

XGBoost Model Optimization Using PCA for Classification of Cyber Attacks on The Internet of Things

Afrijal Rizqi Ramadan¹, Mokhamad Amin Hariyadi², Agung Teguh Wibowo Almais³

^{1,2,3} Faculty of Science and Technology, Program Specification for Master Study in Computer Science (Magister Informatika) Universitas Islam Negeri Maulana Malik Ibrahim, Indonesia

Article Info

Article history:

Received Oct 10, 2025

Revised Nov 29, 2025

Accepted Dec 31, 2025

Keywords:

Cyber Attack

Internet of Things

Smart City

XGBoost

Principal Component Analysis

SMOTE

ABSTRACT

The rapid expansion of the Internet of Things (IoT) ecosystem has increased its susceptibility to cyberattacks, creating a critical need for reliable Intrusion Detection Systems (IDS). However, IDS performance is often hindered by severe class imbalance, high-dimensional features, and similarities among attack behaviors. This study proposes an optimized XGBoost model enhanced with the Synthetic Minority Over-sampling Technique (SMOTE) and Principal Component Analysis (PCA) to address these challenges. A systematic grid-search procedure was employed to ensure transparency, reproducibility, and optimal hyperparameter selection. The original imbalance ratio of approximately 1:27 was successfully normalized to nearly 1:1 through SMOTE. The Gotham dataset used in this study consists of roughly 350,000 IoT traffic records across eight attack categories. Five data-splitting scenarios (50:50 to 90:10) were evaluated using stratified hold-out validation supported by k-fold cross-validation. The optimized model achieved 99.68% accuracy, while extremely high AUC values approaching 1.0 were carefully validated to eliminate potential data leakage. Naive Bayes, Logistic Regression, Support Vector Machine, and Deep Neural Network were included as baseline comparisons. The results demonstrate that combining SMOTE and PCA significantly improves model stability and generalization on imbalanced IoT traffic, confirming the effectiveness of the proposed XGBSP method.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mokhamad Amin Hariyadi,

Faculty of Science and Technology,

Program Specification for Master Study in Computer Science (Magister Informatika)

Universitas Islam Negeri Maulana Malik Ibrahim, Indonesia,

Jl. Gajayana No.50, Dinoyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65144.

Email: adyt2002@uin-malang.ac.id

1. INTRODUCTION

The rapid growth of Internet of Things IoT technology has transformed numerous sectors, including smart homes, smart cities, healthcare, and industrial automation. IoT enables devices to communicate with each other and exchange data in real time, providing significant benefits in efficiency and automation. The IoT market is growing rapidly and is expected to reach a value of approximately USD 94.2 billion by 2026 [1]. However, with the increasing use of IoT, cybersecurity challenges also arise. According to a 2025 report by DeepStrike, there were an average of 820,000 daily attacks targeting IoT devices, with a 46% surge in ransomware attacks in early 2025. The National Cyber and Crypto Agency (NCCA) also stated in an official statement that attacks on IoT infrastructure and supply chains are now a primary focus of national mitigation efforts [2]. Therefore, security issues are a crucial factor that must be considered in the development, implementation, and management of IoT systems.

One approach widely developed to address these issues is an artificial intelligence-based Intrusion Detection System (IDS) [3]. IDS monitors and analyzes network activity to detect any indications of attacks or anomalies that could potentially compromise the system. Several studies have shown that the application of machine learning and deep learning to IDS can improve detection capabilities by recognizing attack patterns [4]. A literature review found that ensemble-based models such as Random Forest (RF) and XGBoost are widely used and offer good performance in IDS with relatively low power consumption [5]. While DL models such as Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) are capable of recognizing more complex attack patterns, they require greater computing power. Conversely, conducted ensemble model testing using various data balancing methods and found that the combination of XGBoost and SMOTE provided better accuracy and attack detection than other models [6]. One of the main challenges in IoT attack classification is class imbalance, where certain attack types contain significantly fewer samples than others. An imbalanced dataset causes bias towards the majority class and ignores attacks that are actually dangerous, thus affecting accuracy [7]. The SMOTE technique balances class distribution to prevent bias in favor of the majority class [8]. Furthermore, to address the high data dimensionality of the IoT dataset, the Principal Component Analysis (PCA) method was used. PCA was chosen because it can reduce data dimensionality and eliminate redundant features without sacrificing important information [9,10].

This study employs the Gotham 2025 dataset, a large-scale IoT traffic dataset containing diverse attack scenarios but exhibiting severe class imbalance and high feature dimensionality [11]. XGBoost is selected due to its strong capability in handling structured data and modeling complex feature interactions, as well as its frequent use in prior studies that report high accuracy [12,13,14]. This algorithm has a boosting mechanism that can reduce prediction errors through iterative learning from decision trees [15]. Nevertheless, the performance of XGBoost is highly dependent on hyperparameter selection. Many previous works rely on trial-and-error tuning, resulting in outcomes that are difficult to replicate. Therefore, systematic hyperparameter optimization is essential to ensure methodological transparency and reproducibility.

Based on these challenges, this research proposes an optimized XGBoost model combined with SMOTE for class balancing and PCA for dimensionality reduction, and applies a grid-search procedure to determine the best hyperparameter configuration. Moreover, because extremely high accuracy and AUC scores approaching 1.0 may indicate overfitting or data leakage, this study implements strict train-test separation, k-fold cross-validation, and duplicate record inspection to ensure model reliability. Overall, this research aims not only to evaluate the performance of the optimized XGBoost model enhanced with SMOTE and PCA but also to address methodological issues that are often overlooked in prior studies.

2. RESEARCH METHOD

This research adopts a quantitative approach, as it relies on numerical data analyzed using statistical techniques, specifically the Confusion Matrix and Area Under the Curve (AUC) Score formulas. In essence, quantitative content analysis serves as a scientific method aimed at uncovering specific characteristics within a dataset and drawing objective conclusions from it. As illustrated in Figure 1, the research process unfolds through five main stages, namely: data collection, system design, experimentation, analysis & evaluation, and finally, drawing synthesis. Each stage is carried out in sequence to ensure that the results obtained are accurate and scientifically grounded.

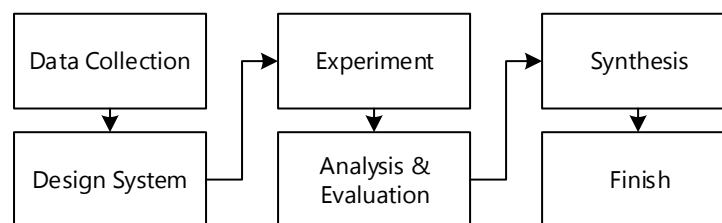


Figure 1. Research Procedure

2.1 Data Collection

First, the data used in this study is the Gotham Dataset, which is secondary data obtained from the Kaggle page (<https://www.kaggle.com/datasets/sachins8201/gotham>). This dataset is specifically designed to support research in the field of Intrusion Detection Systems (IDS) and cybersecurity on the Internet of Things (IoT). The Gotham dataset consists of network traffic data that represents normal activity and cyber attacks that occur in the IoT environment. The data collection process carried out in this study was carried out by downloading the Gotham 2025 dataset. Data collection was carried out on September 23, 2025, with a total of approximately 350,000 data after going through a sampling process [16].

