İSTANBUL UNIVERSITY PRESS

Acta Infologica

Acta Infologica,
ACIN 2025, 9 (2): 682–701

https://doi.org/10.26650/acin.1798154
Submitted: 06.10.2025
Revision Requested: 18.11.2025
Last Revision Received: 01.12.2025
Accepted: 07.12.2025

Research Article

Open Access

Detecting Robotic Cyber Attacks in Robot Operating System Networks

Hamdullah Karamollaoğlu [1]

[1] Electricity Generation Corporation, Ankara, Türkiye

Abstract The increasing deployment of robotic systems in critical sectors such as manufacturing, healthcare, and infrastructure necessitates robust cybersecurity measures. The Robot Operating System (ROS), a core middleware in modern robotics, is inherently susceptible to cyber threats due to its lack of integrated security mechanisms. This study presents a comprehensive benchmark evaluation of intrusion detection solutions for ROS-based environments. Utilizing the novel ROSIDS23 dataset, which includes realistic attack scenarios—such as Denial-of-Service (DoS), Unauthorized Publish, Unauthorized Subscribe, and Subscriber Flood—we rigorously evaluated and compared fifteen state-of-the-art Machine Learning (ML) and Deep Learning (DL) models. Using stratified 5-fold cross-validation, our results demonstrate that ensemble methods significantly outperform deep learning approaches in this context. Gradient Boosting achieved the highest performance, with 99.80% accuracy, precision, recall, and F1-score, followed by Light Gradient Boosting Machine (LightGBM) at 99.51% and Extreme Gradient Boosting (XGBoost) at 99.48%. Among DL models, the best-performing One-Dimensional Convolutional Neural Network (1D-CNN) reached 98.55%. Beyond overall metrics, we examine per-class performance, confusion matrices, and Receiver Operating Characteristic (ROC) curves, highlighting model-specific strengths and weaknesses, particularly in detecting minority attack classes.

Keywords Robot operating system · cyber attacks · machine learning · deep learning · robotic cybersecurity

Citation: Karamollaoğlu, H. (2025). Detecting robotic cyber attacks in robot operating system networks. *Acta Infologica*, *9*(2), 682-701. https://doi.org/10.26650/acin.1798154

Corresponding author: Hamdullah Karamollaoğlu h.karamollaoglu@euas.gov.tr

Introduction

Robotics has rapidly evolved into a cornerstone of technological progress in the 21st century, revolutionizing industries such as healthcare, manufacturing, defense, logistics, and domestic services. Adopting robotic systems has enabled automation, enhanced precision, and reduced human intervention in complex and hazardous environments. However, this increasing reliance on robots has also introduced significant cybersecurity concerns. Unlike conventional IT systems, robots interact with the physical world, meaning that security breaches can cause data loss or financial damage and safety-critical failures, with direct consequences for human life. This dual cyber-physical nature makes robotic platforms uniquely vulnerable to sophisticated attacks, thereby elevating the importance of securing their underlying software and communication infrastructures. The Robot Operating System (ROS) has emerged as the de facto middleware framework for robotic application development. Its widespread adoption is primarily attributed to its modular design, rich set of libraries, and strong community support, all of which have significantly accelerated prototyping, experimentation, and real-world deployment of robotic systems.

Nevertheless, their widespread adoption has also attracted adversaries seeking to exploit security weaknesses. Previous research has demonstrated that ROS deployments are vulnerable to various threats, ranging from denial-of-service and eavesdropping to malicious node injection and command manipulation. Such attacks can disrupt robotic functionality, compromise data integrity, or even lead to catastrophic physical outcomes (Ahmad Yousef et al., 2018; Monoscalco et al., 2022).

Traditional cybersecurity approaches have been widely applied in Information Technology (IT) and Internet of Things (IoT) environments to counter cyber threats. However, direct application of these approaches to robotic systems has proven insufficient. This limitation arises from several unique characteristics of robotic middleware, summarized as follows:

- Distinct communication paradigms such as publisher-subscriber topics, service calls, and action interfaces differ fundamentally from conventional client-server protocols.
- Real-time constraints where delays or false alarms in detection could impair robotic control and safety.
- Heterogeneity of robotic applications spans from industrial arms to autonomous vehicles, each with distinct traffic patterns.

Furthermore, the lack of domain-specific benchmark datasets has hindered the development of effective cybersecurity solutions for ROS. Although generic network intrusion datasets such as NSL-KDD, CICIDS2017, and UNSW-NB15 are widely used in cybersecurity research, they do not adequately capture the semantics of robotic communication or the threats unique to robotic environments. Only a few studies have attempted to build security frameworks for ROS, and simulation-based traffic, narrow attack categories, or the absence of rigorous comparative evaluations often constrain these. As a result, the field continues to lack systematic benchmarking of machine learning (ML) and deep learning (DL) models on robotics-specific datasets (Botta et al., 2023; Martín et al., 2018).

The ROSIDS23 dataset (Değirmenci et al., 2023) represents a significant step forward in this context. As a publicly available dataset designed for intrusion detection in ROS-based environments, it provides a foundation for addressing the aforementioned gaps. ROSIDS23 contains labeled traffic data reflecting real-world robotic communication patterns and multiple attack scenarios, enabling reproducible and comparative evaluation of security models. This study undertakes an extensive comparative analysis of ML and modern DL models on the ROSIDS23 dataset for enhancing the cybersecurity of robotic systems. By focusing on

intrusion detection as a critical first line of defense, this work provides actionable insights for securing robotic middleware. The main contributions of this work are summarized as follows:

- Fifteen state-of-the-art ML and DL models were systematically evaluated on the domain-specific ROSIDS23 dataset. The evaluated models encompass ensemble methods (Gradient Boosting, XGBoost, LightGBM, Random Forest (RF), Categorical Boosting (CatBoost), Extremely Randomized Trees (Extra-Trees)), traditional classifiers (Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, Logistic Regression (LR), Gaussian Naive Bayes (Gaussian NB), Adaptive Boosting (AdaBoost)), and Deep Learning architectures (Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), One-Dimensional Convolutional Neural Network (1D-CNN)), providing a broad comparative analysis across algorithmic paradigms.
- The study employed a rigorous preprocessing pipeline including label encoding, feature standardization, KNN imputation for missing values, outlier handling, and stratified data partitioning. Stratified 5-fold cross-validation was used to ensure the reliability and generalizability of model performance estimates.
- Beyond standard accuracy metrics, model performance was assessed using precision, recall, F1-score, ROC-AUC, and computational efficiency (training and inference times). This holistic evaluation identified predictive effectiveness and practical feasibility for real-time deployment in ROS environments, including accurately detecting minority attack classes.
- Gradient Boosting, LightGBM, and XGBoost were identified as the most effective algorithms, achieving near-perfect detection of DoS, Subflood, Unauthorized Publish, and Unauthorized Subscribe attacks. Detailed confusion matrices and per-class performance analyses provide insight into model strengths and limitations, particularly for subtle attack patterns like UnauthSub.
- The study highlights the trade-offs between model accuracy and computational requirements, demonstrating that LightGBM and XGBoost offer an optimal balance for edge deployment. At the same time, Gradient Boosting serves as the benchmark for offline high-accuracy training. These insights support designing practical, deployable intrusion detection systems for robotic networks.
- By offering a reproducible benchmark using the ROSIDS23 dataset, this study establishes a reference framework for future research, promoting the exploration of hybrid cybersecurity architectures, real-time validation in robotic testbeds, and the creation of larger-scale ROS security datasets.

The remainder of this paper is organized as follows. Section II reviews existing research on robotic cybersecurity, focusing on ROS. Section III details the dataset and the methodological framework, including data preprocessing, model selection, and evaluation procedures. Section IV presents and analyzes the experimental results. Finally, Section V concludes the study by summarizing the key findings and outlining potential avenues for future research, including developing hybrid cybersecurity architectures, real-time validation within robotic testbeds, and creating larger-scale robotic security datasets.

Related Work

The cybersecurity of robotic and cyber-physical systems (CPS) has received increasing attention due to the widespread deployment of robots in industrial, healthcare, and infrastructure environments. Verma et al. (Verma, Kumar, Sheikh, et al., 2025) highlight the role of ML in enhancing CPS security, analyzing architectures, communication protocols, historical attacks, and vulnerabilities, while emphasizing the development of reliable ML-based security solutions. Similarly, Tanimu et al. (Tanimu & Abada, 2025) provide a compre-

hensive overview of cybersecurity challenges in robotics, focusing on deploying network-based intrusion detection systems (NIDS) and discussing methodologies such as ML, DL, and hybrid systems to improve detection accuracy, adaptability, and real-time response.

