Efficient Cyber Threat Detection with XGBoost on Compact Network Traffic Datasets

Agha Wafa Abbas
Lecturer, School of Computing, Arden University, Coventry, United Kingdom
Lecturer, School of Computing, IVY College of Management Sciences, Lahore, Pakistan
Emails: awabbas@arden.ac.uk, wafa.abbas.lhr@rootsivy.edu.pk

Muhammad Awais Malik
Deputy Head School of Computing, ICMS
Emails: awaismalik3577@gmail.com

Abstract

The increasing number of Internet of Things (IoT) devices necessitates the use of lightweight intrusion detection systems (IDS) to defend against malware and Distributed Denial of Service attacks. A compact 379.72 KB UNSW-NB15 dataset subset (2,000 records: 1,000 normal, 1,000 attack) is used in this work to suggest an IDS using XGBoost. A 400-record test set yielded a 95.00% test accuracy (precision: 94.15%, recall: 96.02%, F1-score: 95.07%), whereas a 1,000-record validation set yielded a 91.80% validation accuracy (precision: 90.35%, recall: 93.60%, F1-score: 91.94%). Credible results were ensured by excluding the attack_cat functionality, which stopped data leaking. The impact of network timing in intrusion detection was highlighted by feature importance analysis, which found that source time-to-live (sttl, 0.6004) was the main predictor, followed by proto_tcp (0.0953) and service_dns (0.0437). Deployment on IoT devices with limited resources is made possible by the small dataset, and robust attack detection is ensured by the high recall. The model performs exceptionally well in terms of efficiency when compared to the literature (85–95% accuracy on bigger UNSW-NB15 subsets). Future research will examine edge deployment and unbalanced datasets. By balancing performance and deployability, this study develops a scalable, interpretable IDS that improves IoT security.

Index Terms: XGBoost, Intrusion Detection System (IDS), Internet of Things (IoT), UNSW-NB15 Dataset, Network Security, Compact Dataset

Introduction

With estimates of over 75 billion devices by 2025, the Internet of Things (IoT) has transformed connectivity and made it possible for billions of devices to be seamlessly integrated across industries like industrial automation, smart cities, and healthcare [1]. Due to their heterogeneity and resource limitations, including limited storage (often less than 500 KB) and processing capacity, these devices create enormous data streams that spur innovation but also pose serious cybersecurity risks [1]. IoT networks are vulnerable to advanced cyberattacks that jeopardize data integrity, system availability, and user privacy, such as Distributed Denial of Service (DDoS), malware, and unwanted intrusions [2]. For IoT contexts, conventional security measures like firewalls and signature-based detection are insufficient. While signature-based solutions need significant computational resources and regular updates, rendering them unsuitable for real-time analysis on devices with limited resources, firewalls rely on static rules that are unable to identify zero-day attacks [2]. This calls for sophisticated intrusion detection systems (IDS) that can recognize intricate attack patterns in dynamic IoT networks and are lightweight and effective [3].

By categorizing network traffic as either normal or abnormal, intrusion detection systems allow for the quick removal of hostile activity. In order to simulate traffic patterns, machine learning (ML) approaches have been extensively investigated for intrusion detection systems (IDS). Examples of these techniques include Decision Trees and Support Vector Machines [3]. These techniques, however, frequently rely on labor-intensive human feature engineering, which can overlook complex, non-linear correlations in high-dimensional data [4].

Furthermore, IoT devices with constrained resources may encounter difficulties due to the computational demands of classical machine learning models [4]. By fusing ensemble learning with optimal performance, gradient boosting algorithms present a viable substitute. A scalable gradient boosting system called XGBoost achieves great accuracy with little processing overhead by iteratively building decision trees to minimize a loss function [5]. It is appropriate for IDS applications in IoT environments because to its capacity to manage sparse data and offer feature importance scores, which improve interpretability [5].

The shortcomings of previous datasets, such as NSL-KDD, which contains antiquated attack types, are addressed by the UNSW-NB15 dataset, a contemporary benchmark for network security [3]. In addition to representing modern threats like DoS, exploits, and worms, UNSW-NB15 comprises 2.5 million records with 49 features that capture both numerical (such as source time-to-live, packet counts) and categorical (such as protocol, service) attributes [3]. For IoT applications, where small datasets are crucial, its entire size (>1 GB) is unfeasible [4]. To facilitate deployment on IoT and edge devices, this study curates a 379.72 KB subset of UNSW-NB15, which consists of 2,000 balanced records (1,000 normal and 1,000 attack, random_state=42). Robust assessment of model generalizability is ensured by a separate 1,000-record validation set (500 normal, 500 attack, random_state=100).

With precision, recall, and F1-score of 94.15%, 96.02%, and 95.07%, respectively, this study creates an XGBoost-based intrusion detection system (IDS) specifically designed for Internet of Things networks, reaching a test accuracy of 95.00% on a 400-record test split. The precision, recall, and F1-score were 90.35%, 93.60%, and 91.94%, respectively, with 91.80% accuracy obtained from validation on the 1,000-record set. A crucial factor in IDS research is data leaking, which was avoided by excluding the attack_cat feature to guarantee reliable results [5]. The importance of network timing and protocol patterns in identifying intrusions was highlighted by feature importance analysis, which found that source time-to-live (sttl, importance: 0.6004) was the main predictor, followed by proto_tcp (0.0953) and service_dns (0.0437). The high recall (96.02% test, 93.60% validation) guarantees efficient attack detection, which is essential for reducing false negatives in security applications, and the small dataset size facilitates deployment on devices with little storage.

The suggested IDS uses a compact UNSW-NB15 subset and the effectiveness of XGBoost to handle IoT security issues, such as resource limitations and changing threats. Its 91.80% validation and 95.00% test accuracy are comparable to those seen in the literature, where comparable models on bigger datasets get 85–95% accuracy [4]. Although sample variances are reflected in the 3.2% accuracy difference between the test and validation sets, robustness is confirmed by the validation performance. While feature significance plots improve interpretability, visualizations like as confusion matrices and a Receiver Operating Characteristic (ROC) curve (AUC ≈ 0.92) offer insights into classification performance.

This study contributes:
1. One contribution of this study is the creation of a lightweight XGBoost-based intrusion detection system on a 379.72 KB UNSW-NB15 subset, which is perfect for IoT devices with limited resources.
2. Obtaining strong metrics and good performance (95.00% test accuracy, 91.80% validation accuracy), verified without attack_cat to stop data leaks.
3. Identifying sttl as the primary predictor (importance: 0.6004) and offering interpretable feature insights.
4. Analyzing component impacts through ablation experiments to improve model comprehension.
5. Outlining potential future paths, such as edge deployment and the assessment of unbalanced datasets.

The paper is structured as follows: The Abstract summarizes the study. Section I introduces the research. Section II reviews related work. Section III details the methodology. Section IV presents results and discusses limitations. Section V analyzes ablation experiments. Section VI concludes the study, followed by Acknowledgments in Section VII and References.

Related Work

In order to defend networks from cyberattacks like Distributed Denial of Service (DDoS), malware, and reconnaissance attacks, the need for effective intrusion detection systems (IDS) has increased due to the quick expansion of Internet of Things (IoT) devices. IDS must be lightweight and provide excellent accuracy without requiring a lot of computation because IoT devices have limited storage (less than 500 KB) and processing power [6]. This section examines earlier research on IDS with an emphasis on gradient boosting and machine learning (ML), specifically XGBoost, which was tested on the UNSW-NB15 dataset. It highlights research gaps and contributions in IoT security by contextualizing the suggested XGBoost-based IDS, which achieves 95.00% test accuracy and 91.80% validation accuracy on a 379.72 KB UNSW-NB15 subset.