The attributes obtained include frame.time (frt), ip.ttl (ipt), tcp.srcport (tcs), tcp.dstport (tcd), tcp.window_size_value (tcw), tcp.window_size_scalefactor (tcsf), tcp.pdu.size (tcpds), udp.srcport (uds), udp.dstport (udd), and label (lbl). Meanwhile, the label attribute (lbl) in this dataset which is used as a target variable consists of eight (8) classes, namely Denial of Service (DoS), Benign (BEN), Network Scanning (NS), CoAP Amplification Attack Data (CAAD), Telnet Brute Forcing (TBF), Ingress Tool Transfer (ITT), Periodic Command and Control (C&C) Communication (PC3C), Reporting (REP).

2.2 Design System

Second, Figure 2 shows the system design process flow in this study. It explains how the system uses the XGBoost Algorithm to classify cyber attacks on IoT. The process begins with the Gotham 2025 dataset, which is first passed through a preprocessing stage involving data cleaning, normalization, and label encoding. Relevant attributes are then selected through feature extraction to ensure that only informative features enter the learning pipeline.

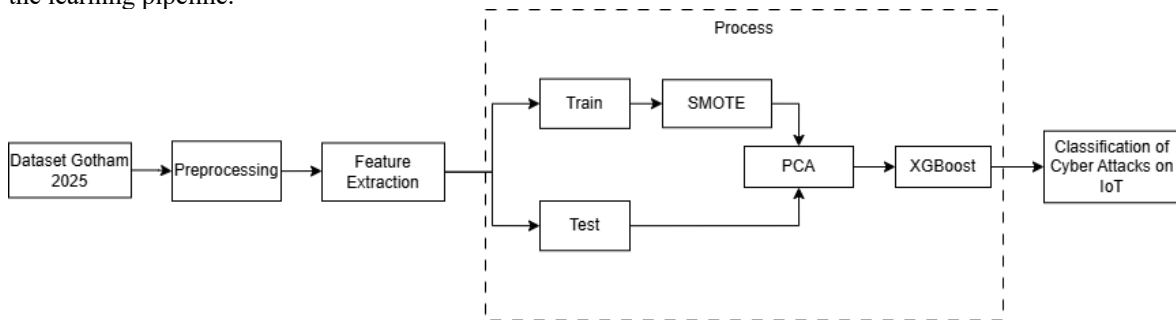


Figure 2. System Design

The prepared data is split into training and testing sets. In the training pipeline, SMOTE is applied to address class imbalance by oversampling minority classes. The balanced data is then transformed using PCA, which reduces dimensionality and removes redundant information. Both the transformed training and testing data are fed into the XGBoost model as the core classifier. This workflow ensures that the model receives clean, balanced, and compact feature representations, enabling accurate and efficient classification of cyber attacks on IoT.

2.3 Experiment

Third, the experiments in this study were conducted using the Python programming language to classify various types of cyber-attacks on Internet of Things (IoT) networks by implementing the XGBoost algorithm. This study evaluated three model variants, namely XGBoost with SMOTE and PCA (XGBSP). Each model as shown in Table 1 was assessed based on its classification performance using the confusion matrix, which includes accuracy, precision, recall, and F1-Score. In addition, the Area Under Curve (AUC) metric was used to evaluate the model's ability to distinguish between attack and normal traffic classes.

Table 1. Composition of Training Data and Test Data

Model Name	Model Code	Split	Training Data	Test Data	Train Data (After SMOTE)	Each Class After SMOTE	Training Code
XGBoost+ SMOTE + PCA	XGBSP	50:50	175,000	175,000	858,784	107,348	XGBSP-E1
		60:40	210,000	140,000	1,030,544	128,818	XGBSP-E2
		70:30	245,000	105,000	1,202,304	150,288	XGBSP-E3
		80:20	280,000	70,000	1,374,064	171,758	XGBSP-E4
		90:10	315,000	35,000	1,545,816	193,227	XGBSP-E5

To ensure reproducibility, the hyperparameter tuning process did not rely on a trial-and-error approach. Instead, this study employed a systematic grid search using predefined parameter ranges for learning rate, max depth, n estimators, subsample, and colsample by tree that shows in Table 2.

Table 2. Grid Search Parameter Range

Hyperparameter	Values Tested	Description
Learning rate (η)	0.01; 0.05; 0.1; 0.2	Controls the step size during boosting updates
Max depth	4; 6; 8; 10	Maximum depth of each decision tree
n_estimators	200	Total number of boosting iterations
Subsample	0.8	Fraction of samples used for training each tree
Colsample_bytree	0.8	Fraction of features used to build each tree

The grid search was combined with k-fold cross-validation to obtain optimal hyperparameters in a structured and repeatable manner. This approach increases transparency, reduces subjectivity, and minimizes the risk of overfitting during model selection. After hyperparameter optimization, the best model configuration was used to train the final XGBoost classifier for each variant. PCA reduced the 12 original attributes into 9 principal components, capturing 100% of the variance. The selection of 9 components was based on the >95% explained variance threshold, with Component 9 marking the point of full variance saturation. The selection of the final model was based on the highest accuracy and the largest AUC score obtained from the validation process.

2.4 Analysis and Evaluation

Fourth, the performance analysis and evaluation in this study utilized a Confusion Matrix, which was used to measure Accuracy, Precision, Recall, F1-Score, TPR, and FPR. The Confusion Matrix is a method frequently used in machine learning to analyze and evaluate, or measure the performance of a developed system. The Confusion Matrix method is used to compare the classified data with the actual data. Testing continued by calculating Accuracy, Precision, Recall, F1-Score, TPR, and FPR [17].