Unique security risks necessitate tailored solutions within the Internet of Robotic Things (IoRT). Raza et al. (Raza et al., 2024) examine ML-based techniques to mitigate IoRT vulnerabilities, presenting robot classifications and the three-layered IoRT architecture alongside various protective measures. Krejčí et al. (Krejčí et al., 2025) further discuss IoRT integration with IoT technologies to enhance efficiency and autonomy across sectors like healthcare and agriculture, while addressing challenges including data security, energy efficiency, and ethical concerns.

ROSs introduce specific challenges due to their publisher-subscriber communication model and distributed architecture. Zafar et al. (Zafar et al., 2024) propose hybrid neural network models combining 1D convolutional neural networks with multi-head attention mechanisms to improve intrusion detection performance in high-volume, high-speed robotic data streams. Degirmenci et al. (Değirmenci et al., 2024) explore Robot-NIDS architectures and novel adversarial attack detection methods, incorporating reconstruction errors with aleatoric, epistemic, and entropy metrics to enhance robustness against attacks such as FGSM, PGD, and BIM. Kang et al. (Kang et al., 2025) focus on offline anomaly detection in ROS, highlighting the need for comprehensive mechanisms to address performance irregularities in callback execution, while Değirmenci et al. (Degirmenci et al., 2023) provide ROSIDS23, a specialized dataset to support IDS evaluation and the evolution of resilient robotic infrastructures.

Beyond ROS-specific studies, Bezemskij et al. (Bezemskij et al., 2016) developed lightweight IDS mechanisms for robotic vehicles, emphasizing energy-efficient local and offloading detection. Pu et al. (Pu et al., 2022) and Holdbrook et al. (Holdbrook et al., 2024) discuss security vulnerabilities, attacks, and mitigation strategies in industrial robots, analyzing NIDS methodologies and integration with emerging technologies such as federated learning and large language models. Sharma et al. (Sharma et al., 2024) review security applications for mobile robots in remotely operated reactors, highlighting their versatility, performance, and interaction capabilities.

Comprehensive reviews by Yaacoub et al. (Yaacoub et al., 2022) and Verma et al. (Verma, Kumar, Verma, et al., 2025) further synthesize robotic cybersecurity mechanisms, discussing vulnerabilities, countermeasures, communication protocols, and frameworks leveraging ML, encryption, and blockchain for secure design, access control, and authentication. Tsapin et al. (Tsapin et al., 2024) also contribute to video-based security in robotic environments, enhancing recognition accuracy and algorithmic robustness under varying conditions. These studies underline the critical role of hybrid ML/DL techniques, adversarial robustness, specialized datasets, and system-specific IDS frameworks in securing complex robotic and CPS environments.

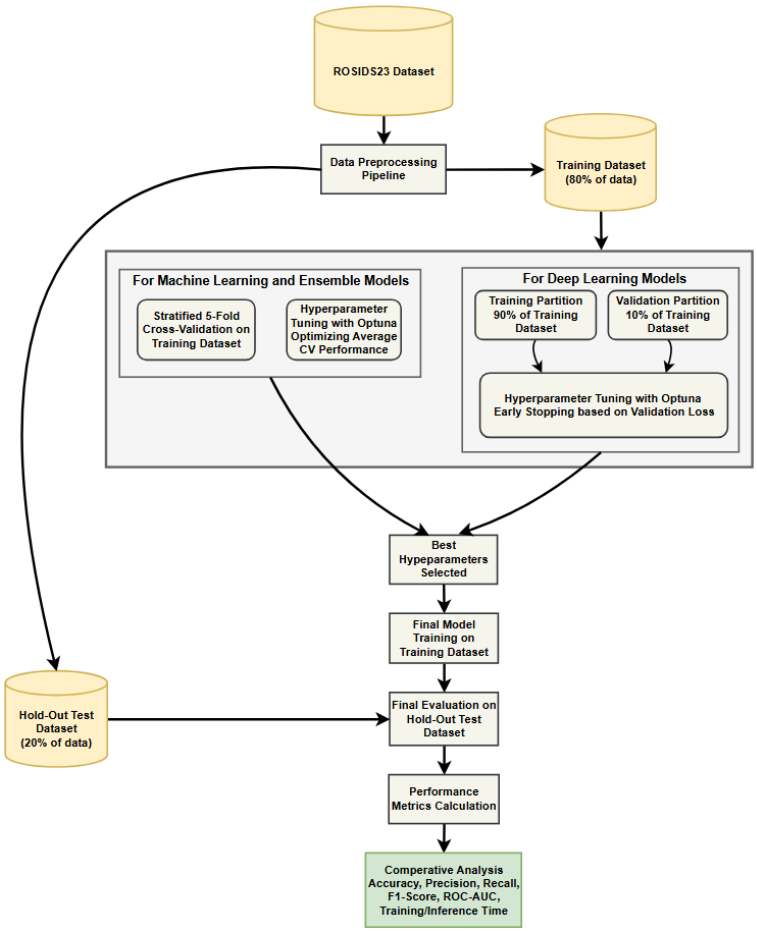
Notwithstanding these contributions, a critical synthesis of the existing literature reveals persistent limitations that hinder the advancement and systematic benchmarking of effective intrusion detection systems for ROS ecosystems. A significant obstacle is the lack of a standardized, domain-specific benchmark dataset that faithfully captures the publisher–subscriber communication semantics and authentic attack vectors characteristic of ROS. Many prior studies rely on generic network intrusion datasets (e.g., NSL-KDD (Dhanabal & Shantharajah, 2015), CICIDS2017 (Panigrahi & Borah, 2018)), which do not reflect the structural and semantic nuances of robotic network traffic, thereby producing models with limited generalizability to operational robotic environments. In addition, the field suffers from a scarcity of comprehensive comparative evaluations that span the breadth of ML and DL paradigms. Although several studies propose

novel architectures, they often neglect rigorous comparisons with state-of-the-art ensemble methods and traditional classifiers under a unified, realistic benchmark, obscuring the determination of this domain's most effective algorithmic approach. Another critical limitation lies in the insufficient consideration of practical constraints intrinsic to robotic systems, such as the necessity of low-latency inference for real-time operation and computational efficiency suitable for deployment on resource-constrained edge devices.

This study is designed to address these scholarly gaps and provide a substantive, transferable contribution to the field. To overcome the dataset deficiency, we employ the recently introduced ROSIDS23 dataset, the first publicly available benchmark purposefully constructed for ROS intrusion detection, encompassing both legitimate traffic and realistic attack scenarios (e.g., Denial-of-Service, Unauthorized Publish/Subscribe, Subscriber Flood). To fill the gap in holistic benchmarking, we conduct a rigorous empirical evaluation of fifteen state-of-the-art models, including ensemble methods, traditional classifiers, and DL architectures, using stratified 5-fold cross-validation to ensure statistically robust performance estimates. Finally, to bridge the divide between theoretical performance and practical deployment, our analysis extends beyond conventional accuracy-based metrics to incorporate computational efficiency, explicitly measuring training time and inference latency, thereby offering crucial insights into the trade-off between detection effectiveness and resource consumption.

Figure 1

Workflow of the proposed methodology for cybersecurity threat detection



Methodology

This study adopted a rigorous experimental methodology encompassing data collection, preprocessing, model selection, and evaluation to establish a comprehensive benchmark for cybersecurity threat detection in ROS environments. The research design was structured to ensure a fair and reproducible comparison between traditional ML and DL approaches, with particular attention to the unique characteristics of robotic network traffic, as illustrated in Figure 1.

As shown in Figure 1, the overall workflow of the proposed framework presents the sequential steps followed in the study. In this framework, the ROSIDS23 dataset was utilized as the primary data source, and a dedicated preprocessing pipeline was applied to clean, transform, and prepare the dataset for model training and evaluation. For ML and ensemble models, stratified 5-fold cross-validation was employed, followed by hyperparameter optimization with Optuna to maximize the average cross-validation performance. In contrast, the training dataset was further partitioned for DL models into training and validation subsets, where hyperparameter tuning was carried out with Optuna, and early stopping was applied based on validation loss. The best-performing hyperparameters obtained from the optimization process were then selected, and final model training was conducted using the complete training dataset. Subsequently, the trained models were evaluated on a hold-out test dataset comprising 20% of the data. Performance assessment was done using multiple evaluation metrics, including Accuracy, Precision, Recall, F1-Score, ROC-AUC, and training/inference time. Finally, a comparative analysis was performed to highlight ML and DL models' relative strengths and weaknesses.

