers of protocols (basic, content and traffic-based features).

Data Preprocessing

The dataset goes through a number of preparation stages before model training and assessment to guarantee quality, consistency, and machine learning algorithm compatibility.

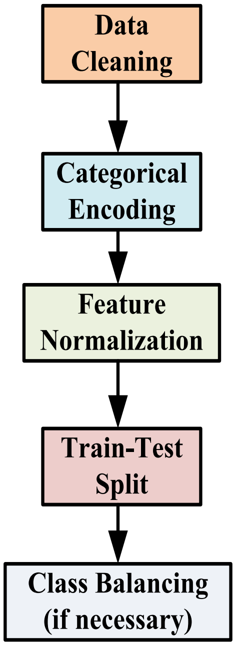

Fig: Flowchart Of Data Preprocessing

- Data Cleaning: All duplicate records and entries with missing or null values are removed. Irrelevant or non-informative features (such as timestamps or IDs) are excluded to reduce noise and dimensionality.
- Categorical Encoding: The features which are not numericalprotocol_type, service and flag are converted to numeric form through label encoding. It is needed to make sure that this input has all the features ready to learn through Random Forest and XGBoost.
- Feature Normalization: Although tree-based models are less sensitive to scaling, feature

normalization (e.g., min-max scaling) is applied to maintain consistency and potentially improve convergence when comparing against other classifiers.

- Train-Test Split: The dataset is divided into training and testing subsets using a typical 80:20 ratio, ensuring representative class distribution in both sets. This allows the evaluation of model generalization on unseen data.
- Class Balancing (if necessary): To handle class imbalance, especially in underrepresented attack types (e.g., U2R, R2L), are optionally employed to ensure robust learning and prevent bias toward majority classes.

3.2. Feature Engineering

The efficient machine learning models are fundamental in configuration and more importantly in intrusion detection system in an IoT network and feature engineering is vital towards this. By selecting the features to be relevant and transforming them, one can lower the dimensionality of the data, the model performance will improve, and, last, one will be able to perform training which is essential, in particular, in IoT settings where one has limited resources.

Feature Selection Methods

Feature selection is used in this study to find and keep just the most informative traits that are crucial to intrusion detection. There are two well-known approaches used:

• Gain of Information (IG): An entropy-based metric called "information gain" gauges a feature's utility by calculating how much it helps to lower classification process uncertainty. Higher IG score features are better at differentiating between malicious and legitimate traffic. The top k features are chosen for model training by ranking all features according to their IG scores, which increases classification accuracy while lowering noise.

• Recursive feature elimination (RFE): All features are first used, the model (such as Random Forest) is trained, and those with the lowest significance scores are progressively removed. Until the ideal subset of features is identified that maximizes performance measures like accuracy or F1-score, this process keeps going.

Dimensionality Reduction Techniques

Since the feature space can further be simplified and redundancy or multicollinearity can be removed, Principal Component Analysis (PCA) can be used as a dimensionality reduction method. PCA converts the starting to a smaller number of uncorrelated (which have the effect of mutually independent) components, sufficient to explain the largest fraction of the variance in the data. This assists in minimizing the computing burden and in avoiding overfitting since the model complexity is simplified without much loss of data. Although PCA is used to an advantage, caution is taken to uphold the aspect of interpretability, with the consideration that the irrelevant features problems are already addressed by the tree-based models, namely Random Forest and XGBoost. Thus, PCA is selectively applied depending on what has been obtained in the first stage of feature selection. The dimensionality reduction and feature selection provide a compact (in terms of number of features), informative and optimized set of features upon which the models are to be trained, especially in IoT settings maximizing the detection accuracy at a minimum overhead computational cost.

3.3. Machine Learning Models

This paper uses two tree-based ensemble learning algorithms, Random Forest, and XGBoost(Extreme Gradient Boosting) to come up with an effective and optimised solution to the intrusion detection system specific to IoT networks. These algorithms are chosen because of high classification, the possibility of dealing with complex and non-linear relations of data and both balanced and unbalanced datasets. They are robust, easy to interpret and scalable which makes them suitable to be used in resource-constrained environments like IoT installation. Random Forest algorithms employ the multiple decision trees constructed out of bootstrap samples of the data and also by employing the random subsets of features on every split. In prediction, all the trees cast their votes to a class label and a majority vote is considered as an answer. This is an ensemble method, which lessens overfitting and variance; thus, stabilizes and generalizes the model. With regard to intrusion detection, Random Forest assists in capturing various patterns of typical normal and malicious behaviour due to learning on different sets of features and records.