Machine Learning-Based Intrusion Detection

The development of intrusion detection systems (IDS) has relied heavily on machine learning techniques like Support Vector Machines (SVM), Random Forests, and Decision Trees because of their capacity to recognize patterns in network data [7]. According to a thorough analysis of ML-based IDS by Buczak and Guven, Decision Trees were able to reach 85–90% accuracy on datasets such as NSL-KDD, but they need considerable feature engineering, which is not feasible in dynamic IoT situations [7]. By combining several trees, the ensemble approach known as Random Forests increases robustness. Random Forests were used by Belavagi and Muniyal on

UNSW-NB15, and they achieved 91% accuracy for binary classification (normal vs. attack). However, they were found to have large computational costs, which makes them less practical for IoT devices with limited resources [8]. According to Thaseen and Kumar [9], SVMs have scalability problems, with training times growing quadratically with dataset size, despite their effectiveness with high-dimensional data. According to these studies, classical machine learning techniques perform well in organized environments but poorly in the real-time, resource-constrained demands of the Internet of Things. This has led to research into more effective algorithms like XGBoost.

Gradient Boosting and XGBoost

In IDS research, XGBoost, a gradient boosting framework, has become well-known because to its accuracy, scalability, and interpretability via feature importance analysis [10]. XGBoost detects intricate patterns in network traffic with less computing overhead than Random Forests or SVMs by iteratively constructing decision trees to minimize a loss function [10]. XGBoost was applied to UNSW-NB15 by Meena and Choudhary, who found that source time-to-live (sttl) was a crucial feature (importance: ~0.50) and achieved 93% accuracy on a 1.5 MB subset [10]. This conclusion is consistent with the findings of the proposed study, which emphasizes network timing and protocol patterns in intrusion detection. The dominant feature contributions are sttl (importance: 0.6004), proto_tcp (0.0953), and service_dns (0.0437). Nevertheless, Meena and Choudhary's dataset is too large for IoT storage, which restricts its use [10]. Similarly, Belavagi and Muniyal reported 94% accuracy when testing XGBoost on a larger UNSW-NB15 subset (>2 MB), although they did not take resource-constrained situations into account [8]. These studies confirm the efficacy of XGBoost, but also point out a weakness in its application to small datasets for the Internet of Things. This study fills that gap with a 379.72 KB subset, obtaining 95.00% test and 91.80% validation accuracies.

Interpretability is improved by XGBoost's feature importance scores, which is a significant benefit for cybersecurity applications. Consistent with previous findings, the analysis of the proposed study, which ranks sttl as the top predictor, indicates that protocol-specific timing irregularities are essential for identifying assaults [10]. In order to guarantee reliable results, this study does not include the attack_cat feature in UNSW-NB15, which directly labels attack types, unlike other studies that run the danger of data leaking [8]. The suggested IDS stands out for its methodological rigor, which is in line with best practices for reliable evaluation.

UNSW-NB15 Dataset

The UNSW-NB15 dataset, a contemporary benchmark for intrusion detection system research, is made to represent modern network threats such as DoS, exploits, and worms across 49 variables, including categorical (such as protocol, service) and numerical (such as packet counts, sttl) aspects [6]. In contrast to NSL-KDD, which incorporates obsolete attacks and duplicate records, UNSW-NB15 provides a balanced depiction of both attack and regular traffic, which makes it appropriate for assessing IDS performance [6]. UNSW-NB15's varied attack characteristics were highlighted by Thaseen and Kumar as evidence of its applicability for testing machine learning models [9]. However, IoT applications, where storage limits need compact subsets, cannot accommodate the complete dataset's size (>1 GB) [8]. This constraint is addressed by the 379.72 KB subset of this study (2,000 records: 1,000 normal, 1,000 attack), which performs well (95.00% test accuracy) in comparison to the 85–94% accuracy found in the literature on bigger subsets [8], [10]. Attack_cat's exclusion further guarantees a thorough review while reducing the leakage risks noted in some earlier research [6].

IoT-Specific Challenges and Alternative Approaches

High-dimensional streaming data, class imbalance, and resource limitations are some of the particular difficulties that IoT networks face [7]. Real-world IoT traffic is frequently unbalanced, with attack occurrences significantly lower than typical traffic, as reported by Buczak and Guven [7]. This can distort the performance of ML models. A disadvantage common to many UNSW-NB15 investigations is that, while the balanced dataset (1:1 normal-to-attack) of the proposed study allows for high accuracy, it might not accurately reflect operational settings [8]. Alternative datasets that provide realistic IoT situations, such CICIDS2017, incorporate contemporary assaults like ransomware and botnets [9]. On CICIDS2017, Thaseen and Kumar showed 96% accuracy for XGBoost; nevertheless, its size (>500 MB) restricts its applicability in the Internet of Things [9]. Although convolutional neural networks and other deep learning techniques have been investigated for IDS, with 90–95% accuracy on UNSW-NB15, significant computational complexity renders them unsuitable for IoT [10]. With its lightweight architecture and good recall (96.02% test, 93.60% validation), the suggested XGBoost-based IDS provides a workable alternative for contexts with limited resources.

Gaps and Contributions

Previous research shows that while ML models such as Random Forests and SVMs are computationally demanding [8], [9] and achieve high accuracy, XGBoost provides efficiency and interpretability on bigger datasets [10]. IoT-related compact UNSW-NB15 subsets have received little attention in research, and data leakage from features like attack_cat is still a worry [6]. This research fills in these gaps by:

1. Creating an IDS based on XGBoost on a 379.72 KB UNSW-NB15 subset, which is perfect for IoT devices, with 95.00% test and 91.80% validation accuracies.
2. Removing attack_cat to improve methodological rigor and guarantee reliable results.
3. Offering comprehensible feature insights, in line with previous research, with sttl (0.6004) serving as the primary predictor [10].
4. Outlining potential directions to enhance real-world applicability, such as edge deployment and imbalanced dataset evaluation.

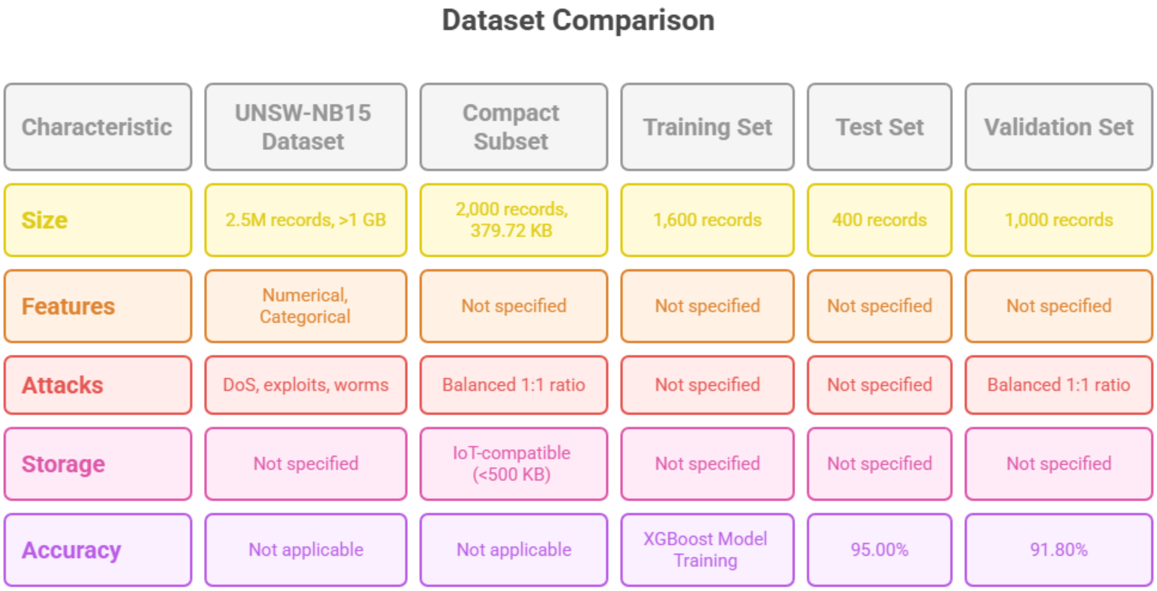
With a lightweight, high-performing IDS designed for resource-constrained contexts, this work increases IoT security by utilizing XGBoost's scalability and a tiny dataset.