Dataset

The efficacy of any cybersecurity solution is fundamentally dependent on the quality and relevance of the data used for its development and validation. Recognizing the critical shortcoming of existing network intrusion datasets—their inability to accurately represent the unique communication paradigms and threat landscape of robotic systems—this study employs the ROSIDS23 dataset. This dataset was explicitly created to benchmark cybersecurity solutions within ROS environments, a prevalent middleware in modern robotics.

The dataset was generated by orchestrating a realistic robotic simulation scenario, encompassing both regular operational traffic and malicious activities designed to exploit known vulnerabilities in ROS. The dataset used in this study contains 136,681 samples, each described by a total of 85 flow-based network features. These features include flow identifiers (e.g., Flow ID, Src IP, Dst IP), transport-layer attributes such as Src Port, Dst Port, and Protocol, as well as timestamp and duration information. The dataset further incorporates detailed packet-level statistics (Tot Fwd Pkts, Tot Bwd Pkts, TotLen Fwd/Bwd Pkts), packet length distribution metrics, inter-arrival time (IAT) characteristics, header lengths, and various flag counts (e.g., PSH, URG, SYN). It also includes higher-level behavioral indicators, such as active and idle time descriptors. These variables consist of integer, floating-point, and categorical fields, and the final feature corresponds to the class label. The attack scenarios were meticulously crafted based on documented threats and include:

- Denial-of-Service (DoS): Attacks aimed at disrupting the availability of critical ROS nodes and services, halting robotic functionality.
- Unauthorized Publish: Scenarios where a malicious node successfully injects spurious or malicious messages into a topic stream without proper authorization, potentially leading to erroneous actuator commands.

- Unauthorized Subscribe: Attacks involving an illegitimate node subscribing to a topic to eavesdrop on sensitive data streams, compromising data confidentiality.
- Subscriber Flood: An attack where a node creates excessive subscribers to a topic, consuming system resources and potentially leading to a service degradation or crash.

Network traffic was captured during these activities and processed into a structured tabular format. Each record in the dataset represents a snapshot of network activity, characterized by a comprehensive set of features (e.g., packet counts, timing statistics, protocol flags) and a categorical label indicating the class of traffic (Benign, DoS, Unauthorized_Publish, etc.) (Değirmenci et al., 2023). The domain-specific nature of ROSIDS23, which accurately mirrors the publisher-subscriber semantics and real-time constraints of robotic middleware, makes it a superior alternative to generic datasets like NSL-KDD or CICIDS2017 for this research context. Its structure facilitates a robust and reproducible benchmarking process for various ML and DL models.

Feature Engineering and Data Preparation

The dataset utilized in this study comprises a combination of continuous numerical and categorical features, requiring a comprehensive preprocessing framework to ensure compatibility with ML and DL models. Categorical attributes, including protocol types and network flag indicators, were transformed using the LabelEncoder utility from scikit-learn, which assigns a unique integer to each category. For the target variable (Label), a dedicated mapping was applied to accommodate the multiclass classification task: Benign → 0, Denial-of-Service (DoS) → 1, Subscriber Flood (Subflood) → 2, Unauthorized Publish (UnauthPub) → 3, Unauthorized Subscribe (UnauthSub) → 4. This encoding preserves categorical distinctions while enabling efficient numerical computation. Exploratory analysis revealed strong correlations among flow-based metrics, such as Total Forward Packets (Tot Fwd Pkts) and Total Backward Packets (Tot Bwd Pkts), with derived features including Flow Duration and Flow Bytes per Second (Flow Byts/s), which are critical for distinguishing regular traffic from various attack scenarios. Although label encoding introduces an implicit ordinal structure, its impact is mitigated by the predominant use of tree-based ensemble models, which are robust to such ordinal assumptions. To ensure robust model development and evaluation, a stratified 80/20 split was applied to divide the dataset into training and test subsets, preserving the original class distributions. The training set was subdivided to create a validation set comprising 10% of the original dataset. This three-way partitioning enabled hyperparameter optimization, early stopping in DL models, model selection, and regularization validation, while reserving the test set for unbiased final evaluation. Stratified sampling ensured consistent representation of all five classes across training, validation, and test partitions.

Feature standardization was applied to numerical attributes using the StandardScaler from scikit-learn to address scale discrepancies among features. Metrics such as Flow Duration, Tot Fwd Pkts, Tot Bwd Pkts, Flow Byts/s, Flow Pkts/s, Forward and Backward Packet Length statistics, and Inter-Arrival Times (IAT) were rescaled to have zero mean and unit variance. Scaling parameters were computed exclusively from the training set to prevent data leakage and applied to the validation and test sets, while previously encoded categorical features remained unchanged. Standardization ensures that features with inherently larger numerical ranges do not dominate model optimization, improving convergence and generalization in distance-based algorithms (e.g., KNN, SVM), gradient-based optimizers (e.g., Neural Networks, LR), and regularized linear models. The dataset was thoroughly examined for missing values and outliers before model training. The Flow Bytes per Second (Flow Byts/s) feature was observed to contain 272 missing values,

representing a negligible proportion (0.2%) of the total 136,681 samples, while all other attributes were complete. This minimal rate of missingness is unlikely to introduce significant bias. To address these missing entries, a KNN imputation strategy was applied, leveraging the similarity among observations to estimate absent values. This approach was specifically chosen over simpler methods like mean or median imputation because it preserves the multivariate relationships between features, thereby maintaining the underlying data structure and ensuring more plausible value estimations. . Specifically, the imputer considered the five nearest neighbors (n_neighbors = 5) for each sample, selected after comparative testing of different k values (3, 5, 7, 10) which demonstrated optimal balance between distribution preservation and computational efficiency. It calculated the imputed value as the average of these neighbors, thereby preserving the underlying distribution of the feature and maintaining relationships with other flow-based metrics. In parallel, outliers were identified using interquartile ranges and domain-specific thresholds and were either capped or removed to prevent disproportionate influence on model learning.

This combined preprocessing pipeline, encompassing feature encoding, KNN-based missing value imputation, outlier handling, data partitioning, and feature standardization, enhanced data quality, reduced noise, and ensured all features were adequately prepared, balanced, and scaled. As a result, it improves model robustness and predictive performance and strengthens the interpretability of intrusion detection results, providing a reliable foundation for both conventional ML and advanced DL architectures.

Hyperparameter Tuning

Hyperparameter optimization was conducted for all ML and DL models to ensure maximal predictive performance and fair comparison across algorithms. To ensure optimal performance and fair comparison, hyperparameter tuning was conducted using Optuna, an adaptive and efficient optimization framework based on Bayesian sampling, which was employed to explore the hyperparameter search space systematically(Joy & Selvan, 2022). The search process was guided by stratified 5-fold cross-validation for ML models and validation-based evaluation for DL models.

Ensemble Methods

This study employed several ensemble-based ML algorithms to enhance classification performance by aggregating multiple weak or base learners into robust predictive models.

The Gradient Boosting (Prettenhofer & Louppe, 2014) was implemented using the GradientBoostingClassifier from scikit-learn. This method constructs an additive model in a forward stage-wise manner, where at each iteration a weak learner is trained to approximate the negative gradient of the loss function. The learning rate and step size control the model update, while the complexity of the trees is regulated through the maximum depth and the number of estimators. For the classification task, the binomial deviance loss function was adopted, ensuring effective optimization of class probabilities. Hyperparameters were explored within the following ranges: n_estimators = 100–500 (selected: 200), learning_rate = 0.01–0.2 (selected: 0.1), and max_depth = 3–10 (selected: 5). The XGBoost algorithm was utilized via the XGBClassifier, which incorporates significant advancements over conventional boosting. Its objective function combines training loss and regularization terms to mitigate overfitting (Deo & Sanju, 2023). By employing histogram-based split finding and column-blocking optimization, XGBoost achieves superior computational efficiency, particularly in high-dimensional feature spaces. Hyperparameter ranges were identical to Gradient Boosting (n_estimators = 100–500, learning_rate = 0.01–0.2, max_depth = 3–10) with Optuna selecting 250 estimators, learning rate 0.1, and depth 5. The LightGBM (Quinto, 2020), applied via the LGBMClassifier, introduces two

key innovations: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which reduce computational cost and dimensionality. LightGBM grows trees leaf-wise, accelerating convergence and reducing memory consumption. Hyperparameter tuning explored n_estimators = 100–500 (selected: 200), learning_rate = 0.01–0.2 (selected: 0.1), and num_leaves = 31–127 (selected: 63).