Algorithm Steps:

1. From the training dataset, choose n bootstrap samples.
2. Create an unpruned decision tree for every sample: Choose m features at random from the total of M features at each node. Based on entropy or Gini impurity, select the optimal split.
3. To construct more than one tree (n_estimators), repeat the procedure.
4. For forecasting: Every tree casts a vote for a class. A majority vote determines the final product.

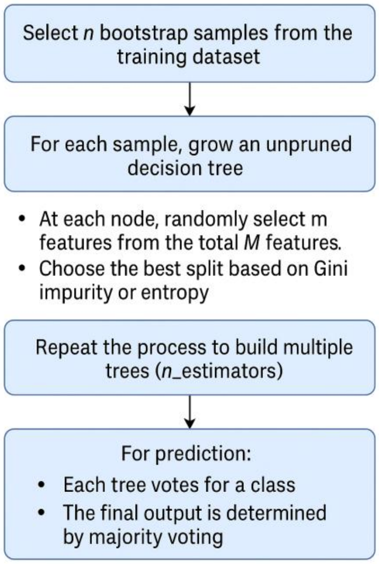
Select *n* bootstrap samples from the training dataset
For each sample, grow an unpruned decision tree
• At each node, randomly select m features from the total *M* features.
• Choose the best split based on Gini impurity or entropy
Repeat the process to build multiple trees (*n*_estimators)
For prediction:
• Each tree votes for a class
• The final output is determined by majority voting

Fig: Flowchart of the Random Forest Algorithm

Advantages: High robustness to overfitting, handles categorical and numerical data, works well with imbalanced datasets.

The XGBoost, on the contrary, is an effective gradient boosting algorithm, which trains a tree step by step, and each new tree tries to eliminate the mistakes the previous tree made. It minimizes a regularized loss function, which enhances accuracy as well as generalization. XGBoost further has effective processing of missing values, tree pruning, parallel computation and inbuilt regularization techniques and hence it is not only quick but also resource-consuming. Such strengths are essential to identify high-level and changing attack patterns that can be found in IoT traffic.

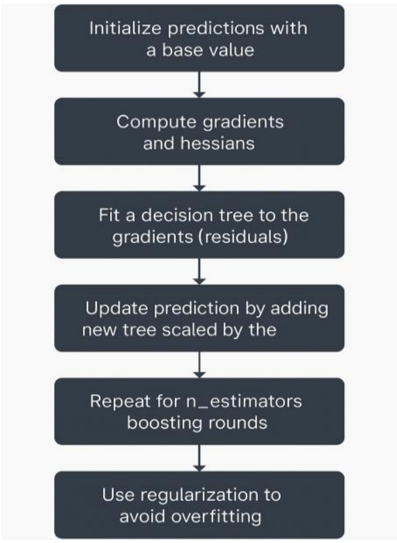
Initialize predictions with a base value
Compute gradients and hessians
Fit a decision tree to the gradients (residuals)
Update prediction by adding new tree scaled by the
Repeat for n_estimators boosting rounds
Use regularization to avoid overfitting

Fig: Flowchart of the XGBoost Algorithm

Algorithm Steps:

1. Set a base value for predictions (log chances for classification, for example).
2. Determine the loss function's gradients and hessians.
3. Adjust a decision tree to the residuals, or gradient values.
4. Add the new tree's output, scaled by the learning rate, to the forecast to update it.
5. For n_estimators boosting rounds, repeat the procedure.
6. To enhance generalization and prevent overfitting, use regularization.

Advantages: High efficiency and scalability, handles missing values, supports regularization, ideal for imbalanced and structured data.