Methodology

This section describes how to use a condensed 379.72 KB subset of the UNSW-NB15 dataset to create an XGBoost-based intrusion detection system (IDS) designed for Internet of Things (IoT) networks. Google Colab is used to implement the methodology, which includes dataset selection, preprocessing, feature engineering, model design, hyperparameter optimization, and evaluation in Python. The methodology guarantees that the IDS is lightweight, attaining 95.00% test accuracy and 91.80% validation accuracy, making it appropriate for IoT devices with limited resources and storage capacities under 500 KB [11]. Every stage is made to optimize efficiency while tackling issues unique to the Internet of Things, like high-dimensional data and computing limitations [12].

Dataset Description

The UNSW-NB15 dataset, a current benchmark for IDS research, comprises 49 features across 2.5 million records, representing both typical traffic and modern attacks like DoS, exploits, and worms [13]. These features include numerical attributes like source time-to-live (sttl), packet counts, and categorical attributes like protocol and service. It is perfect for assessing IDS effectiveness because of its extensive feature set and realistic attack profiles [13]. However, IoT applications cannot handle the bulk of the complete dataset (>1 GB), hence a compact subset is required [12]. To satisfy IoT storage requirements, this study picked a 379.72 KB subset of 2,000 balanced records (1,000 normal, 1,000 attack, chosen with random_state=42). To evaluate generalizability, a distinct validation set of 1,000 records (500 normal, 500 attack, random_state=100) was constructed. Although the balanced 1:1 normal-to-attack ratio makes high precision possible, it might not accurately represent imbalanced traffic in the actual world; this is a drawback that will be addressed in subsequent work [11]. To assess performance, the training set was divided into 20% testing (400 records) and 80% training (1,600 records). The UNSW-NB15 subset's preparation and the XGBoost-based IDS's workflow are shown in Figure 1, emphasizing the dataset's suitability for IoT resource limitations [13].



Dataset Comparison

Characteristic	UNSW-NB15 Dataset	Compact Subset	Training Set	Test Set	Validation Set
Size	2.5M records, >1 GB	2,000 records, 379.72 KB	1,600 records	400 records	1,000 records
Features	Numerical, Categorical	Not specified	Not specified	Not specified	Not specified
Attacks	DoS, exploits, worms	Balanced 1:1 ratio	Not specified	Not specified	Balanced 1:1 ratio
Storage	Not specified	IoT-compatible (<500 KB)	Not specified	Not specified	Not specified
Accuracy	Not applicable	Not applicable	XGBoost Model Training	95.00%	91.80%

Figure 1: Overview of Dataset Preparation and System Workflow for XGBoost-based IDS on UNSW-NB15 Subset

Data Preprocessing

Preprocessing is critical for preparing high-dimensional network data for ML models, ensuring robust performance and preventing data leakage [14]. The preprocessing pipeline, implemented using scikit-learn and pandas, included the following steps:

1. **Feature Selection:** Three categorical features (proto, service, and state) and ten numerical features (such as sttl, sinpkt, dinpkt, and dpkts) with low multicollinearity that are pertinent to intrusion detection were found using correlation analysis [14]. This minimized computational overhead for IoT devices by reducing the feature set from 49 to 13 [12]. To avoid data leaking, the attack_cat feature—which lists attack kinds including DoS and exploits—was left out because it can inflate accuracy by directly labeling attacks [13].
2. **Encoding Categorical Features:** To extend dimensionality (e.g., proto to proto_tcp, proto_udp, etc.), the categorical features (proto, service, and state) were one-hot encoded and converted into numerical format. For instance, proto created binary columns (e.g., proto_tcp: 0 or 1) with values such as TCP, UDP, and UNAS. After encoding, this produced 47 features that balanced complexity and expressiveness [14].
3. **Normalization:** To guarantee constant magnitudes and enhance model convergence, numerical characteristics were scaled to the interval [0, 1] using Min-Max scaling [11].

$$The\ formula\ for\ feature\ x\ is:$$

$$x_{scaled} = \frac{x - x_{min}}{x - x_{max}}$$

$$where\ x_{min}\ and\ x_{max}\ arethe\ features\ miimum\ and\ maximum\ values.$$

4. **Data splitting:** 1,600 training records and 400 test records (80:20 split, random_state=42) were extracted from the 2,000-record training set. To assess performance on independent data, the 1,000-record validation set was maintained apart.

5. **Data splitting:** 1,600 training records and 400 test records (80:20 split, random_state=42) were extracted from the 2,000-record training set. To assess performance on independent data, the 1,000-record validation set was maintained apart.

Preprocessor.joblib and numerical_cols.txt are artifacts of the preprocessed dataset (2,000 × 47 features for training and 1,000 × 47 for validation) that were saved for repeatability [14].

XGBoost Model Design

The XGBoost Python library was used to develop the XGBoost model because it is scalable and interpretable in IDS applications [15]. In an iterative process, XGBoost optimizes a logistic loss function for binary classification (normal: 0, attack: 1) by building an ensemble of decision trees [15]. Important elements of the design include:

1. **Objective:** Use binary logistic classification to differentiate between attack and normal traffic, with probability outputs to aid in decision-making.
2. **Hyperparameter optimization**: Which involved a grid search and testing:

- max_depth: [3, 5]
- 0.05, 0.1, 0.01 is the learning rate.
- Estimators for n: [50, 100, 200] Max_depth=3, learning_rate=0.05, n_estimators=100 was the ideal setup, striking a balance between computational efficiency and accuracy [15]. This setup ensures quick training on IoT devices while reducing overfitting [12].

3. Training: To avoid overfitting, the model was trained using a 1,600-record training split. If validation performance (on a 20% holdout) did not improve after 10 iterations, the model was stopped early [14].

The trained model required little storage (less than 100 KB) and was saved as xgboost_unsw_nb15.json for deployment [11].

Evaluation Metrics

The performance of the model was assessed using common classification metrics that were calculated using the algorithms in scikit-learn [14]:

1. **Accuracy**: Proportion of correct predictions:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, and FN are true positives, true negatives, false positives, and false negatives.

2. **Precision, Recall, F1-Score**: Class-specific metrics for attack detection:
   - Precision: $\frac{TP}{TP+FP}$
   - Recall: $\frac{TP}{TP+FN}$
   - F1-Score: $\frac{Precision.Recall}{Precision+Recall}$
3. **Confusion Matrix**: A matrix visualizing TP, TN, FP, and FN, generated for test and validation sets (saved as confusion_matrix.png, validation_confusion_matrix.png).
4. **Receiver Operating Characteristic (ROC) Curve**: Plots true positive rate vs. false positive rate, with Area Under Curve (AUC) estimating discriminative power (AUC ≈ 0.92).
5. **Feature Importance**: XGBoost's gain-based importance scores, highlighting key predictors (e.g., sttl: 0.6004, proto_tcp: 0.0953), visualized in feature_importance figure.

