Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2025) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

A Real-Time Intrusion Detection System for Enhancing Cybersecurity in Robotic Systems

Narinder Verma [1], Neerendra Kumar[2]

[1,2] *Department of Computer Science & IT, Central Uuniverity of Jammu, Jammu, 181143, India*

Abstract

As robotic systems become increasingly integrated into various sectors, the need for robust cybersecurity measures to protect these systems from cyber threats has become paramount. This research presents a novel Real-Time Intrusion Detection System (IDS) designed specifically for the cybersecurity of robotic systems. The proposed IDS continuously monitors network traffic to identify potential threats, distinguishing between normal and malicious activities A testbed simulates network traffic, capturing real-time data between a robotic device and a PC server under normal and simulated attack scenarios. The system employs a comprehensive approach that includes data collection, preprocessing, model training using machine learning models, deployment, and real-time prediction. Network traffic data is captured using tools such as Wireshark. The preprocessing phase involves feature extraction, data cleaning, and normalization to prepare the dataset for training. The machine learning models are trained on labeled data to learn patterns indicative of normal operations versus attacks. Upon deployment, the IDS analyzes incoming traffic in real time, classifying it as benign or malicious. Experimental results indicate that the IDS effectively distinguishes between normal and attack traffic, with the decision tree classifier achieving an accuracy of 96.61%.

Peer-review under responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering
Keywords: robotic cybersecurity; secure robotic communication; intrusion detection system; intelligent framework

1. Introduction

The rapid proliferation of robotic systems across various sectors, including manufacturing, healthcare, and defense, has ushered in a new era of automation and efficiency. However, this advancement has also introduced significant cybersecurity challenges. As robotic systems become increasingly interconnected and integrated into critical infrastructure, the system present attractive targets for cyber-attacks, potentially leading to severe consequences ranging from operational disruptions to safety hazards and economic losses [1]. The development of robust and efficient real-time intrusion detection systems (IDS) for robotic systems has thus become a critical imperative in the field of cybersecurity [2]. Robotic systems are inherently complex, comprising numerous components such as

Peer-review under responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering

sensors, actuators, controllers, and communication interfaces [3] . Traditional cybersecurity measures often fall short in protecting these systems, as they lack the capability to detect and respond to threats in real-time without compromising system performance [4] . Recent studies have underscored the unique challenges in securing robotic systems. For instance, the vulnerability of industrial robots to cyberattacks highlights the need for specialized security solutions [5]. Similarly, the security vulnerabilities in teleoperated surgical robots emphasized the critical importance of cybersecurity in robotic healthcare applications [6].

While existing studies have explored various machine learning solutions, none have demonstrated implementation in a real-time architecture [7][8][9][10]. Our work addresses this gap by testing the proposed framework on real-time network traffic data. The methodology involves capturing network traffic via pcap files using Wireshark [11] preprocessing and cleaning the data. The CICflowmeter [12] tool is used for feature extraction. Further, the ML models such as DT, RF, and KNN trained on captured real-time dataset containing both normal and attack data. The attack data is generated using scapy packages via python scripts. Finally, the trained model is deployed and tested on real-time traffic data, achieving good accuracy in detecting abnormal or attack traffic. This study presents a novel approach to real-time intrusion detection for robotic systems, employing machine learning techniques to identify and classify potential security threats. By continuously monitoring communication between robotic devices and server machines, the IDS can promptly detect and respond to abnormal activities, thereby ensuring the security of the system.

The main contributions of the presented study are as follows:

1. **Analysis of Existing Cybersecurity Challenges:** The study provides an analysis of the current cybersecurity challenges facing robotic systems, highlighting the vulnerabilities and risks associated with interconnected robotic networks.
2. **Design and Development of a Novel Framework:** A new IDS framework is designed and developed, tailored specifically for the unique requirements of real-time robotic systems.
3. **Real-Time Dataset Capture and Model Training:** A testbed environment, comprising a robotic device and a server machine, is created to capture real-time network traffic. The dataset generated from this testbed includes both normal and attack scenarios, providing a comprehensive basis for training the IDS model.
4. **Performance Analysis:** The trained IDS model is evaluated using real-time network traffic in the testbed environment. The performance of the model is analyzed in terms of accuracy, precision, recall, and response time, demonstrating its effectiveness in identifying and mitigating security threats.