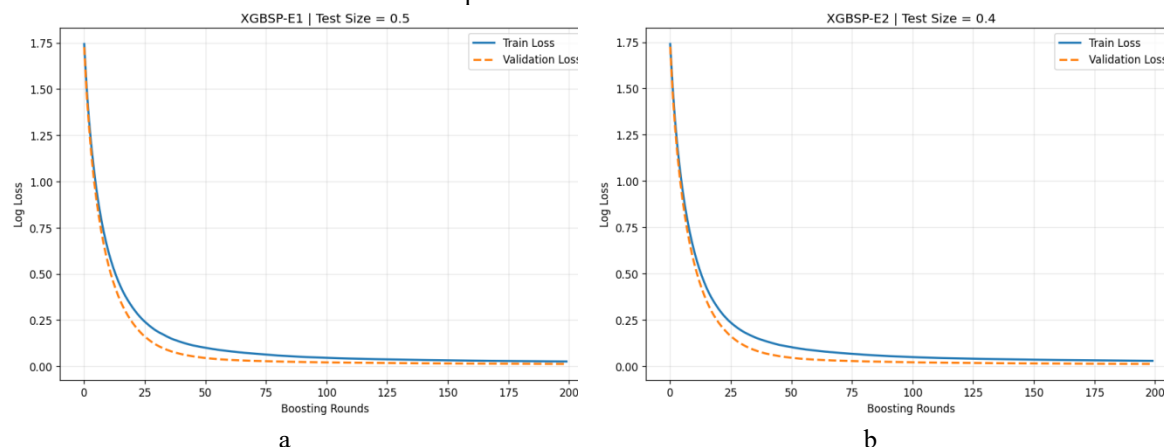
In addition to these metrics, this study also included evaluation using the Area Under the Curve (AUC) analysis of the Receiver Operating Characteristic (ROC). The AUC score is used to assess how well the model is able to distinguish between positive and negative classes. The AUC ranges from 0 to 1, with a value closer to 1 indicating better classification performance. The AUC is an important complement to performance evaluation because it provides comprehensive information on the balance between sensitivity (recall) and specificity (1 - FPR), especially when the dataset has an unbalanced class distribution [18].

2.5 Synthesis

Finally, after all the data collection, system design, experiments, and performance analysis and evaluation processes, a synthesis of each model was conducted to identify the performance patterns and effectiveness of the XGB-SP model in detecting various types of cyberattacks on IoT networks. This synthesis emphasizes the performance comparison between model variants (XGBSP-E1 to XGBSP-E5) based on the metrics of accuracy, precision, recall, F1-Score, and discriminative ability measured through AUC.

3. RESULTS AND DISCUSSION

The experimental evaluation was conducted across five scenarios (XGBSP-E1 to XGBSP-E5) using different train-test split ratios (50:50, 60:40, 70:30, 80:20, and 90:10) to assess the robustness of the proposed model under varying data availability. All experiments used hyperparameters optimally selected through a systematic grid search with 5-fold cross-validation, ensuring full reproducibility and reducing the likelihood of overfitting. The use of consistent hyperparameters across all scenarios allows fair comparative evaluation and isolates the effect of data size on model performance.



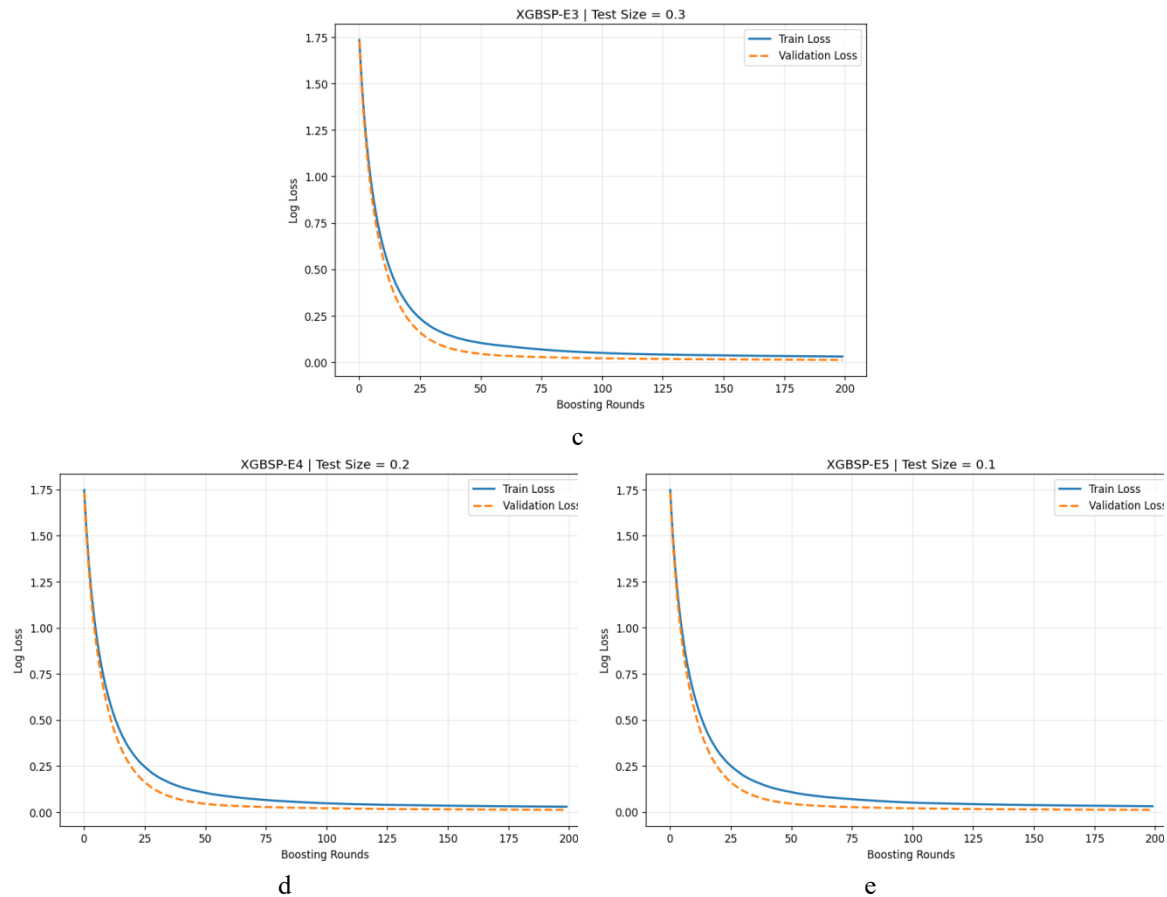


Figure 3. Train Value and Loss Validation Model: (a) XGBSP-E1, (b) XGBSP-E2, (c) XGBSP-E3, (d) XGBSP-E4, (e) XGBSP-E5

3.1 Loss Behavior and Training Dynamics

Across the five experimental scenarios (XGBSP-E1 to XGBSP-E5), as shown in Figure 3 the training and validation loss curves reveal a clear trend in how the model responds to increasing amounts of training data. In the initial scenario (50:50 split), the model achieved a training loss of 0.025754 and a validation loss of 0.012604, showing that even with limited training data, XGBSP was able to learn meaningful patterns. As the training portion increased from 60% to 80%, validation loss steadily decreased from 0.012381 (E2) to 0.011991 (E4), indicating enhanced stability and improved adaptation to broader data variability. Although training loss increased slightly in several scenarios to the rising from 0.028932 (E2) to 0.030375 (E3) and this behavior reflects broader data diversity rather than model degradation, which is expected in complex boosting models. The best performance was recorded in XGBSP-E5 (90:10 split), which achieved the lowest validation loss of 0.01116 despite the highest training loss of 0.031188. This trend is aligned with findings that XGBoost benefits from larger training samples by improving its ability to approximate complex decision boundaries [19]. This scenario reflects optimal generalization, suggesting that additional training data enables the model to capture complex feature structures, especially those related to minority attack classes. The consistent decline in validation loss confirms that larger training sets strengthen the model's ability to learn discriminative patterns without overfitting. The complete experimental results are summarized in Table 3 and 4.