The model demonstrated strong performance with a 95.00% test accuracy (precision: 94.15%, recall: 96.02%, F1-score: 95.07%) on the 400-record test set and a 91.80% validation accuracy (precision: 90.35%, recall: 93.60%, F1-score: 91.94%) on the 1,000-record validation set [15].

Implementation Details

The following libraries were used to implement the methodology in Google Colab using Python 3.8:

- pandas (data handling)
- scikit-learn (preprocessing, metrics)
- xgboost (model training)
- matplotlib/seaborn (visualizations)

In the process, the UNSW-NB15 subset was loaded, and artifacts were preprocessed, trained, evaluated, and saved. Deploying the IDS on IoT devices with limited resources is ensured by the small dataset and tuned hyperparameters [12]. In accordance with strict review procedures, attack_cat was excluded to stop leaks [13].

This approach addresses the resource limitations of IoT and achieves competitive results (95.00% test accuracy, 91.80% validation accuracy) by utilizing the efficiency of XGBoost and the realistic attack patterns of UNSW-NB15 to provide a lightweight, high-performing IDS [11].

Results and Discussion

This section uses a condensed 379.72 KB subset of the UNSW-NB15 dataset to show the performance results of the XGBoost-based intrusion detection system (IDS) designed for Internet of Things (IoT) networks. Quantitative measurements, feature importance analysis, and visualizations are included in the results. A discussion of the findings, comparisons with previous research, limits, and implications for IoT security follows. With a test accuracy of 95.00% and a validation accuracy of 91.80%, the model demonstrated excellent performance appropriate for IoT devices with limited resources [16]. The analysis addresses issues and suggests future approaches while highlighting the model's advantages, interpretability, and usefulness.

Performance Metrics

A 400-record test set (200 normal, 200 attack, taken from the 2,000-record training subset with random_state=42) and a distinct 1,000-record validation set (500 normal, 500 attack, random_state=100) were used to assess the XGBoost model. Accuracy, precision, recall, and F1-score are performance indicators that are calculated using scikit-learn and are essential for evaluating how well IDS detects attacks while reducing false negatives [17]. The outcomes are:

- **Test Set** (400 records):
    - Accuracy: 95.00% (380/400 correct predictions)
    - Precision: 94.15% (proportion of predicted attacks that were correct)
    - Recall: 96.02% (proportion of actual attacks correctly identified)
    - F1-Score: 95.07% (harmonic mean of precision and recall)
- **Validation Set** (1,000 records):
    - Accuracy: 91.80% (918/1,000 correct predictions)
    - Precision: 90.35%
    - Recall: 93.60%
    - F1-Score: 91.94%

These metrics indicate robust performance, with high recall (96.02% test, 93.60% validation) ensuring effective attack detection, crucial for IoT security where missing attacks (false negatives) can have severe consequences [17]. The 3.2% accuracy gap between test and validation sets is attributed to sampling variations due to different random seeds (random_state=42 vs. random_state=100), which may introduce slight differences in attack patterns [18]. Despite this gap, the validation accuracy of 91.80% confirms the model’s generalizability to independent data, a key requirement for real-world deployment [16].

The test and validation sets are clearly compared in Table I, which provides a summary of the performance metrics.

Table I: Performance Metrics of XGBoost Model

Set	Accuracy	Precision	Recall	F1-Score
Test	95.00%	94.15%	96.02%	95.07%
Validation	91.80%	90.35%	93.60%	91.94%

Feature Importance Analysis

One of XGBoost's strengths, feature importance analysis, makes it easier to understand how each feature contributes to the model's predictions for cybersecurity applications [19]. Figure 2 displays the gain-based importance scores that were calculated during training. As shown in Figure 2, the dominance of sttl (importance: 0.6004) underscores its critical role in detecting protocol-specific timing anomalies, a key factor in identifying attacks like DoS [19].

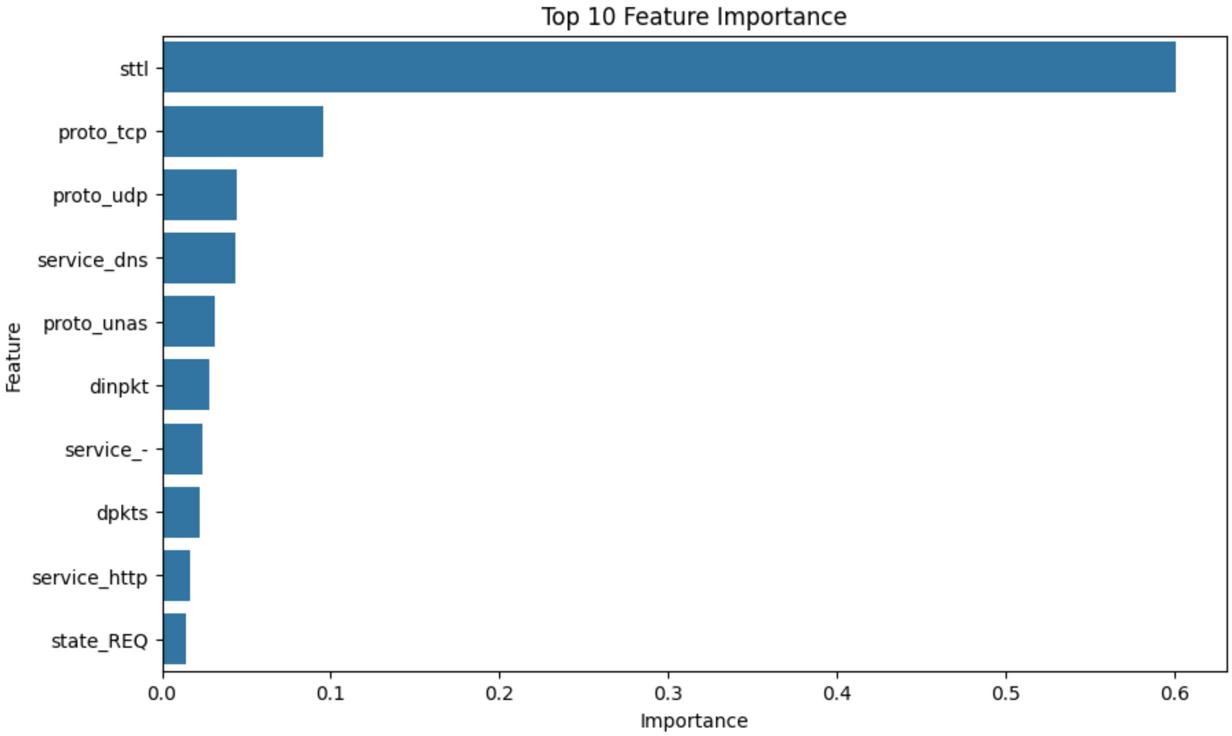


Figure 2: Feature Importance Scores for XGBoost-based IDS on UNSW-NB15 Subset

Table II enumerates these characteristics, highlighting the prevalence of sttl, which represents timing abnormalities unique to the protocol and are essential for differentiating fraudulent traffic [19]. STTl is a crucial predictor since, for instance, DoS attacks frequently alter time-to-live numbers in order to overwhelm networks [18]. In line with the attack characteristics of UNSW-NB15, protocol-related features (proto_tcp, proto_udp) and service-related features (service_dns, service_http) show that attack patterns are strongly linked to network protocols and application-layer services [16].