The CatBoostClassifier addresses challenges in categorical feature handling and prediction shift. Using permutation-driven categorical encoding and ordered boosting, CatBoost prevents target leakage and reduces overfitting (Ibrahim et al., 2020). Hyperparameters were tuned over iterations = 100–500 (selected: 200), learning_rate = 0.01–0.2 (selected: 0.1), and depth = 3–10 (selected: 5). The RF classifier, implemented via RandomForestClassifier (Chaudhary et al., 2016), relies on bootstrap aggregating (bagging) of decision trees, with predictions obtained through majority voting. Hyperparameter ranges: n_estimators = 100–500 (selected: 200), max_depth = 5–15 (selected: 10). The Extra Trees method, applied via ExtraTreesClassifier, introduces additional randomness by selecting split points randomly rather than optimizing them (Hussein & Zeebaree, 2024). While slightly increasing bias, it reduces computational overhead and variance. Hyperparameter ranges: n_estimators = 100–500 (selected: 200), max_depth = 5–15 (selected: 10). Finally, AdaBoost (An & Kim, 2010) was implemented using AdaBoostClassifier. This algorithm iteratively adjusts training sample weights, emphasizing misclassified instances, and combines weak learners into a strong predictive ensemble. Hyperparameter tuning explored n_estimators = 50–200 (selected: 100) and learning_rate = 0.01–1 (selected: 0.1). By integrating Optuna-based hyperparameter optimization, all ensemble models were systematically tuned to achieve robust and reproducible performance across the ROSIDS23 dataset.

Traditional Machine Learning Models

In addition to ensemble methods, several well-established ML classifiers were implemented to establish baseline performance and enable comparative evaluation across diverse algorithmic paradigms. Optuna-based hyperparameter tuning was applied for each classifier to ensure optimal model performance using stratified 5-fold cross-validation.

The SVM (Rahman et al., 2015) was implemented using the SVC class with a Radial Basis Function (RBF) kernel, which constructs non-linear decision boundaries by projecting data into higher-dimensional space. Hyperparameters explored via Optuna included C = 0.1–100 (selected: 10) and gamma = 0.001–1 (selected: 0.01). The KNN algorithm (Zhang, 2021), implemented using KNeighborsClassifier, assigns labels to query instances based on the majority class of their nearest neighbors. Distance weighting was enabled, and hyperparameters tuned included n_neighbors = 3–15 (selected: 5) and p = 1–2 for Minkowski distance (selected: 2). The DT classifier partitions the feature space recursively through binary splits. Optuna explored max_depth = 3–20 (selected: 10), min_samples_split = 2–10 (selected: 5), and min_samples_leaf = 1–5 (selected: 2). The Gini impurity criterion was employed to select optimal splits. The LR (Peng et al., 2002) was implemented with L2 regularization using LogisticRegression. Hyperparameter tuning focused on the regularization strength C = 0.01–10 (selected: 1) and solver selection (lbfgs). Finally, the Gaussian NB classifier (Reddy et al., 2022) was implemented via GaussianNB. While it has no complex hyperparameters, Optuna optimized the var_smoothing parameter within the range 1e-12–1e-8 (selected: 1e-9) to improve numerical stability and predictive performance. All traditional classifiers were trained with parallel processing enabled and fixed random seeds (random_state=42) to ensure reproducibility across experiments.

Deep Learning Models

To capture complex, high-level feature interactions, three DL architectures were implemented using TensorFlow/Keras (Lee et al., 2021). These include CNN, which is particularly effective for extracting local patterns and spatial relationships in data through convolutional, pooling, and fully connected layers; GRU, a type of recurrent neural network that efficiently captures temporal dependencies and sequential patterns by using gating mechanisms to control information flow; and LSTM, which models long-range dependencies and non-linear feature relationships using memory cells and gating mechanisms (Shiri et al., 2023). Hyperparameter optimization for layer sizes, dropout rates, learning rates, and batch sizes was conducted using Optuna, ensuring adaptive exploration of optimal configurations.

The LSTM network was employed to model potential long-range dependencies and non-linear feature relationships. The architecture consisted of stacked LSTM layers with layer sizes explored in the range 32–128 units per layer (selected: 64), dropout rates 0.2–0.5 (selected: 0.3), and learning rates 0.0001–0.01 (selected: 0.001). Fully connected layers were incorporated before the softmax output layer. Early stopping and dynamic learning rate adjustments were applied to prevent overfitting and accelerate convergence. The GRU architecture was implemented as a lighter alternative to the LSTM. Layer sizes, dropout rates, and learning rates were tuned using Optuna with ranges identical to LSTM, resulting in 64 units, 0.3 dropout, and 0.001 learning rate as optimal selections. This enabled efficient training while retaining a comparable representational capacity to that of LSTM. The 1D-CNN treated input feature vectors as structured sequences, capturing localized patterns through stacked convolutional and pooling layers. Hyperparameter ranges explored included filter sizes 32–128 (selected: 64), kernel sizes 3–7 (selected: 5), dropout rates 0.2–0.5 (selected: 0.3), and learning rates 0.0001–0.01 (selected: 0.001). Fully connected layers preceded the softmax output layer to generate class probabilities. Early stopping and dynamic learning rate schedules were employed to ensure robust convergence. All DL models were compiled using adaptive optimization algorithms (Adam) and trained with categorical cross-entropy loss. Optuna-based hyperparameter tuning facilitated the discovery of optimal network configurations, enhancing predictive performance and generalizability across the ROSIDS23 dataset.

Evaluation Methodology and Metrics

A rigorous evaluation framework was adopted to systematically assess and compare the performance of all implemented models, ensuring both statistical robustness and practical relevance in the context of ROS intrusion detection.

For ML Models, a stratified 5-fold cross-validation (Szeghalmy & Fazekas, 2023) was employed on the training set to ensure generalizability and obtain robust performance estimates. In this procedure, the training set was divided into five equally sized folds, four for training and one for validation. This process was repeated five times so that each fold was the validation set once. The average training time per fold was recorded to evaluate computational efficiency. Following cross-validation, each model was retrained on the entire training set prior to evaluation on the hold-out test set, ensuring that the final performance metrics accurately reflected the models' capacity to generalize. For DL models, the dataset was split into 80% for training and 20% for testing. Within the training portion, 10% was further reserved as a validation set. This configuration enabled callbacks such as early stopping and dynamic learning rate adjustment, which helped prevent overfitting and accelerate convergence. Final performance metrics were calculated on the unseen test set, providing an unbiased estimate of predictive capability.

Multiple complementary metrics were utilized to provide a holistic assessment of model performance. Detection Accuracy was measured as the proportion of correctly classified instances across all classes. Precision, Recall, and F1-Score (Goutte & Gaussier, 2005) were computed using a weighted average across classes to account for class imbalance, ensuring that majority classes do not dominate the evaluation. The F1-score, as the harmonic mean of precision and recall, offers a single balanced measure of classification effectiveness. Computational Efficiency was also considered, with training time (per fold for ML models and total for DL models) and inference time on the whole test set recorded, providing insight into the feasibility of deploying the models in real-time or near-real-time intrusion detection system environments.

All experiments were conducted on a high-performance workstation featuring an AMD Ryzen 9 5950X processor and 32 GB RAM. The software stack included Python 3.9.13, TensorFlow 2.10. and scikit-learn 1.2.2. Fixed random seeds (random_state=42) were applied across NumPy, Python's built-in random module, scikit-learn, and TensorFlow to ensure full reproducibility and deterministic results. This setup guarantees that data splits, model initializations, and stochastic processes during training remain consistent across repeated runs, supporting reliable comparison and benchmarking of all ML and DL models.