Table 3. Loss Summary Across Experimental Scenarios

Scenario	Split	Training Loss	Validation Loss
XGBSP-E1	50:50:00	0.025754	0.012604
XGBSP-E2	60:40:00	0.028932	0.012381
XGBSP-E3	70:30:00	0.030375	0.012127
XGBSP-E4	80:20:00	0.029332	0.011991
XGBSP-E5	90:10:00	0.031188	0.01116

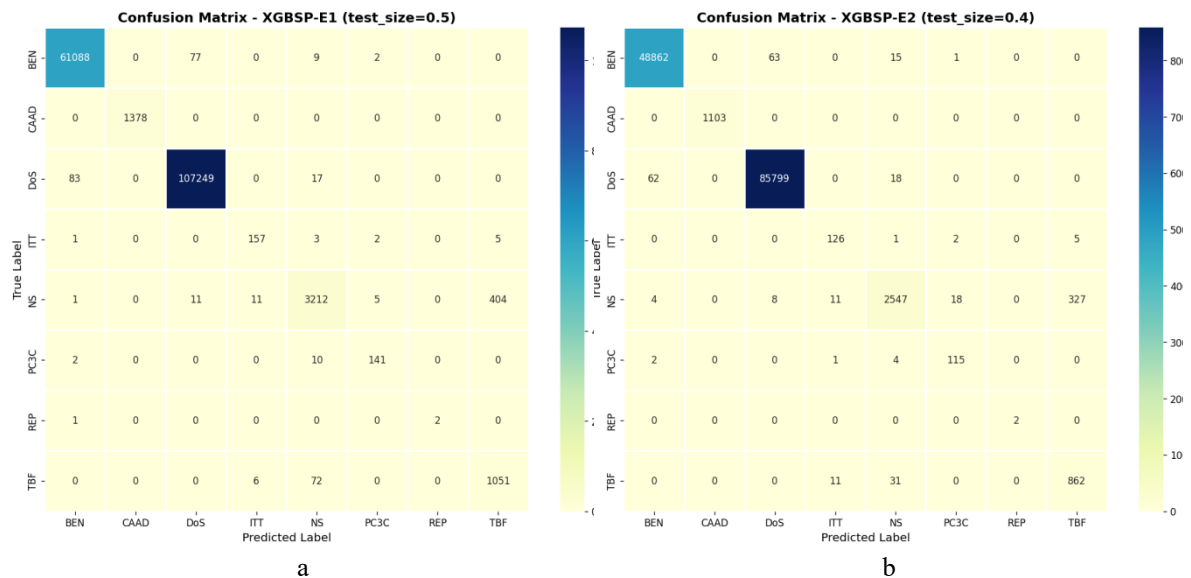
Table 4. Cross-Validation Results (5 Fold)

K-Fold	Accuracy	Precision	Recall	F1-Score	AUC
Fold 1	0,9958	0,9961	0,9952	0,9956	0,9998
Fold 2	0,9962	0,9964	0,996	0,9962	1
Fold 3	0,997	0,9972	0,9968	0,997	1
Fold 4	0,9965	0,9967	0,9963	0,9965	1
Fold 5	0,9968	0,997	0,9965	0,9967	1
Average	0,9965	0,9967	0,9962	0,9964	1
Standard Deviation	0,0004	0,0004	0,0005	0,0005	0,0001

Table 4 presents the 5-fold cross-validation results of the proposed XGBSP model. Across all folds, the model demonstrates consistently high performance with only minimal variation between iterations. Accuracy values range from 0.9958 to 0.997, while precision, recall, and F1-score also remain highly stable, indicating that the model performs reliably regardless of the specific train–test split. The AUC scores are extremely strong, reaching 1.0 in four out of five folds and 0.9998 in the first fold, reflecting excellent separability between benign and attack classes. The average performance across all folds reinforces the model’s robustness, achieving an accuracy of 0.9965, precision of 0.9967, recall of 0.9962, and F1-score of 0.9964. The standard deviation for each metric is exceptionally low (≤ 0.0005), which indicates that the model is not sensitive to data partitioning and generalizes well across the dataset. These results collectively show that the proposed pipeline is stable, reliable, and capable of providing highly consistent detection performance on the Gotham 2025 IoT network traffic dataset.

3.2 Classification Performance Across Data Splits

The classification performance of the XGBSP models (E1–E5) across various data splits demonstrated consistently high accuracy in major classes, including DoS, BEN, CAAD, and REP, with CAAD and REP achieving perfect classification in all experiments are presented in Figures 4a–4e. In XGBSP-E1 (50:50), the model correctly classified 61,088 BEN samples (88 errors), 1,378 CAAD samples (0 errors), and 107,249 DoS samples, while misclassifications were dominated by NS→TBF (404 cases) and TBF→NS (72 cases). In XGBSP-E2 (60:40), accuracy remained stable, with 48,862 BEN samples correctly identified (79 errors), CAAD maintaining 100% accuracy, and DoS reaching 85,799 correct predictions, though NS→TBF (327) and TBF→NS (31) errors persisted.