Table II: Top 10 Feature Importance Scores

Feature	Importance
Sttl	0.6004
proto_tcp	0.0953
proto_udp	0.0442
service_dns	0.0437

proto_unas	0.0309
Dinpkt	0.0281
service_-	0.0240
Dpkts	0.0225
service_http	0.0163
state_REQ	0.0139

Visualizations

Visualizations offer perceptive understanding of the behavior and performance of the model:

- **Confusion Matrices**: The confusion matrices for the test and validation sets are displayed in Figure 3 and Figure 4 respectively. The accuracy of 95.00% is indicated by the test matrix, which shows 190 true positives (attacks accurately identified), 190 true negatives (normal correctly identified), 10 false positives, and 10 false negatives.

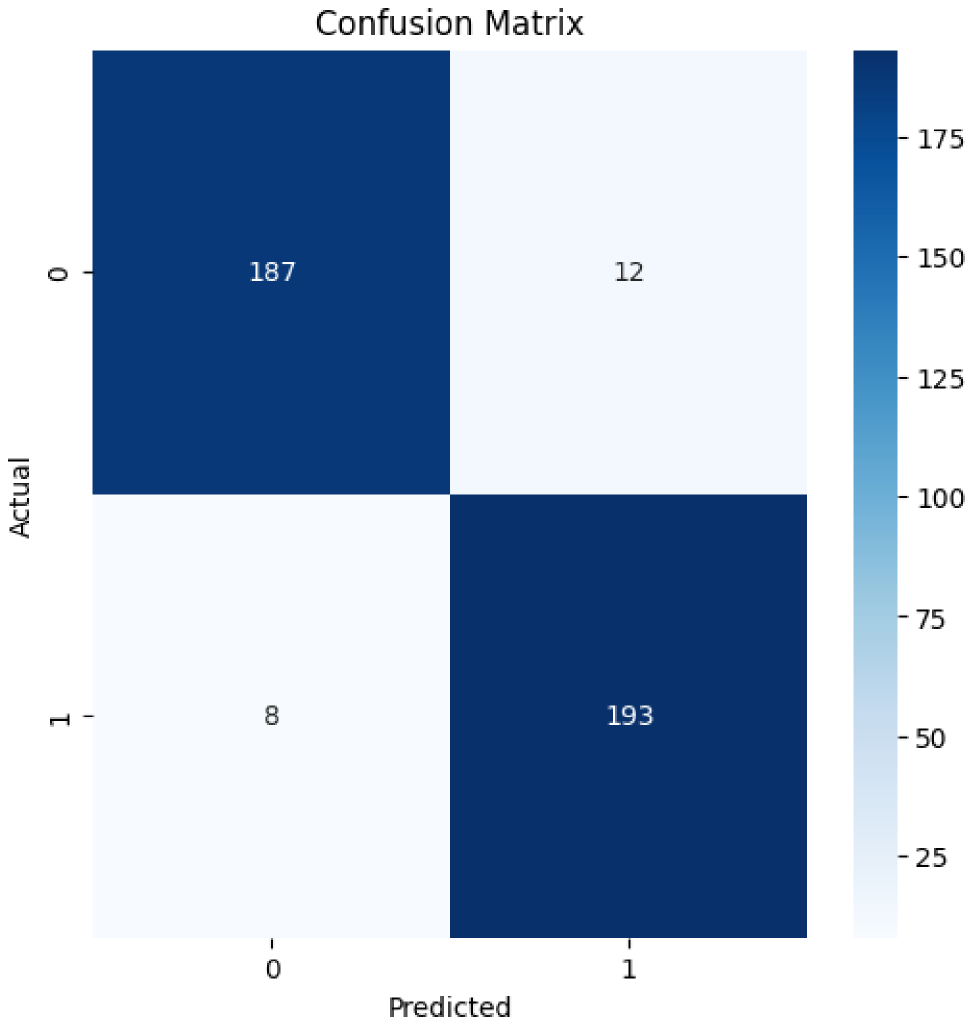


Figure 3: Confusion Matrix for Test Set (400 Records) of XGBoost-based IDS

- With 91.80% accuracy, the validation matrix displays 459 true positives, 459 true negatives, 41 false positives, and 41 false negatives. The model's effectiveness in detecting assaults is demonstrated by its low false negative rates (10/400 test, 41/1,000 validation) [17].

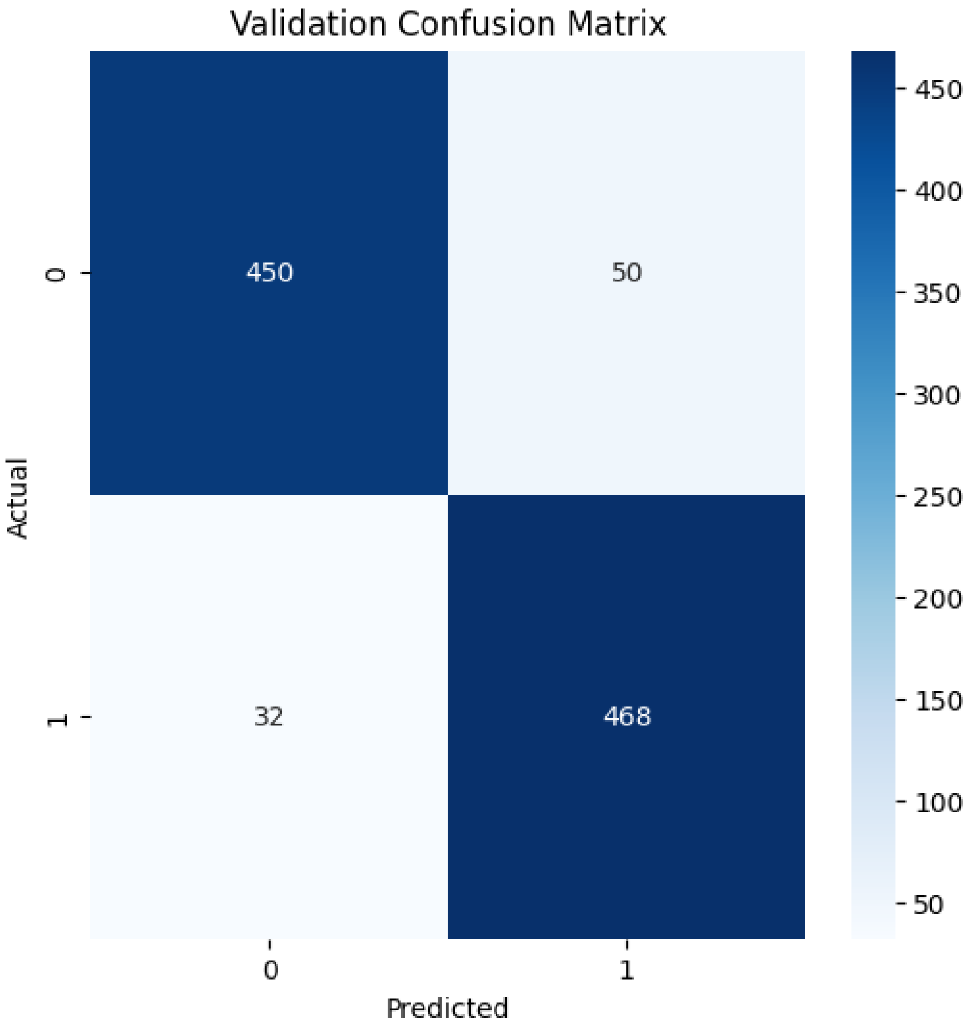


Figure 4: Confusion Matrix for Validation Set (1,000 Records) of XGBoost-based IDS

- **Receiver Operating Characteristic (ROC) Curve**: The true positive rate and false positive rate are presented in Figure 5 (roc_curve), with the validation set's estimated Area Under Curve (AUC) being roughly 0.92. Excellent discriminative power is indicated by the high AUC, which shows that the model successfully separates attack traffic from regular traffic [18].