Experimental Results and Analysis

This section comprehensively evaluates the performance of various ML and DL models in detecting intrusions within ROS communication networks. The analysis is structured first to provide a holistic comparison of all models, followed by a detailed dissection of the classification behavior of the top performers through confusion matrix analysis. The results are contextualized within the broader field of network intrusion detection to highlight their significance and potential implications.

Overall Performance Comparison

A rigorous comparative analysis evaluated the efficacy of fifteen distinct classifiers, encompassing a spectrum from traditional ML algorithms to advanced DL architectures and ensemble methods. The models were assessed based on four key metrics: Accuracy, Precision, Recall, and F1-Score. The unified value for each metric across all models, as detailed in Table 1, indicates a highly balanced performance between identifying positive classes (Recall) and maintaining the correctness of those identifications (Precision), which the F1-Score encapsulates.

Table 1

Classification Performance Metrics of Evaluated Models

Model	Accuracy	Precision	Recall	F1-Score
Gradient Boosting	0.99802	0.99802	0.99802	0.99802
LightGBM	0.99506	0.99506	0.99506	0.99506
XGBoost	0.99477	0.99477	0.99477	0.99477
DT	0.99038	0.99040	0.99038	0.99039
CatBoost	0.99023	0.99028	0.99023	0.99023
RF	0.98778	0.98787	0.98778	0.98779
CNN	0.98548	0.98565	0.98548	0.98550
KNN	0.98288	0.98301	0.98288	0.98289
GRU	0.97864	0.97883	0.97864	0.97864
LSTM	0.97652	0.97671	0.97652	0.97651

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.97099	0.97116	0.97099	0.97069
RF	0.96386	0.96344	0.96386	0.96328
Gaussian NB	0.79965	0.91008	0.79965	0.82752
AdaBoost	0.68303	0.49702	0.68303	0.56621

The results reveal a clear and statistically significant performance hierarchy. Ensemble methods, particularly gradient boosting algorithms, demonstrated paramount superiority. Gradient Boosting emerged as the unequivocal benchmark, achieving near-perfect scores across all metrics (0.99802). This exceptional performance is attributed to the model's sequential learning process, which iteratively corrects the errors of previous weak learners, making it highly adept at capturing complex, non-linear patterns and interactions within the network traffic data. The other gradient-boosting variants, LightGBM (0.99506) and XGBoost (0.99477), followed closely. Their high performance, computational efficiency, and built-in regularization techniques make them efficient choices for real-time intrusion detection systems where both accuracy and low latency are critical. Traditional ensemble methods like RF (0.98778) and tree-based models like DT (0.99038) showed strong, competitive results. Their performance underscores the effectiveness of combining multiple decision trees to reduce variance and overfitting, though the more advanced boosting techniques slightly outperformed them.

A noteworthy observation is the comparative performance of DL models (CNN: 0.98548, GRU: 0.97864, LSTM: 0.97652). While achieving respectable accuracy, they lagged behind the top ensemble models. The nature of the dataset can explain this. However, ROS network traffic has temporal sequences (captured by GRU/LSTM) and can be structured as packet-based features (for CNN); the dataset size might not have been sufficiently large to fully leverage the DL models' capacity without extensive hyperparameter tuning. In contrast, ensemble methods often achieve superior performance on structured tabular data, as is the case here, with greater computational efficiency. The performance of AdaBoost (0.68303) and Gaussian NB (0.79965) was significantly lower. AdaBoost's weakness may stem from its sensitivity to noisy data and outliers, which are often present in network traffic. Gaussian NB's lower accuracy but relatively high precision suggests it was conservative in its predictions, correctly identifying attacks but missing a substantial number of them (low recall), likely due to its assumption of feature independence being violated in this complex dataset.

Confusion Matrix Analysis of Top Performers

To move beyond aggregate metrics and understand the specific nature of classification errors, a detailed confusion matrix analysis (Heydarian et al., 2022) was conducted for the three top-performing models: Gradient Boosting, XGBoost, and LightGBM.

As illustrated in Figure 2, the Gradient Boosting model demonstrated near-perfect discriminative capability. Its performance on the Benign class was outstanding, correctly identifying 12,492 instances (99.91%) with only 11 misclassifications distributed across various attack categories. Maintaining such a low false positive rate is critical for cybersecurity systems to prevent unnecessary disruptions in robotic operations. Regarding attack detection, the model performed almost flawlessly for high-volume attacks: DoS (99.97%) and Subflood (99.93%), which generate highly distinctive traffic patterns that are easily separable. The model also achieved excellent results in detecting Unauthorized Publish (99.36%). The greatest number of errors occurred with the Unauthorized Subscribe attack, resulting in 27 false negatives (2.55% missed detection rate). This indicates that unauthorized subscription attempts exhibit subtler patterns that are

more challenging to differentiate from benign behavior, highlighting the need for more sophisticated feature learning.