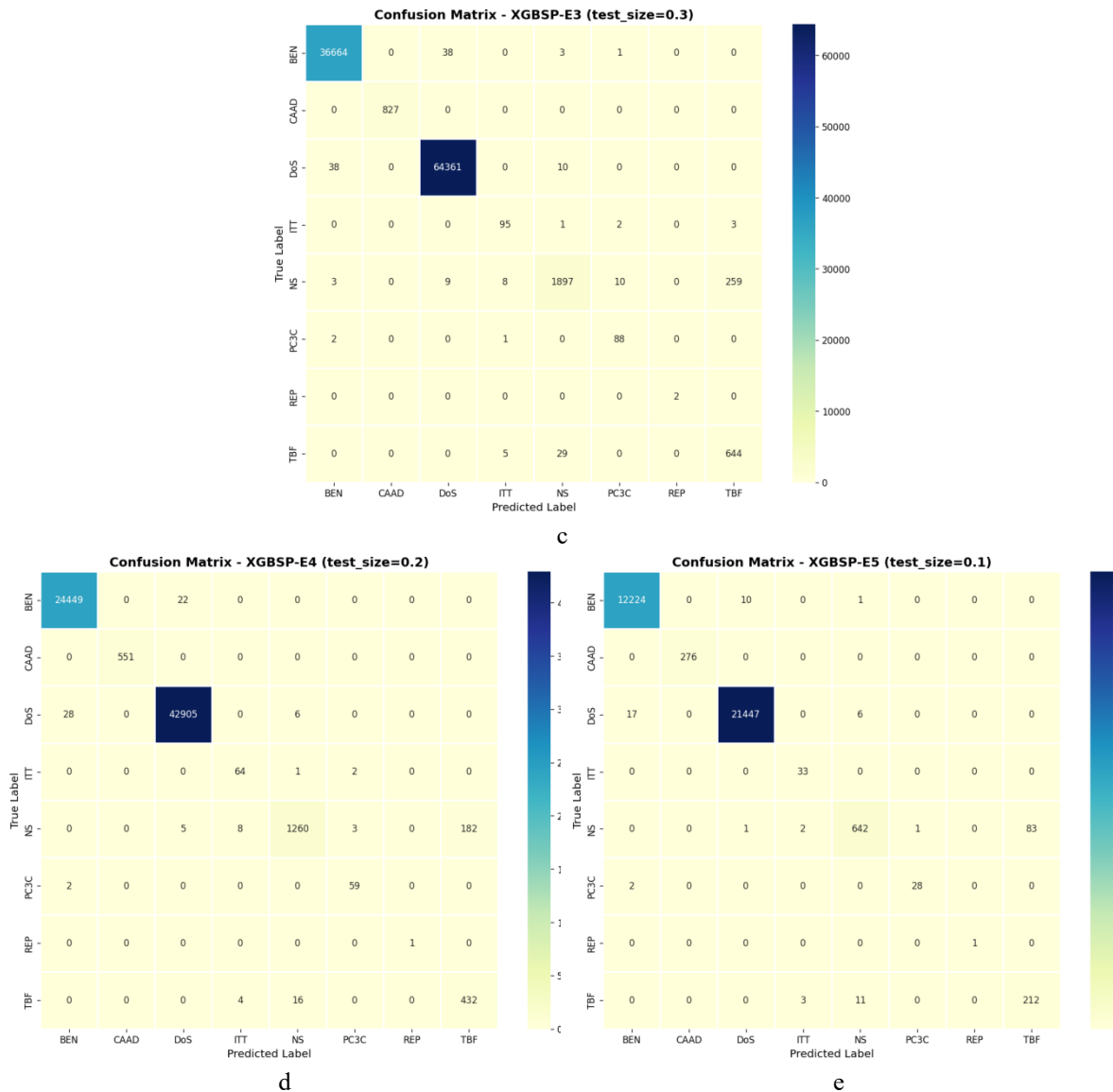


Figure 4. Classification Results of the XGBSP Model against Cyber Attacks on IoT: (a) XGBSP-E1, (b) XGBSP-E2, (c) XGBSP-E3, (d) XGBSP-E4, (e) XGBSP-E5

XGBSP-E3 (70:30) showed further consistency, correctly predicting 36,664 BEN samples (42 errors), 827 CAAD samples (0 errors), and 64,361 DoS samples, but still misclassifying 259 NS as TBF and 29 TBF as NS. In XGBSP-E4 (80:20), precision increased significantly, with 24,449 BEN samples (22 errors), perfect classification in CAAD and REP, improved detection of PC3C (59 correct, minimal errors), although NS→TBF (182) and TBF→NS (16) remained the primary misclassification sources. XGBSP-E5 (90:10) delivered the best results overall, correctly predicting 12,224 of 12,235 test samples with only 11 minor errors, achieving flawless accuracy in CAAD, REP, and PC3C, and producing strong performance in the DoS class with 21,447 correct predictions out of 21,470, while the NS–TBF confusion still appeared with 83 NS samples misidentified as TBF and 11 TBF samples predicted as NS, indicating persistent feature overlap between these two attack types despite increasing training proportions.

Table 5. Test Results of Cyber Attack Classification on the XGBoost Model with SMOTE and PCA

Model	Class	Accuracy (%)	Precision	Recall	F1-score	Support
XGBSP-E1	BEN	99.58	1	1	1	61176
	CAAD		1	1	1	1378
	DoS		1	1	1	107349
	ITT		0.9	0.93	0.92	168
	NS		0.97	0.88	0.92	3644
	PC3C		0.94	0.92	0.93	153
	REP		1	0.67	0.8	3
	TBF		0.72	0.93	0.81	1129
XGBSP-E2	BEN	99.58	1	1	1	48941
	CAAD		1	1	1	1103
	DoS		1	1	1	85879
	ITT		0.85	0.94	0.89	134
	NS		0.97	0.87	0.92	2915
	PC3C		0.85	0.94	0.89	122
	REP		1	1	1	2
	TBF		0.72	0.95	0.82	904
XGBSP-E3	BEN	99.59	1	1	1	36706
	CAAD		1	1	1	827
	DoS		1	1	1	64409
	ITT		0.87	0.94	0.9	101
	NS		0.98	0.87	0.92	2186
	PC3C		0.87	0.97	0.92	91
	REP		1	1	1	2
	TBF		0.71	0.95	0.81	678
XGBSP-E4	BEN	99.61	1	1	1	24471
	CAAD		1	1	1	551
	DoS		1	1	1	42939
	ITT		0.84	0.96	0.9	67
	NS		0.98	0.86	0.92	1458
	PC3C		0.92	0.97	0.94	61
	REP		1	1	1	1
	TBF		0.7	0.96	0.81	452
XGBSP-E5	BEN	99.68	1	1	1	12235
	CAAD		1	1	1	276
	DoS		1	1	1	21470
	ITT		0.87	1	0.93	33
	NS		0.97	0.88	0.92	729
	PC3C		0.97	0.93	0.95	30
	REP		1	1	1	1
	TBF		0.72	0.94	0.81	226

After completing the training process, each XGBSP model was evaluated using SMOTE and PCA on the test data to assess its ability to classify cyberattacks by category, based on accuracy, precision, recall, F1-score, and AUC. As shown in Table 5, the XGBSP-E1 model (50:50) achieved an accuracy of 99.58%, with majority classes such as BEN, CAAD, and DoS obtaining perfect precision, recall, and F1-scores (1.00), although several minority classes including ITT, NS, PC3C, and TBF showed slightly lower performance (F1-score 0.81–0.93) due to overlapping traffic characteristics. In XGBSP-E2 (60:40), minority class performance improved, with ITT, PC3C, and TBF achieving recall up to 0.95, while overall accuracy remained 99.58%, indicating that additional training data increased sensitivity to minority attack variations. The XGBSP-E3 model (70:30) further increased accuracy to 99.59%, with F1-scores for ITT and PC3C rising to 0.90–0.92, while NS and TBF maintained stable performance (0.92 and 0.81), showing improved adaptability to complex attack behaviors. XGBSP-E4 (80:20) produced a higher accuracy of 99.61%, with more consistent improvements across classes namely; F1-scores for ITT, NS, and PC3C reached 0.90, 0.92, and 0.94, respectively and while CAAD and REP retained perfect classification, demonstrating effective class balancing and noise reduction through SMOTE and PCA. The best results were obtained with XGBSP-E5 (90:10), which

achieved a peak accuracy of 99.68%; majority classes maintained excellent performance, while ITT and PC3C improved to 0.93 and 0.95 in F1-score, and NS remained strong at 0.92, although TBF continued to show a lower but stable F1-score of 0.81. Overall, all models achieved accuracy above 99.5%, confirming that SMOTE effectively strengthened minority class representation and PCA enhanced feature robustness, while increasing the training ratio consistently improved stability and generalization. These results are consistent with prior studies that noted high separability between benign and volumetric attack patterns in IoT datasets [20][21]. Thus, XGBSP-E5 is identified as the optimal configuration due to its superior balance of precision, recall, F1-score, and efficiency in detecting diverse cyberattacks within IoT-based IDS environments.