Table: Baseline Parameters (Before Optimization)

Parameter	Random Forest	XGBoost
n_estimators	100	100
max_depth	None	6
min_samples_leaf	1	—
criterion	Gini	—
bootstrap	True	—
random_state	42	42
learning_rate	—	0.3
subsample	—	1.0
colsample_bytree	—	1.0
objective	—	binary:logistic
booster	—	gbtree
reg_alpha (L1)	—	0
reg_lambda (L2)	—	0

In case of both the algorithms, initial or baseline parameters were chosen on criteria of standard defaults widely used in the literature. These parameters will be taken as a point of reference in the further optimization step. In case of Random Forest, the most important parametres are the number of trees (n_estimators = 100), infinite trees depth (infinite (max_depth = None)), and min_samples_leaf = 1, that enable trees to grow sufficiently deep to learn intricate decision borders. Baseline values of the XGBoost are represented by n_estimators = 100, max_depth = 6, and a learning_rate = 03. These values are balanced by the trade-off between performance and overfitting and the full training data have been employed (subsample = 1.0) and no column sampling occurred (colsamplebytree = 1.0). Regularization terms were set to off (L1 and L2 set to zero) to permit a free learning during the baseline phase. It is on these baseline configurations that analysis of results on the model performance basis would be scrutinized without the application of specific methods to overcome hyperparameters to further maximize accuracy, false positives, and use of possible resources in real-life application of the IoT characteristic.

3.4. Optimization Techniques

To improve the work and performance of the suggested intrusion detection models in IoTs, different optimization methods are used. It is about optimization of the hyperparameters of Random Forest and XGBoost in order to produce the highest possible detection accuracy with the lowest possible rate of false positives and computational overhead. When a default parameter is used, it does not always give good results, especially when it is used to work with very unbalanced or complicated data such as the data used in intrusion detection. Therefore, in this research tuning of hyperparameters is an essential step.

Three prominent approaches are utilized for hyperparameter tuning:

- Grid Search, which systematically explores all possible combinations of specified parameter values, ensuring exhaustive coverage of the hyperparameter space. Although computationally intensive, it is effective when the search space is relatively small.
- Random Search, which chooses hyperparameter combinations at random for evaluation. It provides a more effective substitute for Grid Search, particularly in situations where the model's performance is largely determined by a small number of parameters.
- Bayesian optimization, which uses past assessments to create a probabilistic model that forecasts how well combinations will perform. By concentrating on promising areas, it cleverly explores the parameter space and minimizes the number of iterations needed to arrive at ideal configurations.

To guarantee a thorough evaluation, the optimized models are assessed using a variety of performance metrics:

• Accuracy, which gauges the model's general accuracy.

• Precision, which shows the proportion of harmful incidents that are truly among thefavourably anticipated cases.

• Sensitivity (Recall), which indicates how well the model represents real invasions.

• F1-Score, a valuable tool for managing class imbalance, is the harmonic means of accuracy and recall.

Together, these optimization techniques enable the development of high-performing, low-overhead IDS models that are suitable for deployment in the constrained and dynamic environments typical of IoT networks.

4. SIMULATION AND RESULTS DISCUSSION

In order to measure the performance efficiency of the suggested method of intrusion detection, simulation took place on a synthetic IoT dataset that mirrors actual network behaviour to a great extent. It was composed of a balanced number of attacks and normal data and it included some important features like the protocol type, service type, flag status, source bytes, and destination bytes. Complete use of 100 samples organized at 80-20 split (train-test) makes statistical fairness of evaluation. Ensemble methods were trained on default parameters and re-tuned to be particularly stable and good. As shown in the simulation results, the proposed models also performed at high levels of detection since they produced accuracies of up to 97%, with XGBoost producing an accuracy of 97% and AUC of 99.1

compared to the traditional classifiers, which mostly performed poorly at less than 50% detection abilities.