Figure 2

Gradient Boosting Confusion Matrix

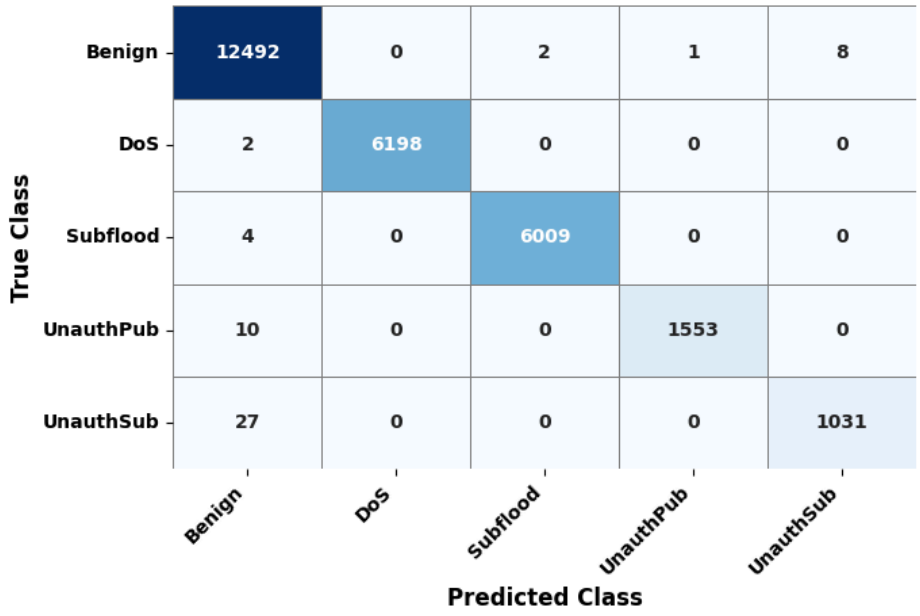


Figure 3

XGBoost Confusion Matrix

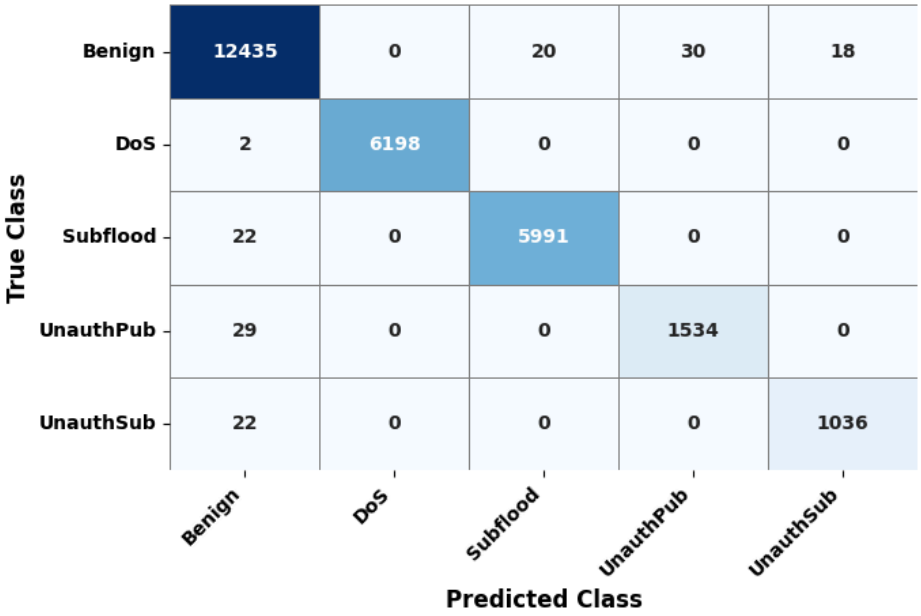
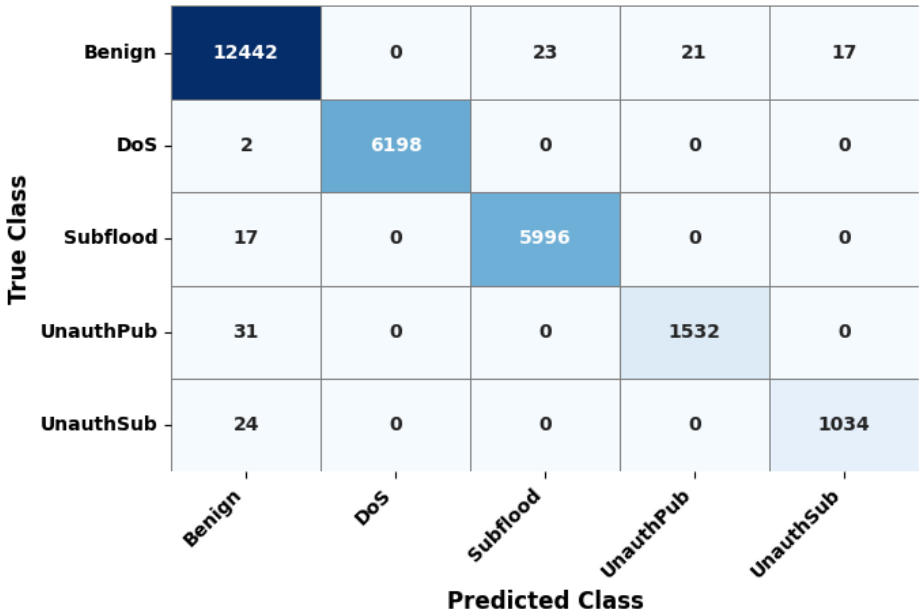


Figure 4

LightGBM Confusion Matrix



As detailed in Figure 3 and Figure 4, XGBoost and LightGBM demonstrated robust performance with a slight but observable increase in error rates compared to Gradient Boosting. Both models showed a higher number of misclassified Benign instances (68 and 61, respectively), primarily as Subflood, UnauthPub, and UnauthSub. This indicates a marginally higher false positive rate. Similarly, their performance on the UnauthSub attack was slightly weaker, with 22 and 24 false negatives, respectively. Crucially, all three models maintained perfect recall (100%) for the DoS attack, with no instances being missed. This consistent result across architectures confirms that DoS attacks leave an unambiguous fingerprint in the ROS network data, making them the easiest category to detect.

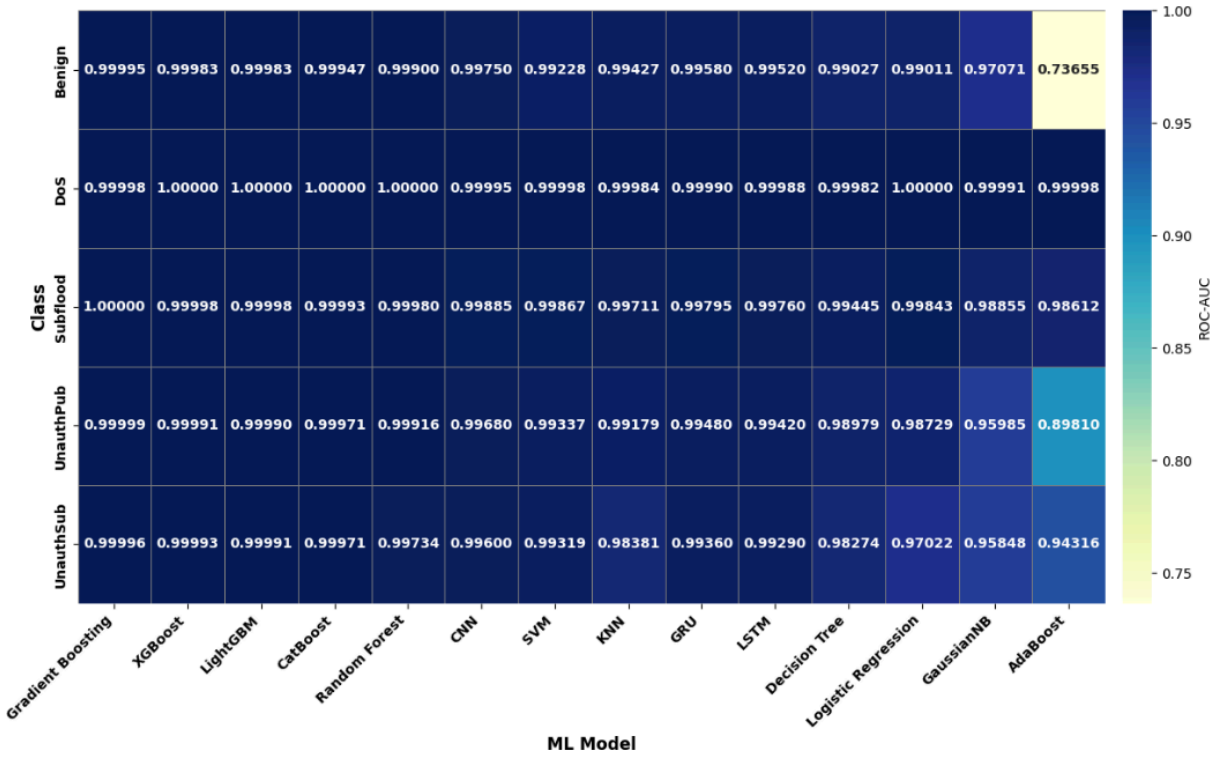
Additionally, the overall classification integrity of the three models was evaluated using the Matthews Correlation Coefficient (MCC) (Chicco & Jurman, 2020) , a balanced metric particularly well-suited for multi-class and potentially imbalanced classification tasks. The obtained MCC scores were 0.9992 for Gradient Boosting, 0.9978 for XGBoost, and 0.9980 for LightGBM. These near-perfect values demonstrate that, despite minor variations in error distribution, all three models deliver highly accurate, stable, and well-balanced classification performance across all five classes.

ROC-AUC Performance Analysis

While accuracy and F1-score provide a snapshot of performance at a specific decision threshold, the ROC Area Under the Curve (ROC-AUC) metric (Chicco & Jurman, 2023) evaluates a model's ability to discriminate between classes across all possible classification thresholds. This analysis is particularly valuable for understanding the inherent separability of the feature space and the model's confidence in its predictions, as illustrated in Figure 5.

Figure 5

ROC-AUC Performance Metrics



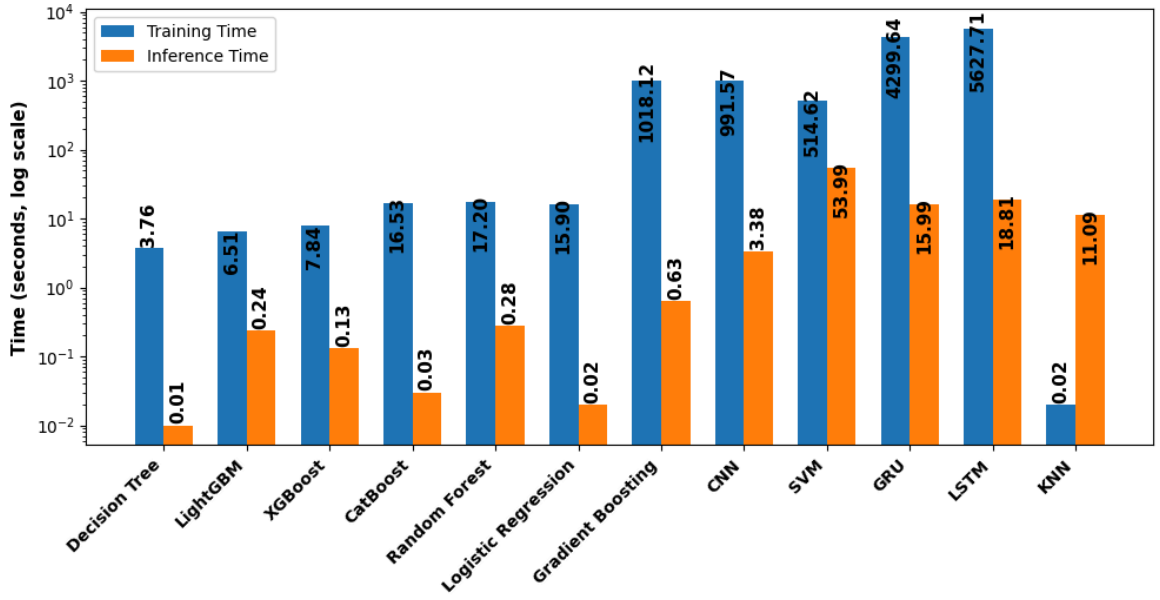
As shown in Figure 5, the ROC-AUC results further substantiate the superiority of gradient boosting algorithms. Gradient Boosting achieved an almost perfect macro-average AUC of 0.99998, indicating an exceptional capability to discriminate between all classes, which is consistent with its perfect F1-score. A notable observation is that the DoS attack attained perfect or near-perfect AUC values (≥0.99998) across nearly all evaluated models, including simpler approaches such as Logistic Regression. This suggests that DoS traffic exhibits features that are inherently distinctive from both benign traffic and other attack types. Similarly, the Subflood attack demonstrated high separability, with top-performing models achieving AUC values of ≥0.99993. In contrast, the UnauthSub attack consistently yielded the lowest AUC scores among the top-performing models (although still excellent, e.g., 0.99996 for Gradient Boosting). This finding aligns with the confusion matrix analysis, indicating that its behavioral patterns closely resemble legitimate subscription activities, thereby posing the most significant classification challenge.