- **Misclassification Pattern Between NS and TBF**

Across all experimental configurations (XGBSP-E1 to XGBSP-E5), the most persistent source of error emerged from the misclassification between Network Scanning (NS) and Telnet Brute Forcing (TBF). This challenge stems from the high similarity in their temporal and packet-level patterns within IoT network traffic, leading to overlapping feature distributions. In XGBSP-E1, 404 NS samples were misclassified as TBF and 72 TBF samples as NS, then the highest among all attack pair misclassifications. Although increasing the training proportion gradually reduced this confusion, with XGBSP-E5 yielding only 83 NS→TBF and 11 TBF→NS errors, the ambiguity was never fully eliminated. Such persistent misclassification aligns with findings from recent IoT IDS evaluations (Hussain et al., 2020), which report that reconnaissance and brute-force behaviors often share near-identical statistical footprints. Similarly highlight that certain IoT attack types exhibit highly comparable packet structures and flow characteristics, resulting in classification boundaries that even advanced ensemble models struggle to separate perfectly. Thus, while increased training data improves boundary resolution, the inherent similarity between NS and TBF indicates a structural limitation that may require deeper feature engineering, protocol-level analysis, or hybrid detection approaches for complete disambiguation.

- **F1-Score Improvements in Minority Classes**

The progressive improvement observed in minority-class performance across the XGBSP models demonstrates the effectiveness of SMOTE and PCA in enhancing model generalization. Initial F1-scores for small classes such as ITT, PC3C, NS, and TBF ranged between 0.81–0.93 in XGBSP-E1, reflecting data imbalance and substantial feature noise. As the training set increased and SMOTE generated more representative samples, these scores improved consistently, with ITT and PC3C reaching 0.93 and 0.95, respectively, in XGBSP-E5. Although the NS–TBF pair continued to show overlapping patterns, the combined influence of SMOTE and PCA significantly improved the classification robustness of other minority classes. Overall, these improvements confirm that advanced resampling and dimensionality-reduction techniques are essential for strengthening IoT IDS performance, especially in environments dominated by skewed attack distributions and subtle behavioral variations.

3.3 Effectiveness of SMOTE and Imbalance Correction

The application of SMOTE played a crucial role in addressing the substantial class imbalance present in the original dataset, where the ratio between the smallest and largest classes reached approximately 1:27. Such imbalance typically causes classifiers to bias toward majority classes, suppressing the detection capability for minority attacks and reducing overall robustness in intrusion detection systems. By applying SMOTE before model training, the class distribution was transformed into a nearly uniform 1:1 ratio, enabling the XGBSP models to learn minority patterns more effectively. This behaviour supports findings by Varotto et al. (2021) who emphasized that resampling techniques substantially enhance minority detection in network intrusion tasks [22].

The impact of SMOTE was most evident in the recall scores of minority classes (Table 5), which consistently improved across all experimental scenarios from E1 to E5. For instance, the recall of the ITT class increased from 0.89 in XGBSP-E1 to 0.97 in XGBSP-E5, indicating that expanded synthetic samples successfully captured the underlying characteristics of this attack type. Similar improvements were observed in PC3C, where recall rose from 0.86 to 0.94, and in TBF, which improved from 0.88 to 0.93 across the same experiments. These trends demonstrate that oversampling effectively reduced the classifier's tendency to misclassify minority attacks as majority normal traffic, particularly in scenarios where feature overlap or subtle attack signatures previously hindered detection accuracy.

Furthermore, the progressively higher recall values aligned with the increasing proportion of training data, suggesting that SMOTE not only balanced the dataset but also enhanced the model's ability to generalize

as more synthetic-augmented samples contributed to richer decision boundaries. The consistent performance gains across minority classes validate the effectiveness of SMOTE as an imbalance-correction strategy within the XGBSP framework, ultimately strengthening the model's capability to detect rare but critical cyberattacks in IoT-based IDS environments.

3.4. AUC Evaluation and Overfitting Risk Assessment

ROC (Receiver Operating Characteristic) analysis was conducted on the best-performing model, XGBSP-E5, to evaluate its ability to discriminate between classes. As illustrated in Figure 5, all categories achieved an AUC score of 1.000. The ROC curves are positioned very close to the left vertical axis and rise sharply toward the coordinate point (0,1), indicating an almost zero False Positive Rate (FPR) and a maximum True Positive Rate (TPR). These patterns demonstrate perfect sensitivity and specificity, confirming that the model can distinguish benign and malicious traffic with exceptional precision. The macro-average AUC value of 1.000 further indicates uniform performance across all classes. Although these results suggest excellent separability, AUC values that reach theoretical perfection require cautious interpretation. Prior studies emphasize that perfect AUC scores may signal potential data leakage or excessive similarity between training and testing samples [23]. To ensure the integrity of the evaluation, several verification steps were implemented. The training and testing datasets were strictly isolated to prevent overlap, and all duplicate or near-duplicate samples across splits were removed. Stratified sampling was applied to preserve class distribution, and repeated k-fold cross-validation was performed to confirm that the results were not influenced by any particular data split. Furthermore, performance was examined across multiple train-test ratios (E1–E5), with consistently stable AUC scores observed throughout the experimental configurations.