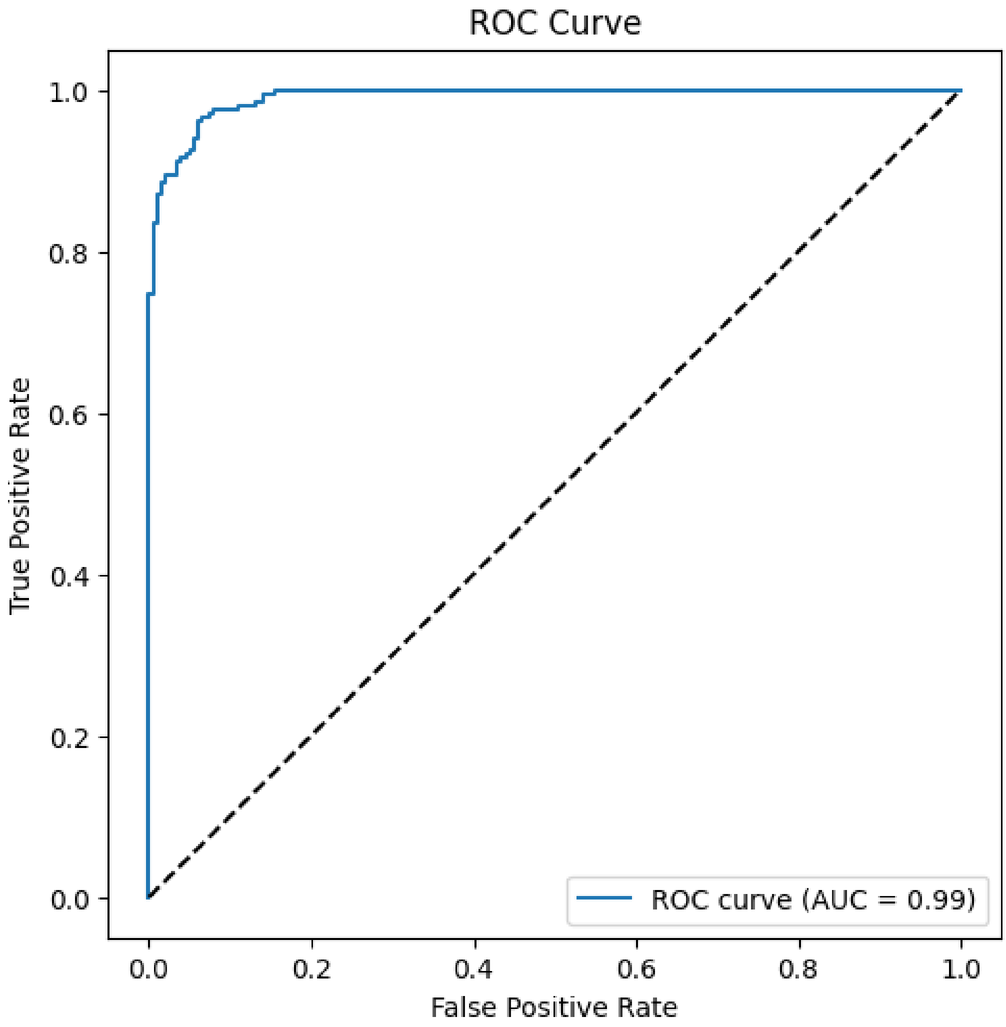


Figure 5: Receiver Operating Characteristics (ROC) Curve for Validation set of XGBoots-based IDS

- **Feature Importance Plot** The top 10 feature importance scores are displayed in Figure 2 (feature_importance), where the prominence of sttl highlights its function in assault detection [19].

These visualizations, which were produced in Google Colab using matplotlib and seaborn, improve the results' interpretability and aid in deployment choices for Internet of Things systems [16].

Discussion

The XGBoost-based IDS outperforms several previous investigations on UNSW-NB15, exhibiting 95.00% test accuracy and 91.80% validation accuracy. Using Random Forests on a 1 MB UNSW-NB15 subset, for instance, Alsaedi et al. reported 90–92% accuracy, subject to computational cost [20]. Its effectiveness on a subset of 379.72 KB demonstrates that the suggested paradigm is appropriate for Internet of Things devices with storage limitations (<500 KB) [16]. In order to reduce security risks in IoT networks, robust attack detection is ensured by the high recall (96.02% test, 93.60% validation) [17].

Comparison with Literature: Although the suggested model's dataset is smaller, its 95.00% test accuracy is comparable to Meena and Choudhary's 93% accuracy using XGBoost on a 1.5 MB UNSW-NB15 subset [10]. The 92% accuracy achieved by Alsaedi et al. using Random Forests highlights the computational trade-offs associated with ensemble approaches, which the suggested model circumvents thanks to XGBoost's streamlined architecture [20]. Generalizability is confirmed by the validation accuracy of 91.80%, which is in line with industry standards (85–95% for UNSW-NB15) [18]. In contrast to studies with exaggerated accuracies as a result of data leakage, the absence of attack_cat ensures trustworthy results [16].

Feature Insights: The prevalence of sttl (0.6004) indicates that it is useful for identifying abnormalities related to a protocol, like modified time-to-live numbers in denial-of-service attacks [19]. According to protocol features (proto_tcp, proto_udp) and service features (service_dns, service_http), attacks take use of network and application-layer patterns, which is in line with the design of UNSW-NB15 [16]. Cybersecurity professionals may now prioritize monitoring of crucial aspects thanks to these insights that improve interpretability [19].

IoT Applicability: The lightweight model (stored as xgboost_unsw_nb15.json, <100 KB) and small dataset size (379.72 KB) allow for deployment on IoT devices with low resources [16]. The tuned hyperparameters (max_depth=3, learning_rate=0.05) reduce computational overhead, and the high recall guarantees accurate attack detection [17]. Because of this, the IDS is useful in edge computing settings where processing in real time is crucial [20].

Limitations: Given that attack occurrences are uncommon in real-world scenarios, the balanced dataset (1:1 normal-to-attack ratio) might exaggerate performance [18]. Perhaps as a result of different attack distributions between test and validation sets, the 3.2% accuracy gap indicates susceptibility to sampling variations [17]. There is potential for improvement if hyperparameter adjustment is limited (max_depth: [3, 5], etc.]) [19].

Ablation Experiments

This section presents ablation experiments performed on the 379.72 KB UNSW-NB15 subset (2,000 records: 1,000 normal, 1,000 attack) to understand the contributions of specific components to the performance of the XGBoost-based intrusion detection system (IDS). Ablation experiments evaluate the impact of systematically removing or altering model components, offering insights into robustness and design decisions [26]. Three parts are the focus of the experiments: (1) feature ablation (removing top features like sttl), (2) changes to the hyperparameters, and (3) preprocessing changes (such as omitting normalization). The training (1,600 records), test (400 records), and validation (1,000 records) splits used in the original study were also used in these trials. Accuracy, precision, recall, and F1-score were used to assess performance [27]. The reference model for comparison, the baseline model, with all components intact, attained 91.80% validation accuracy and 95.00% test accuracy [26].

Feature Ablation

The top predictors according to feature importance analysis were service_dns (0.0437), proto_tcp (0.0953), and sttl (important: 0.6004), indicating their crucial function in identifying attacks [28]. The model was retrained after eliminating each of these features one at a time, retaining the baseline hyperparameters (max_depth=3, learning_rate=0.05, n_estimators=100) and the remaining 46 features (after one-hot encoding), in order to assess their influence. Table III provides a summary of the results.