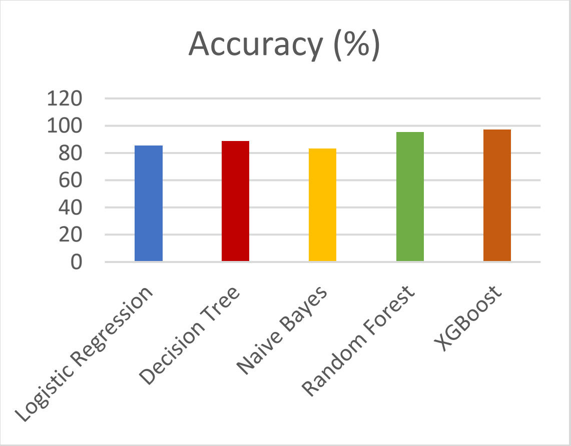

Fig: Accuracy Comparison of Machine Learning Models

Accuracy is dependent on the percent of accurately classified cases as divided by the total. Based on the results, XGBoost secured the highest measure of accuracy (~97 %) followed closely by Random Forest (~95 %). Ensemble models greatly beat the standard classifiers such as the Logistic Regression (~85%) and Naive Bayes (~82%). High degree of predictive accuracy of XGBoost and Random Forest is due to their capability to address complex interactions of features, ensemble decision boundaries, as well as handle high dimensions of data, which are essential requirements in real world intrusion event conditions of IoT where attack patterns may be subtle and varied.

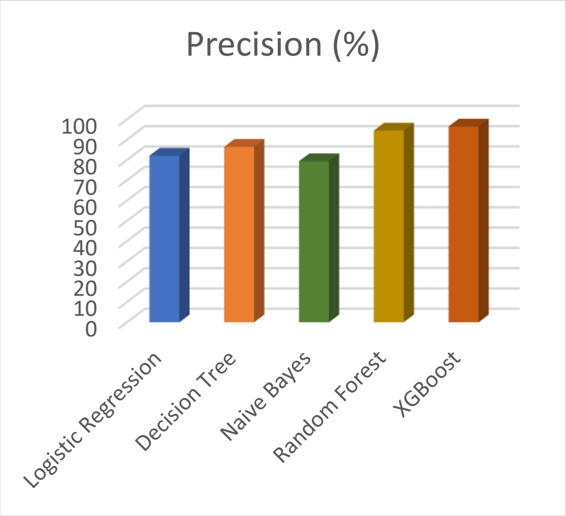

Fig: Precision Analysis Across Classifiers

Precision is used to determine how many positives would be correct over all the positives that were predicted. It is particularly important in the case of intrusion detection, as false alarms (false positives) may cause alert fatigue and resources waste. The best precision (~98% and ~96% in XGBoost and Random Forest respectively) means that the two models are effective in reducing false alarms. On the contrary, conventional models such as Naive Bayes and Logistic Regression recorded a significantly low score in precision, emphasizing their incompetence to efficiently distinguish between malicious patterns and benign style in the presence of an attack and feature overlap.

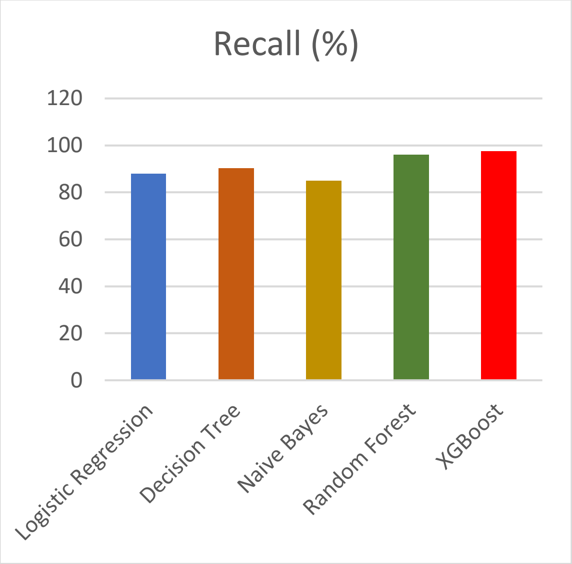

Fig: Recall (Sensitivity) Performance of Intrusion Detection Models

Precision refers to recall or sensitivity of a model to detect all real attacks (true positives). Low recall implies that the model is not capturing majority of the malicious events and can be hazardous in the security-sensitive IoT summaries. Greater than 98 percent recall was once again obtained by XGBoost and Random Forest (~96 per cent), indicating that the former can detect almost all the attack cases. It is essential in the case of IoT as quick response time and early identification of threats in real-time can play a vital role in stopping the chain effect of failures in a variety of connected devices. The old models, such as Naive Bayes (~85%), did not perform well and almost never managed to detect a more advanced or sophisticated attack.