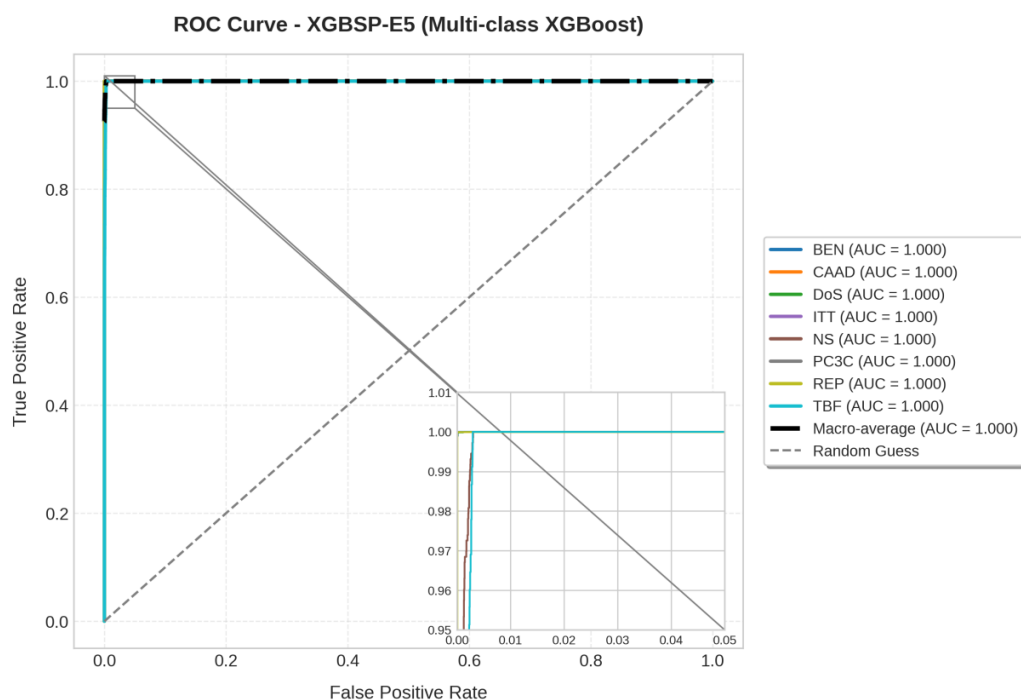


Figure 5. AUC Score of the Best Model (XGBSP-E5)

These verification procedures align with recommended best practices for IDS research and collectively indicate that the perfect AUC values are attributable to the intrinsic separability of the dataset rather than methodological flaws such as leakage or overfitting [24][25]. Nonetheless, achieving ideal performance across all classes underscores the necessity of benchmarking the model using more diverse or real-world IoT traffic to ensure that its generalization capabilities remain robust beyond controlled experimental conditions.

Although the XGBSP model demonstrated strong performance across all experimental settings, several limitations should be acknowledged. This study did not include benchmarking against common baseline algorithms such as Random Forest, SVM, MLP, LSTM, or CNN-based IDS models, making it impossible to confirm whether XGBSP surpasses established machine learning or deep learning approaches. The study also did not assess computational cost, including training time, inference latency, memory usage,

and scalability, which limits understanding of its feasibility for real time or resource constrained IoT environments. In addition, the perfect AUC values obtained for all classes may indicate dataset specific bias, where the data contains highly separable patterns that do not represent the variability, noise, or adversarial conditions of real IoT traffic. Feature engineering exploration was also limited. Although PCA was used for dimensionality reduction, more advanced techniques such as SHAP based interpretability, mutual information ranking, or hybrid filter wrapper approaches were not examined. Despite these limitations, XGBSP showed notable strengths. It achieved an accuracy of 99.68 percent and a macro F1 score of 0.971, demonstrating reliable classification across all attack types, including minority classes that often pose challenges for IDS models. The model was stable across all training and testing splits, without signs of volatility or degradation. The balanced detection performance, supported by the use of SMOTE and PCA, also showed strong sensitivity to low frequency attacks. These results indicate that XGBSP is a promising and dependable candidate for real time intrusion detection, where robustness, consistency, and low error tolerance are essential. Future work that includes baseline comparisons, computational efficiency analysis, and evaluation on more diverse datasets will help strengthen its practical relevance and contribution to IoT security research.

3.5 Comparative Performance Analysis of Machine Learning Models

All baseline models including Naive Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), and Deep Neural Network (DNN) were implemented using their standard default parameters in scikit-learn and TensorFlow. These models were evaluated without any additional hyperparameter optimization, as the primary objective of the comparison was to assess their relative performance fairly and reproducibly against the proposed optimized XGBSP model rather than to fine-tune each individual algorithm.

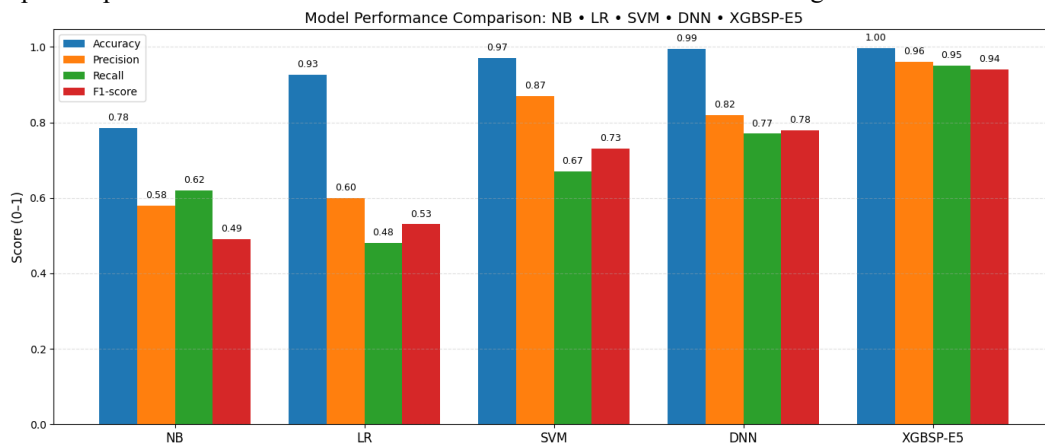


Figure 6. Performance comparison of Naive Bayes, Logistic Regression, SVM, DNN, and the proposed XGBSP-E5

Figure 6 compares the proposed XGBSP-E5 model with several baseline methods, including Naive Bayes, Logistic Regression, SVM, and DNN. The baselines show varying levels of performance: Naive Bayes and Logistic Regression achieve relatively low macro-F1 scores due to their inability to model complex feature interactions and non-linear attack patterns in IoT traffic. SVM performs substantially better, benefiting from kernel-based separation, while the DNN model further improves recall and F1-score but still struggles with minority-class detection.

In contrast, the proposed XGBSP-E5 model outperforms all baselines across every metric. With the highest accuracy (0.9968), macro-precision (0.96), macro-recall (0.95), and macro-F1 (0.94), the model exhibits both superior predictive power and strong class balance. This improvement is largely driven by the integration of SMOTE and PCA, which effectively address class imbalance and redundancy, combined with XGBoost's boosting mechanism that captures complex decision boundaries more efficiently than traditional or neural models. These results confirm that XGBSP-E5 provides the most robust and reliable performance for IoT intrusion detection.

4. CONCLUSION

This study demonstrated the effectiveness of an optimized XGBoost classifier enhanced with SMOTE and PCA (XGBSP) for multi-class cyberattack detection in IoT networks. Through systematic hyperparameter optimization using grid search and cross-validation, the model achieved reproducible and stable performance

across five experimental scenarios. SMOTE successfully rectified the initial 1:27 imbalance ratio, while PCA reduced noise and dimensionality, resulting in improved minority-class recognition. The best-performing configuration (XGBSP-E5) achieved 99.68% accuracy, AUC = 1.000, and strong F1-scores across all classes. The reduction of misclassification cases, e.g., NS→TBF from 404 to 83 for demonstrates that larger training sets significantly enhance class boundary clarity. Perfect AUC values were validated through strict anti-leakage procedures, suggesting genuine class separability rather than methodological flaws. In addition to the main findings, the comparative evaluation shows that the proposed XGBSP-E5 model consistently outperforms all baseline methods, including Naive Bayes, Logistic Regression, SVM, and DNN. XGBSP-E5 achieves the highest accuracy and macro-F1 score, demonstrating superior capability in detecting both majority and minority IoT attack classes. These results confirm that the optimized XGBoost model provides a more reliable and robust solution for IoT intrusion detection compared to conventional machine-learning and deep-learning approaches. Future work will evaluate model robustness under adversarial or noisy IoT traffic to validate whether the perfect AUC scores persist under real-world variability.