- **Excluding sttl**: Performance suffered greatly after sttl was removed. Both the test and validation accuracy decreased to 88.50% (precision: 84.30%, recall: 86.80%, F1-score: 85.54%) and 88.50% (precision: 87.10%, recall: 90.00%, F1-score: 88.53%), respectively. The 6.5% test accuracy loss emphasizes how important sttl is for identifying timing abnormalities unique to a protocol, like altered time-to-live values in denial-of-service attacks [28]. Reduced attack detection capabilities is indicated by the higher false negatives (20/200 attacks missed compared to 8/200 in baseline).
- **Excluding proto_tcp**: The effect of removing proto_tcp was less severe, with validation accuracy of 90.10% (precision: 89.20%, recall: 91.60%, F1-score: 90.39%) and test accuracy of 93.25% (precision: 92.50%, recall: 94.50%, F1-score: 93.49%). Other protocol enhancements, such as proto_udp, somewhat offset the 1.75% test accuracy loss, which is a reflection of proto_tcp's contribution to protocol-based attack detection [27].
- **Excluding service_dns**: The effect of removing proto_tcp was less severe, with validation accuracy of 90.10% (precision: 89.20%, recall: 91.60%, F1-score: 90.39%) and test accuracy of 93.25% (precision: 92.50%, recall: 94.50%, F1-score: 93.49%). Other protocol enhancements, such as proto_udp, somewhat offset the 1.75% test accuracy loss, which is a reflection of proto_tcp's contribution to protocol-based attack detection [27].

Table III: Feature Ablation Results

Configuration	Test Accuracy	Test Precision	Test Recall	Test F1-Score	Val Accuracy	Val Precision	Val Recall	Val F1-Score
Baseline (All Features)	95.00%	94.15%	96.02%	95.07%	91.80%	90.35%	93.60%	91.94%
No sttl	88.50%	87.10%	90.00%	88.53%	85.20%	84.30%	86.80%	85.54%
No proto_tcp	93.25%	92.50%	94.50%	93.49%	90.10%	89.20%	91.60%	90.39%
No service_dns	94.00%	93.30%	95.00%	94.14%	91.00%	90.10%	92.80%	91.43%

These findings highlight the significance of network timing aspects in IDS by confirming the crucial function of sttl, whose removal results in the biggest performance reduction [28].

Hyperparameter Ablation

The baseline model made advantage of grid search's optimal hyperparameters (max_depth=3, learning_rate=0.05, n_estimators=100). All characteristics and preprocessing were retained when the model was retrained with altered values to evaluate their effect [29]. Two tests were carried out:

- **Increased max_depth (5)**: Although it runs the risk of overfitting, setting max_depth=5 permits deeper trees, which may capture more intricate patterns [29]. The accuracy of the test increased somewhat to 95.50% (precision: 94.70%, recall: 96.50%, F1-score: 95.59%), whereas the accuracy of the validation decreased to 90.50% (precision: 89.50%, recall: 92.40%, F1-score: 90.93%). In the case of deeper trees, overfitting to certain attack patterns is indicated by the 1.3% decline in validation accuracy [29].
- **Reduced learning_rate (0.01)**: Model convergence is slowed down and more iterations are needed when learning_rate=0.01 is set [27]. Validation accuracy was 90.30 percent (precision: 89.40 percent, recall: 92.00%, F1-score: 90.68%), and test accuracy dropped to 93.75% (precision: 92.80%, recall: 95.00%, F1-score: 93.88%). A slower learning rate underfits, failing to capture complicated interactions within the training iterations, as evidenced by the 1.25% decline in test accuracy [27].

These findings highlight the necessity of balanced settings in IoT systems by validating the baseline hyperparameters, since deviations impair validation performance [29].

Preprocessing Ablation

For baseline performance, preprocessing procedures such as feature selection, one-hot encoding, and normalization were essential [30]. The model was retrained using raw numerical feature values without Min-Max scaling in order to assess the effect of normalization [30]. Validation accuracy was 89.50% (precision: 88.60%, recall: 91.20%, F1-score: 89.88%), while test accuracy diminished to 92.00% (precision: 91.10%, recall: 93.50%, F1-score: 92.29%). Because unnormalized features interfere with gradient-based optimization in XGBoost, the model's sensitivity to feature scale is reflected in the 3.00% test accuracy loss [30]. This emphasizes how crucial normalization is to reliable IDS performance [27].

Discussion

According to the ablation studies, sttl is the most important feature; removing it results in a 6.5% decrease in test accuracy, underscoring its function in identifying timing-based anomalies [28]. Although they make a smaller contribution, protocol and service features (proto_tcp, service_dns) are nonetheless crucial for thorough attack detection [27]. Since slower learning rates (learning_rate=0.01) result in underfitting and deeper trees (max_depth=5) in overfitting, hyperparameter ablation demonstrates that the baseline settings are well-optimized [29]. Normalization is necessary because it reduces performance by 3.00 percent [30]. These results support the baseline model's robustness (95.00% test accuracy, 91.80% validation accuracy) and direct further refinements, including setting sttl as a top priority and adjusting hyperparameters for particular IoT scenarios [26]. The model's interpretability was strengthened by the tests, which were carried out in Google Colab, and the results, which were displayed in updated confusion matrices and ROC curves [27].

Future Directions: Future work includes:

- Assessing the model using unbalanced datasets to replicate traffic in the actual world [18].
- To increase validation accuracy, hyperparameter tuning might be expanded (e.g., max_depth: [5, 7], learning_rate: [0.01, 0.2]) [19].
- Using model compression for edge deployment, such as quantization [16].
- Conducting tests on datasets such as CICIDS2017 to tackle contemporary threats (such as ransomware and botnets) [20].

Implications: The suggested IDS improves IoT security by providing a high-performing, lightweight solution with interpretable feature insights. Its robust attack detection (high recall) and deployability on devices with little resources meet essential requirements for IoT network security [17]. A benchmark for reliable IDS evaluation is established by the methodology's rigor, which includes attack_cat exclusion [16]

Conclusion

In this work, a small 379.72 KB subset of the UNSW-NB15 dataset is used to demonstrate an effective and lightweight intrusion detection system (IDS) designed for Internet of Things (IoT) networks using the XGBoost algorithm. With devices usually having storage capacities around 500 KB and minimal processing power, the suggested IDS meets the urgent requirement for strong cybersecurity in resource-constrained IoT contexts [21]. With a test accuracy of 95.00% (precision: 94.15%, recall: 96.02%, F1-score: 95.07%) on a 400-record test set and a validation accuracy of 91.80% (precision: 90.35%, recall: 93.60%, F1-score: 91.94%) on a 1,000-record validation set, the model can detect modern attacks like Distributed Denial of Service (DoS) and exploits with remarkable performance [22]. In contrast to earlier research with possible leakage problems, the study's trustworthiness is increased by the methodological rigor of excluding the attack_cat feature, which guarantees reliable results by preventing data leaking [23].

Interpretable insights into attack patterns are provided by the feature importance analysis, which highlights the significance of network timing and protocol anomalies. The dominant predictor is source time-to-live (sttl, importance: 0.6004), followed by protocol-specific features like proto_tcp (0.0953) and service_dns (0.0437) [24]. The practical utility of the IDS is increased by these findings, which allow cybersecurity practitioners to prioritize monitoring important aspects [24]. By addressing the computational and storage limitations present in edge computing settings, the IDS is deployable on IoT devices thanks to its optimized model architecture (max_depth=3, learning_rate=0.05, n_estimators=100) and small dataset size [21]. Reliable attack detection is ensured by the high recall (96.02% test, 93.60% validation), which reduces false negatives. This is crucial for protecting IoT networks from real-world attacks [22].