Computational Efficiency Analysis

Beyond raw accuracy, the practical deployment of cybersecurity solutions in resource-constrained robotic and edge computing environments necessitates a careful evaluation of computational efficiency. This assessment examines the critical trade-off between model performance and the computational resources required for both training and inference, a factor paramount to real-world applicability, as shown in Figure 6.

Figure 6

Computational Efficiency Analysis



As shown in Figure 6, the results reveal profound disparities in computational demands across the model spectrum. DT algorithm was the most efficient by a significant margin, with a total computation time of merely 3.77 seconds. This exceptional efficiency is attributed to its simple, non-parametric structure, which requires no complex iterative optimization or hyperparameter tuning. Among the top-performing models, LightGBM and XGBoost demonstrated an exceptional performance-to-efficiency ratio. Their total times of 6.75s and 7.97s, respectively, are orders of magnitude lower than the top-performing Gradient Boosting model, while their accuracy metrics remained highly competitive (F1-Score > 0.994). This efficiency stems from their histogram-based learning and leaf-wise growth strategies, which drastically reduce computational overhead during training.

A critical finding is the dissociated nature of training and inference costs. While Gradient Boosting required a substantial training investment (1018.12s), its inference time was remarkably low (0.63s). This characteristic is ideal for a deployment paradigm where a model is trained once in a high-resource environment and then deployed for continuous, low-latency inference on edge devices. Similarly, DL models (GRU, LSTM, CNN) exhibited exorbitantly high training times due to the backpropagation through time and numerous learnable parameters. Their inference times, though higher than tree-based models, were still acceptable for near-real-time analysis.

The KNN algorithm presented a unique profile: near-instantaneous training (0.02s) but the highest inference time (11.09s) among the non-DL models. This is because KNN is a lazy learner, deferring all computation to the inference phase by comparing each new instance to the entire training dataset. This makes it unsuitable for large-scale or rapid deployment scenarios. Conversely, SVM with likely a non-linear kernel (e.g., RBF) showed high training and inference times, confirming their known scalability issues with large datasets.

In summary, for real-time ROS cybersecurity, LightGBM and XGBoost offer the most compelling balance, providing high accuracy while maintaining a minimal computational footprint. Gradient Boosting continues

to set the benchmark for accuracy, but is better suited to scenarios where longer training times are acceptable.

Discussion

The evaluation of gradient boosting-based intrusion detection models demonstrates that these algorithms, particularly Gradient Boosting, LightGBM, and XGBoost, offer state-of-the-art performance for ROS environments. Gradient Boosting consistently achieved near-perfect macro-average AUC values (0.99998) and F1-scores, indicating an exceptional ability to discriminate between all attack and benign classes. This is further supported by confusion matrix analyses, which show that DoS and Subflood attacks are detected with near-100% accuracy across all top models. These results suggest that such attacks produce highly distinctive traffic patterns, which are inherently easier to classify. Consequently, Gradient Boosting is most appropriate for scenarios where detection accuracy is the highest priority, even if training demands significant computational resources. This makes it ideal for offline training and periodic model updates, where resources are abundant.

For real-time deployment on edge devices, LightGBM offers a compelling alternative. Achieving 0.99506 overall accuracy with only 6.75 seconds of total training time, it provides a practical balance between performance and computational efficiency. This enables deployment on robots or edge gateways, ensuring low-latency inference without compromising detection capabilities. XGBoost serves as a robust alternative, combining strong performance with moderate resource requirements, and its precise regularization helps mitigate overfitting risks, making it suitable for both real-time and resource-constrained settings. Despite XGBoost being an optimized variant, its slightly lower performance on this specific task highlights how the optimal algorithm choice is highly dependent on the dataset characteristics, where Gradient Boosting's exact split-finding and simpler regularization mechanism proved more effective for the nuanced class boundaries in this cybersecurity domain.

The analysis also reveals attack-specific considerations. While DoS and Subflood attacks are reliably detected by nearly all top-performing models, UnauthPub and UnauthSub attacks remain more challenging. UnauthSub, in particular, exhibited minor misclassifications due to behavioral similarities with legitimate subscriptions, as reflected in slightly lower AUC scores (e.g., 0.99996 for Gradient Boosting). These observations suggest that incorporating additional contextual information, such as authentication logs or sequence-based traffic features, could further enhance the detection of these subtle threats.

The consistently high AUC scores across top models indicate significant operational flexibility. Administrators can adjust classification thresholds based on specific objectives, whether minimizing false positives to prevent operational disruptions or maximizing detection of sophisticated attacks. Overall, the findings illustrate that gradient boosting algorithms successfully reconcile the need for high-accuracy, multi-class detection with practical deployment constraints. Gradient Boosting, LightGBM, and XGBoost provide robust, deployable solutions for ROS systems, ensuring reliable protection across research, industrial, and critical application environments.

Despite the promising results, this study has several limitations that must be acknowledged. Firstly, while the ROSIDS23 dataset is a valuable benchmark, its simulation-based nature may not fully capture the full spectrum of real-world complexities, such as anomalies from unpredictable physical interactions, true hardware diversity across platforms, sensor noise, or actual network latencies, potentially limiting the generalizability of models to operational deployments. Secondly, the models were evaluated under static

conditions and did not account for adaptive adversaries capable of evolving their strategies to evade detection. Thirdly, the high computational cost of top-performing ensemble models like Gradient Boosting could be prohibitive for large-scale or resource-constrained edge deployments. Finally, the persistent challenge in detecting subtle attacks like Unauthorized Subscribe (UnauthSub) suggests the need for more sophisticated feature engineering or sequence-based analysis. Future work should therefore focus on validation with real-world data, incorporating adversarial simulations, developing more efficient models, and enhancing feature sets to improve robustness and resilience.

Conclusion

This study presents a comprehensive evaluation of fifteen state-of-the-art ML and DL models for intrusion detection in ROS environments, using the novel ROSIDS23 dataset. The results demonstrate that ensemble methods, particularly Gradient Boosting, LightGBM, and XGBoost, consistently outperform DL architectures and traditional classifiers in detecting multi-class robotic cyber attacks, achieving near-perfect accuracy, precision, recall, F1-score, and ROC-AUC values. Gradient Boosting emerged as the top performer, providing an optimal balance of accuracy and robust detection across all attack types, while LightGBM offers a practical trade-off between high performance and computational efficiency suitable for real-time deployment on edge devices. Detailed confusion matrix and ROC-AUC analyses reveal that high-volume attacks such as DoS and Subflood are reliably detected by all top-performing models. Nevertheless, more subtle threats, such as Unauthorized Subscribe attacks, remain particularly difficult to detect due to their behavioral resemblance to legitimate traffic. This indicates that future models may benefit from incorporating sequence-based features or additional contextual information. The study positions ROSIDS23 as a benchmark dataset for robotic intrusion detection and demonstrates the practical viability of ensemble-based cybersecurity solutions for ROS systems. The findings offer valuable insights for deploying accurate and low-latency security mechanisms in robotic applications across healthcare, manufacturing, and critical infrastructure. Promising directions for future research include evaluating models under adaptive adversarial conditions, exploring hybrid feature augmentation techniques, and integrating ensemble and DL approaches to improve the detection of subtle attacks. To enhance the practical validity and robustness of the proposed framework, future work should focus on validating the top-performing models using real network traffic and the ROS2 infrastructure. Expanding the dataset to encompass more diverse, complex, and large-scale robotic network traffic scenarios will also be crucial. Furthermore, to build more adaptive and generalizable intrusion detection systems, investigating advanced learning paradigms such as online learning for continuous model adaptation in dynamic robotic networks and transfer learning to leverage pre-existing knowledge from related cybersecurity domains is highly recommended. Addressing these challenges will enable the development of next-generation cybersecurity frameworks that further strengthen the resilience and safety of cyber-physical robotic systems.