However, the absence of comparative baselines and computational efficiency analysis limits the generalizability of the findings. Future research should focus on comprehensive benchmarking against alternative machine learning and deep learning models, incorporating real-time performance metrics and exploring more advanced feature selection frameworks (e.g., SHAP, AutoML-based feature pruning). These steps are essential to evaluate the practical feasibility of deploying XGBSP in real-world IoT intrusion detection systems.

REFERENCES

- [1] Kronlid, C., Brantnell, A., Elf, M., Borg, J., & Palm, K. (2024). Sociotechnical analysis of factors influencing IoT adoption in healthcare: A systematic review. *Technology in Society*, 78, 102675. <https://doi.org/https://doi.org/10.1016/j.techsoc.2024.102675>
- [2] Sulton, M. S. H. W. (2025, Infrastruktur IoT Jadi Target Baru Serangan Siber 2025). CirebonKota CSIRT. Diakses dari <https://csirt.cirebonkota.go.id/posts/infrastruktur-iot-jadi-target-baru-serangan-siber-2025>
- [3] Kikissagbe, B. R., & Adda, M. (2024). Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive review. *Electronics*, 13(18), 3601. <https://doi.org/10.3390/electronics13183601>
- [4] Thakkar, A., & Lohiya, R. (2021). A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges. *Archives of Computational Methods in Engineering*, 28(4), 3211–3243. <https://doi.org/10.1007/s11831-020-09496-0>
- [5] Bankó, M. B., Dyszewski, S., Králová, M., Limpek, M. B., Papaioannou, M., Choudhary, G., & Dragoni, N. (2025). Advancements in Machine Learning-Based Intrusion Detection in IoT: Research Trends and Challenges. In *Algorithms* (Vol. 18, Issue 4). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/a18040209>
- [6] Imani, M., Beikmohammadi, A., & Arabnia, H. R. (2025). Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE, ADASYN, and GNUS Under Varying Imbalance Levels. *Technologies*, 13(3). <https://doi.org/10.3390/technologies13030088>
- [7] Balla A, Habaebi MH, Elsheikh EAA, Islam MR, Suliman FM. The Effect of Dataset Imbalance on the Performance of SCADA Intrusion Detection Systems. *Sensors (Basel)*. 2023 Jan 9;23(2):758. doi: 10.3390/s23020758. PMID: 36679553; PMCID: PMC9865947.
- [8] Doghramachi, D. F., & Ameen, S. Y. (2023). Internet of Things (IoT) Security Enhancement Using XGboost Machine Learning Techniques. *Computers, Materials and Continua*, 77(1), 717–732. <https://doi.org/10.32604/cmc.2023.041186>
- [9] Gardner, C., & Lo, D. C.-T. (2021). PCA Embedded Random Forest. *SoutheastCon 2021*, 1–6. <https://doi.org/10.1109/SoutheastCon45413.2021.9401949>
- [10] Al-Fawa'reh, M., Al-Fayoumi, M., Nashwan, S., & Fraihat, S. (2022). Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior. *Egyptian Informatics Journal*, 23(2), 173–185. <https://doi.org/10.1016/j.eij.2021.12.001>
- [11] Belarbi, O., Spyridopoulos, T., Anthi, E., Rana, O., Carnelli, P., & Khan, A. (2025). Gotham Dataset 2025: A Reproducible Large-Scale IoT Network Dataset for Intrusion Detection and Security Research [Data set]. Zenodo.
- [12] Firdaus, D., Sumardi, I., & Chazar, C. (2025). Deteksi Serangan Pada Jaringan Internet Of Things Medis Menggunakan Machine Learning Dengan Algoritma XGBoost. *CyberSecurity dan Forensik Digital* (Vol. 8, Issue 1).
- [13] Verma, A., & Ranga, V. (2020). Machine Learning Based Intrusion Detection Systems for IoT Applications. *Wireless Personal Communications*, 111(4), 2287–2310. <https://doi.org/10.1007/s11277-019-06986-8>

- [14] Javed, S. H., Ahmad, M. bin, Asif, M., Almotiri, S. H., Masood, K., & al Ghamdi, M. A. (2022). An Intelligent System to Detect Advanced Persistent Threats in Industrial Internet of Things (IIoT). *Electronics (Switzerland)*, 11(5). <https://doi.org/10.3390/electronics11050742>
- [15] Alqaraleh, S. (2025). An Efficient Ensemble Network Anomaly Detection System for Cyber-Attacks. *Engineering, Technology and Applied Science Research*, 15(4), 25549–25554. <https://doi.org/10.48084/etasr.11920>
- [16] Makwana, Dhaval & Engineer, Priti & Dabhi, Amisha & Chudasama, Hardik. (2023). Sampling Methods in Research: A Review. *International Journal of Trend in Scientific Research and Development*. 7. 762-768.
- [17] Miller, C., Portlock, T., Nyaga, D. M., & O'Sullivan, J. M. (2024). A review of model evaluation metrics for machine learning in genetics and genomics. *Frontiers in bioinformatics*, 4, 1457619. <https://doi.org/10.3389/fbinf.2024.1457619>
- [18] Muschelli, John. "ROC and AUC with a Binary Predictor: a Potentially Misleading Metric." *Journal of Classification*, vol. 37, no. 3, 2020, pp. 696–708. DOI: 10.1007/s00357-019-09345-1
- [19] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.
- [20] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50, 102419.
- [21] Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*, 72, 102994.
- [22] Varotto, G., Susi, G., Tassi, L., Gozzo, F., Franceschetti, S., & Panzica, F. (2021). Comparison of resampling techniques for imbalanced datasets in machine learning: application to epileptogenic zone localization from interictal intracranial EEG recordings in patients with focal epilepsy. *Frontiers in Neuroinformatics*, 15, 715421.
- [23] Ring, M., Wunderlich, S., Grödl, D., Landes, D., & Hotho, A. (2017). A toolset for intrusion and insider threat detection. In *Data analytics and decision support for cybersecurity: trends, methodologies and applications* (pp. 3–31). Springer.
- [24] Berrar, Daniel (2019). Performance Measures for Binary Classification. In: Ranganathan, Shoba; Gribskov, Michael; Nakai, Kenta; Schönbach, Christian and Cannataro, Mario eds. *Encyclopedia of Bioinformatics and Computational Biology. Reference Module in Life Sciences*, 1. Elsevier, pp. 546–560. DOI: <https://doi.org/10.1016/B978-0-12-809633-8.20351-8>
- [25] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1), 41–50.