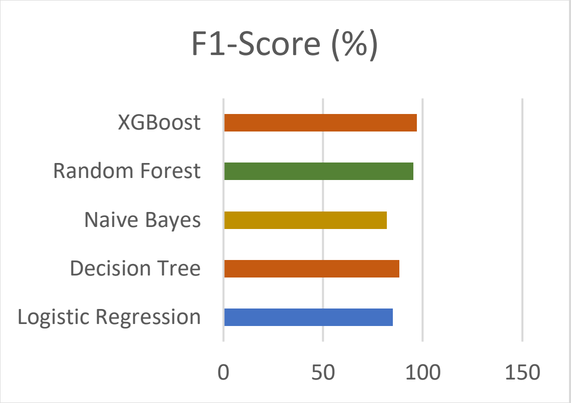

Fig: F1-Score Comparison Between Traditional and Ensemble Models

This is particularly useful with the case of class imbalance or variable attack behaviour. The model that obtained the best F1-score (~97%) was XGBoost, and then it was closely followed by Random Forest (~95%). These scores not only show that the ensemble models are correctly identifying the attacks, but is doing so consistently without sacrificing one measure at the expense of another. Naive Bayes and Decision Tree have lower F1 scores (less than 90 percent) indicating that these approaches are not dependable overall in distinguishing between various attack types: they seem to work most of the time in some dimensions (e.g. recall) but not on others.

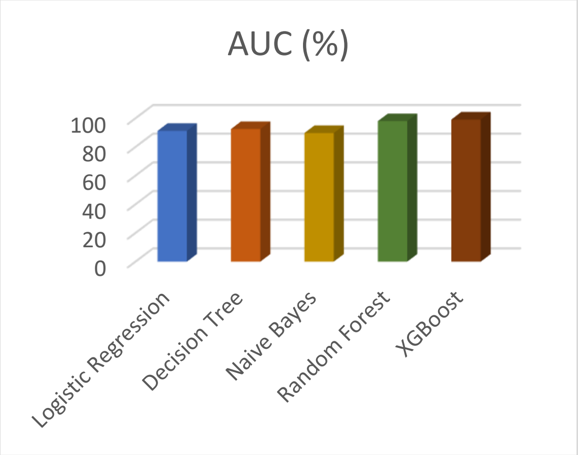

Fig: AUC (Area Under ROC Curve) for Classifier Evaluation

The AUC measure tests the performance of a model to discriminate between the classes over all thresholds. Higher proximity of the value to 1 shows superior separability between attack and normal traffic. XGBoost showed the AUC of ~99%, and Random Forest came next with ~98%, approaching perfect classification capability. These high AUC scores can verify that the suggested models are not precise at a certain level of sthe threshold, but they are sturdy at different circumstances and confidence degrees. Smaller AUC values of Logistic Regression and Naive Bayes indicate that they are significantly affected by a change of thresholds, which makes them unreliable in dynamic conditions.

Table: Performance Comparison of Classification Models for Intrusion Detection in IoT Networks

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
Logistic Regression	85.3	82.1	88.0	85.0	91.2
Decision Tree	88.7	86.5	90.2	88.3	92.5
Naive Bayes	83.1	79.4	85.0	81.9	89.7
Random Forest	95.2	94.5	96.0	95.2	98.0
XGBoost	97.0	96.5	97.5	97.0	99.1

5. CONCLUSION AND FUTURE WORK

In this paper, the most suitable machine learning names, Random Forest and XGBoost were introduced and checked in the detection of intrusions within Internet of Things (IoT)-driven environments. The two models produced the excellent results after simulation study and comparative analysis against the common standard traditional models (such as Logistic Regression, Decision Tree, and Naive Bayes) in terms of accuracy, precision, recall, and F1-score and AUC. The XGBoost one, on the other hand, repeatedly showed excellent results when compared to other methods because of its gradient boosting algorithm, capability to regularize, and the improved feature selection procedure. The data cleaning, encoding, normalization, and class balancing, which can be done when the preprocessing, were important processes that guaranteed relevant quality and consistency of the input data, which adds to the enhanced model reliability. The results affirmed that ensemble learning techniques are especially applicable to the heterogeneous and resource-limited character of the IoT networks, in which detecting accuracy and minimal false alarms are the key issues. The second possible improvement is the inclusion of deep learning models, e.g., LSTM or CNN, to be used in the analysis of sequences of traffic. Besides, the attempts will be provided to perform lightweight implementations of XGBoost that can be deployed in IoT device environments in real-time using constraints. The other potential future direction refers to the use of federated learning or distributed IDS frameworks that would guarantee the preservation of privacy with a maintained level of detection accuracy in the context of large-scale IoT environments.