Peer Review Externally peer-reviewed.
Conflict of Interest The author has no conflict of interest to declare.
Grant Support The author declared that this study has received no financial support.

Author Details Hamdullah Karamollaoğlu
[1] Electricity Generation Corporation, Ankara, Türkiye
0000-0001-6419-2249 h.karamollaoglu@euas.gov.tr

References

Ahmad Yousef, K. M., AlMajali, A., Ghalyon, S. A., Dweik, W., & Mohd, B. J. (2018). Analyzing cyber-physical threats on robotic platforms. *Sensors*, *18*(5), 1643.

An, T.-K., & Kim, M.-H. (2010). A new diverse AdaBoost classifier. *2010 International conference on artificial intelligence and computational intelligence*, 359–363.

Bezemskij, A., Loukas, G., Anthony, R. J., & Gan, D. (2016). Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle. *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, 61–68.

Botta, A., Rotbei, S., Zinno, S., & Ventre, G. (2023). Cyber security of robots: A comprehensive survey. *Intelligent Systems with Applications*, *18*, 200237.

Chaudhary, A., Kolhe, S., & Kamal, R. (2016). An improved random forest classifier for multi-class classification. *Information Processing in Agriculture*, *3*(4), 215–222.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 6.

Chicco, D., & Jurman, G. (2023). The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification. *BioData Mining*, *16*(1), 4.

Değirmenci, E., Kırca, Y. S., Özçelik, İ., & Yazıcı, A. (2023). ROSIDS23: Network intrusion detection dataset for robot operating system. *Data in Brief*, *51*, 109739.

Değirmenci, E., Özçelik, I., & Yazıcı, A. (2024). Adversarial Attack Detection Approach for Intrusion Detection Systems. *IEEE Access, 12,* 195996–196009.

Deo, T. Y., & Sanju, A. (2023). Data imputation and comparison of custom ensemble models with existing libraries like XGBoost, CATBoost, AdaBoost and Scikit learn for predictive equipment failure. *Materials Today: Proceedings*, *72*, 1596–1604.

Dhanabal, L., & Shantharajah, S. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6), 446–452.

Goutte, C., & Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. *27th European Conference on IR Research*, 345–359.

Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-label confusion matrix. *Ieee Access*, *10*, 19083–19095.

Holdbrook, R., Odeyomi, O., Yi, S., & Roy, K. (2024). Network-Based Intrusion Detection for Industrial and Robotics Systems: A Comprehensive Survey. *Electronics*, *13*(22), 4440.

Hussein, N., & Zeebaree, S. R. (2024). Performance evaluation of extra trees classifier by using cpu parallel and non-parallel processing. *The Indonesian Journal of Computer Science*, *13*(2), 1859–1872.

Ibrahim, A. A., Ridwan, R. L., Muhammed, M. M., Abdulaziz, R. O., & Saheed, G. A. (2020). Comparison of the CatBoost classifier with other machine learning methods. *International Journal of Advanced Computer Science and Applications*, *11*(11), 738–748.

Joy, J., & Selvan, M. P. (2022). A comprehensive study on the performance of different Multi-class Classification Algorithms and Hyperparameter Tuning Techniques using Optuna. *International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, 1–5.

Kang, J., Kim, K., & Kwon, D. (2025). Watch Your Callback: Offline Anomaly Detection using Machine Learning in ROS 2. *IEEE Access, 13,* 60763-60775.

Krejčí, J., Babiuch, M., Suder, J., Krys, V., & Bobovský, Z. (2025). Internet of robotic things: Current technologies, challenges, applications, and future research topics. *Sensors*, *25*(3), 765.

Lee, T., Singh, V. P., & Cho, K. H. (2021). Tensorflow and keras programming for deep learning. *Deep learning for hydrometeorology and environmental science*, 151–162.

Martín, F., Soriano, E., & Cañas, J. M. (2018). Quantitative analysis of security in distributed robotic frameworks. *Robotics and Autonomous Systems*, *100*, 95–107.

Monoscalco, L., Simeoni, R., Maccioni, G., & Giansanti, D. (2022). Information security in medical robotics: A survey on the level of training, awareness and use of the physiotherapist. *Healthcare, 10(1)*, 159–175.

Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *International Journal of Engineering & Technology*, *7*(3.24), 479–482.

Peng, C.-Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*, *96*(1), 3–14.

Prettenhofer, P., & Louppe, G. (2014). *Gradient boosted regression trees in scikit-learn*. PyData 2014.

Pu, H., He, L., Cheng, P., Sun, M., & Chen, J. (2022). Security of industrial robots: Vulnerabilities, attacks, and mitigations. *IEEE Network*, *37*(1), 111–117.

Quinto, B. (2020). *Next-generation machine learning with spark: Covers XGBoost, LightGBM, Spark NLP, distributed deep learning with keras, and more*. Apress.

Rahman, H. A. A., Wah, Y. B., He, H., & Bulgiba, A. (2015). Comparisons of ADABOOST, KNN, SVM and logistic regression in classification of imbalanced dataset. *International conference on soft computing in data science,* 54–64.

Raza, A., Memon, S., Nizamani, M. A., Dhomeja, L. D., Memon, N., & Charan, K. (2024). Machine Learning Techniques for Cyber Security in Internet of Robotic Things. *VFAST Transactions on Software Engineering*, *12*(3), 01–10.

Reddy, E. M. K., Gurrala, A., Hasitha, V. B., & Kumar, K. V. R. (2022). Introduction to Naive Bayes and a review on its subtypes with applications. *Bayesian Reasoning and Gaussian Processes for Machine Learning Applications*, 1–14.

Sharma, U., Medasetti, U. S., Deemyad, T., Mashal, M., & Yadav, V. (2024). Mobile robot for security applications in remotely operated advanced reactors. *Applied Sciences*, *14*(6), 2552.

Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *arXiv Preprint arXiv:2305.17473*.

Szeghalmy, S., & Fazekas, A. (2023). A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning. *Sensors*, *23*(4), 2333.

Tanimu, J. A., & Abada, W. (2025). Addressing cybersecurity challenges in robotics: A comprehensive overview. *Cyber Security and Applications*, *3*, 100074.

Tsapin, D., Pitelinskiy, K., Suvorov, S., Osipov, A., Pleshakova, E., & Gataullin, S. (2024). Machine learning methods for the industrial robotic systems security. *Journal of Computer Virology and Hacking Techniques*, *20*(3), 397–414.

Verma, N., Kumar, N., Sheikh, Z. A., Koul, N., & Ashish, A. (2025). Machine Learning for the Cybersecurity of Robotic Cyber-Physical Systems: A Review. *Procedia Computer Science*, *259*, 1817–1826.

Verma, N., Kumar, N., Verma, C., Illés, Z., & Singh, D. (2025). A systematic review on cybersecurity of robotic systems: Vulnerabilities trends, threats, attacks, challenges, and proposed framework. *International Journal of Information Security*, *24*(3), 127.

Yaacoub, J.-P. A., Noura, H. N., Salman, O., & Chehab, A. (2022). Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security*, *21*(1), 115–158.

Zafar, M. H., Langas, E. F., Aftab, M. F., & Sanfilippo, F. (2024). Enhanced intrusion detection in robot operating systems via grid search based multi-head attention stacked convolutional network. *20th International Conference on Automation Science and Engineering (CASE)*. 3880–3885.

Zhang, S. (2021). Challenges in KNN classification. *IEEE Transactions on Knowledge and Data Engineering*, *34*(10), 4663–4675.