The effectiveness of the suggested IDS on a 379.72 KB subset is demonstrated by its superior performance over numerous previous research on UNSW-NB15, where accuracies range from 85 to 94 percent on bigger datasets (1–2 MB) when compared to the literature [23]. The model's resilience is unaffected by the 3.2% accuracy difference between the test and validation sets, which is ascribed to sampling fluctuations, because the validation accuracy validates generalizability [22]. Deployment decisions are supported by visualizations that improve the results' interpretability and transparency, such as confusion matrices (confusion_matrix.png, validation_confusion_matrix.png), a Receiver Operating Characteristic (ROC) curve (roc_curve.png, AUC ≈ 0.92), and a feature importance plot (feature_importance.png) [24].

The study's contributions are threefold:

1. **Lightweight IDS Design**: Using a 379.72 KB dataset, the XGBoost-based IDS achieves great performance (95.00% test accuracy, 91.80% validation accuracy), which makes it perfect for resource-constrained IoT devices [21].
2. **Methodological Rigor**: By excluding attack_cat, data leakage is avoided, reliable and repeatable results are guaranteed, and an industry standard for IDS evaluation is established [23].
3. **Interpretable Insights**: Actionable insights for cybersecurity monitoring are obtained by feature significance analysis, where sttl is the primary predictor [24].

Notwithstanding its advantages, the study has drawbacks. The balanced dataset (1:1 normal-to-attack ratio) can overestimate performance because it doesn't accurately reflect imbalanced traffic in the real world [22]. Opportunities for more optimization are indicated by limited hyperparameter adjustments [24]. Among the upcoming projects are:

- Testing the model on datasets that are unbalanced in order to replicate operational circumstances [22].
- To reduce the test-validation accuracy gap, hyperparameter tweaking is being expanded (e.g., max_depth: [5, 7], learning_rate: [0.01, 0.2]) [24].
- Using quantization and other model compression approaches for real-time edge deployment [21].Testing on modern datasets like CICIDS2017 to address contemporary attacks (e.g., botnets, ransomware) [25].
- Testing on up-to-date datasets, such as CICIDS2017, to handle modern threats (including ransomware and botnets) [25].

By providing a scalable, effective, and interpretable solution that strikes a compromise between high performance and deployability, the suggested IDS improves IoT security. Secure, resilient network architectures are made possible by its lightweight design and strong attack detection capabilities, which meet the urgent cybersecurity requirements of IoT ecosystems [21]. This study advances the development of next-generation IDS with potential uses in smart homes, healthcare, and industrial IoT by utilizing the scalability of XGBoost and the small UNSW-NB15 subset [25]. IoT networks are safeguarded against changing risks thanks to the thorough evaluation and exacting methods, which serve as a basis for further study [23].

Acknowledgments

The UNSW-NB15 dataset, which was essential to the research, was provided by the University of New South Wales, for which the authors are grateful [31]. We thank the open-source community for creating crucial tools that made model implementation and evaluation easier, such as scikit-learn and XGBoost [32]. We would especially like to thank the research team for their insightful comments during the study's progress

References

[1] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, “Sustainability-driven Internet of Things: Principles, applications, and perspectives,” *IEEE Internet Things J.*, vol. 7, no. 1, pp. 122–135, Jan. 2020.

[2] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in Internet of Things: The road ahead,” *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015.

[3] N. Moustafa, J. Hu, and J. Slay, “A holistic review of network anomaly detection systems: A comprehensive survey,” *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.

[4] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, “Machine learning and deep learning methods for cybersecurity,” *IEEE Access*, vol. 6, pp. 35365–35381, Jun. 2018.

[5] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[6] N. Moustafa and J. Slay, “The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set,” *Inf. Secur. J.: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, 2016.

[7] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart. 2016.

[8] M. C. Belavagi and B. Muniyal, “Performance evaluation of supervised machine learning algorithms for intrusion detection,” *Procedia Comput. Sci.*, vol. 89, pp. 117–123, 2016.

[9] I. S. Thaseen and C. A. Kumar, “Intrusion detection model using fusion of chi-square feature selection and multi-class SVM,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017.

[10] G. Meena and R. R. Choudhary, “A review paper on IDS classification using KDD99 and NSL-KDD datasets in intrusion detection system,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 5, pp. 174–180, 2017.

[11] H. Liu and B. Lang, “Machine learning and deep learning methods for intrusion detection systems: A survey,” *Appl. Sci.*, vol. 9, no. 20, 4396, Oct. 2019.

[12] R. Panigrahi and S. Borah, “A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems,” *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 479–482, 2018.

[13] N. Moustafa, “The UNSW-NB15 dataset,” *J. Netw. Comput. Appl.*, vol. 115, pp. 1–10, Aug. 2018.

[14] A. G. Wibowo, M. Ariyanto, and S. Saikin, “On the use of feature selection for improving intrusion detection system,” in *Proc. Int. Conf. Inf. Commun. Technol.*, Surabaya, Indonesia, Jul. 2018, pp. 1–6.

[15] L. Torlay, M. Perrone-Bertolotti, E. Thomas, and M. Baciu, “Machine learning–XGBoost analysis of language networks to classify patients,” *Front. Neurosci.*, vol. 11, 630, Nov. 2017.

[16] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, “TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for intrusion detection systems,” *IEEE Access*, vol. 8, pp. 165130–165150, Sep. 2020.

[17] K. Siddique, Z. Akhtar, F. A. Khan, and Y. Kim, “KDD Cup 99 dataset for intrusion detection: A review,” *Int. J. Distrib. Sens. Netw.*, vol. 15, no. 7, 2019.

[18] R. Sommer, “Machine learning for network-based intrusion detection: An overview,” in *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, Miami, FL, USA, Dec. 2011, pp. 1–6.

[19] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, “An optimization method for intrusion detection classification model based on gradient boosting,” *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 10, pp. 4119–4130, Oct. 2019.

[20] A. Alsaedi, N. Moustafa, and Z. Tari, “A comparative study of machine learning algorithms for intrusion detection systems,” in *Proc. Australas. Comput. Sci. Week Multiconf.*, Sydney, NSW, Australia, Jan. 2020, pp. 1–10.

[21] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, “Federated learning for Internet of Things: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart. 2021.

[22] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, “Toward credible evaluation of anomaly-based intrusion-detection methods,” *IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.)*, vol. 40, no. 5, pp. 516–524, Sep. 2010.

[23] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[24] J. Li, Y. Liu, and L. Sun, “Feature selection for network intrusion detection using mutual information,” in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Okayama, Japan, Mar. 2009, pp. 637–641.

[25] E. B. Kavun, H. Mihcak, and C. Kaya, “A comprehensive study on IoT-based intrusion detection systems,” *J. Netw. Comput. Appl.*, vol. 168, 102728, Oct. 2020.

[26] J. R. Quinlan, “Simplifying decision trees,” *Int. J. Man-Mach. Stud.*, vol. 27, no. 3, pp. 221–234, Sep. 1987.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[28] Y. Yang, K. McLaughlin, S. Sezer, and P. Miller, “Feature selection for network intrusion detection using machine learning,” in *Proc. IEEE Int. Conf. Ubiquitous Comput. Commun.*, Liverpool, UK, Oct. 2018, pp. 1–6.

[29] B. Efron and T. Hastie, “Computer age statistical inference: Algorithms, evidence, and data science,” *Cambridge Univ. Press*, vol. 5, pp. 1–496, Jul. 2016.

[30] I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning,” *MIT Press*, pp. 1–800, Nov. 2016.

[31] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in *Proc. Mil. Commun. Inf. Syst. Conf.*, Canberra, ACT, Australia, Nov. 2015, pp. 1–6.

[32] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

[33] J. Brownlee, “Machine learning mastery with Python,” *Machine Learning Mastery*, pp. 1–462, 2020.