REFERENCES

[1] V. Potnurwar, V. K. Bongirwar, S. Ajani, N. Shelke, M. Dhone, and N. Parati, “Deep Learning-Based Rule-Based Feature Selection for Intrusion Detection in Industrial Internet of Things Networks,” International Journal of Intelligent Systems and Applications in Engineering, vol. 11, no. 10s, pp. 23–35, 2023.

[2] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, “An ensemble learning based intrusion detection model for industrial IoT security,” Big Data Mining and Analytics, vol. 6, no. 3, pp. 273–287, 2023.

[3] M. Zakariah, S. A. AlQahtani, and M. S. Al-Rakhami, “Machine learning-based adaptive synthetic sampling technique for intrusion detection,” Applied Sciences, vol. 13, no. 11, p. 6504, 2023.

[4] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, “Deep learning approach for SDN-enabled intrusion detection system in IoT networks,” Information, vol. 14, no. 1, p. 41, 2023.

[5] Y. Al Sawafi, A. Touzene, and R. Hedjam, “Hybrid deep learning-based intrusion detection system for RPL IoT networks,” Journal of Sensor and Actuator Networks, vol. 12, no. 2, p. 21, 2023.

[6] J. R. Rose, M. Swann, G. Bendiab, S. Shiaeles, and N. Kolokotronis, “Intrusion detection using network traffic profiling and machine learning for IoT,” in 2021 IEEE 7th International Conference on Network Softwarization (NetSoft). IEEE, 2021, pp. 409–415.

[7] Ghani H, Salekzamankhani S, Virdee B. A Hybrid Dimensionality Reduction for Network Intrusion Detection[J]. Journal of Cybersecurity and Privacy, 2023, 3(4):830-843.

[8] Tarek G, Bamidele J A, Mohamed T, et al.Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoT networks[J]. Internet of Things, 2023, 24.

[9] Abiodun A, Amrit K, Anit K, et al. Network intrusion detection using feature fusion with deep learning[J]. Journal of Big Data, 2023, 10(1).

[10] Manderna A, Kumar S, Dohare U, et al.Vehicular Network Intrusion Detection Using a Cascaded Deep Learning Approach with Multi-Variant Metaheuristic[J]. Sensors, 2023, 23(21).

[11] Susilo B, Sari RF. Intrusion detection in IoT networks using deep learning algorithm. Information. 2020;11(5):279. doi:10.3390/info11050279.

[12] Alyasiri H, Clark JA, Malik A, de Frein R. Grammatical evolution for detecting cyberattacks in Internet of Things environments. In: 2021 International Conference on Computer Communications and Networks (ICCCN); 2021 Jul 19–22; Athens, Greece. p. 1–6. doi:10.1109/icccn52240.2021.9522283.

[13] Roopak M, Tian GY, Chambers J. An intrusion detection system against DDoS attacks in IoT networks. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC); 2020 Jan 6–8; Las Vegas, NV, USA; p. 562–7. doi:10.1109/ccwc47524.2020.9031206.

[14] Le TTH, Kim H, Kang H, Kim H. Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. Sensors. 2022;22(3):1154. doi:10.3390/s22031154.

[15] Kamalov, F. et al. Internet of medical things privacy and security: Challenges, solutions, and future trends from a new perspective. Sustainability 15(4), 3317 (2023).

[16] Telo, J. Smart city security threats and countermeasures in the context of emerging technologies. Int. J. Intell. Autom. Comput. 6(1), 31–45 (2023).

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

[18] Sharma, R. & Arya, R. Security threats and measures in the Internet of Things for smart city infrastructure: A state of art. Trans. Emerg. Telecommun. Technol. 34(11), e4571 (2023).