The remainder of this paper is organized as follows. Section 2 presents an overview of the existing cybersecurity challenges in robotic systems and the motivation for this study. Section 3 details the design and development of the proposed IDS framework, including the machine learning algorithms, the experimental setup and the process of capturing real-time data from the testbed. Section 4 discusses the results of the performance analysis. Finally, Section 5 concludes the paper, summarizing the findings and suggesting directions for future research.

2. Related work

This section examines recent advancements in intrusion detection systems (IDS) for robotic environments, with a particular emphasis on real-time detection capabilities and machine-learning approaches. To study vulnerabilities in robotic Systems authors in [5] conducted a comprehensive study on the cybersecurity of industrial robots, revealing significant vulnerabilities in their software and communication protocols. Their work demonstrated how these vulnerabilities could be exploited to compromise the integrity and availability of robotic systems, emphasizing the need for robust security measures. Building on this, researchers in [13]) explored security issues in teleoperated surgical robots, highlighting the potential risks to patient safety and privacy in healthcare robotics. These studies underscored the critical nature of developing specialized security solutions for robotic systems across different domains [14].

The application of machine learning techniques to intrusion detection has gained significant traction in recent years. Authors in [15] proposed a machine learning-based IDS for cyber-physical systems, including robotic platforms. Their approach utilized a combination of network traffic analysis and physical system metrics to detect attacks, achieving promising results in experimental settings. However, their work primarily focused on offline analysis rather than real-time detection. The use of deep learning techniques for intrusion detection in industrial control

systems is presented in [16], which share many similarities with robotic environments. Their study demonstrated the potential of convolutional neural networks (CNNs) in detecting various types of attacks, including DoS attacks. While their approach showed high accuracy, it did not address the challenges of real-time implementation in dynamic robotic environments.

The need for real-time intrusion detection in robotic systems presents unique challenges. Authors in [17] discussed the time-sensitive nature of robotic operations and the potential consequences of delayed threat detection. They proposed a framework for near-real-time anomaly detection but did not provide a comprehensive implementation or evaluation in a live robotic environment.

Kravchik et. al. in [18] presented a study on detecting cyberattacks in industrial control systems using convolutional neural networks, achieving high accuracy in near-real-time scenarios [6]. While their work provided valuable insights into the application of deep learning for rapid threat detection, it did not specifically address the unique characteristics of robotic systems or demonstrate implementation in a fully real-time architecture.

Table 1 presents the overview of existing studies related to the cybersecurity of robotic systems. While existing literature has made significant strides in developing machine learning-based intrusion detection systems and addressing the security challenges of robotic systems, there remains a notable gap in demonstrating the implementation and evaluation of these approaches in real-time robotic environments. Most studies have focused on offline analysis or near-real-time detection, leaving the challenges of true real-time implementation largely unaddressed.

Table 1. Summary of existing studies related to cybersecurity of robotic system

Ref	**Technique used**	**Application Context**	**Dataset utilized**	**Devices utilized**	**Pros**	**Cons**
[19]	Ensemble learning methods such as KNN, eXtreme Gradient Boosting, and RF	IoT MQTT Networks	Custom Dataset	Raspberry PI	Accuracy >92%	Detailed analysis of the computational complexity of the proposed framework is not presented
[20]	Hybrid IDS based on SVM and GWO	IOT	NSL-KDD and TON_IoT	-	Accuracy =98%(approx.)	The feasibility of the proposed system for other datasets is not presented
[21]	SNN, DT, BT, KNN, SVM	IoT system	IoTID20	IoT devices	100% accuracy in detection and 99.4–99.9% accuracy in classification	Feasibility in real-world IoT scenarios needs to be considered.
[17]	DNN	IoT System	CIC-IDS2017	PC and Raspberry Pi	Average accuracy of 99.57%	• Training the attack detection model requires labelled data. • The presented system is vulnerable to adversarial attacks
[22]	ML-based IDS	IOMT	ToN_IoT	_	Best Accuracy of 99.18%	• Real-time implementation is not presented

3. Methodology

This section outlines the proposed methodology, framework, and algorithm developed for this research. The methodology is divided into two main stages: the first stage focuses on data collection, where real-time network traffic data is captured for training the model. The second stage involves evaluation, during which the model predicts and classifies network traffic as either normal or abnormal.

The data collection model is a fundamental component in the development of an Intrusion Detection System (IDS) designed for robotic devices. This model is responsible for capturing, processing, and organizing network traffic data, which is crucial for training and evaluating the IDS. The flow diagram of the data collection model is presented in Figure 1. The model contains main components such as a robotic device and sever, focusing on the interaction between the robotic device (acting as the client) and the PC (acting as the server).

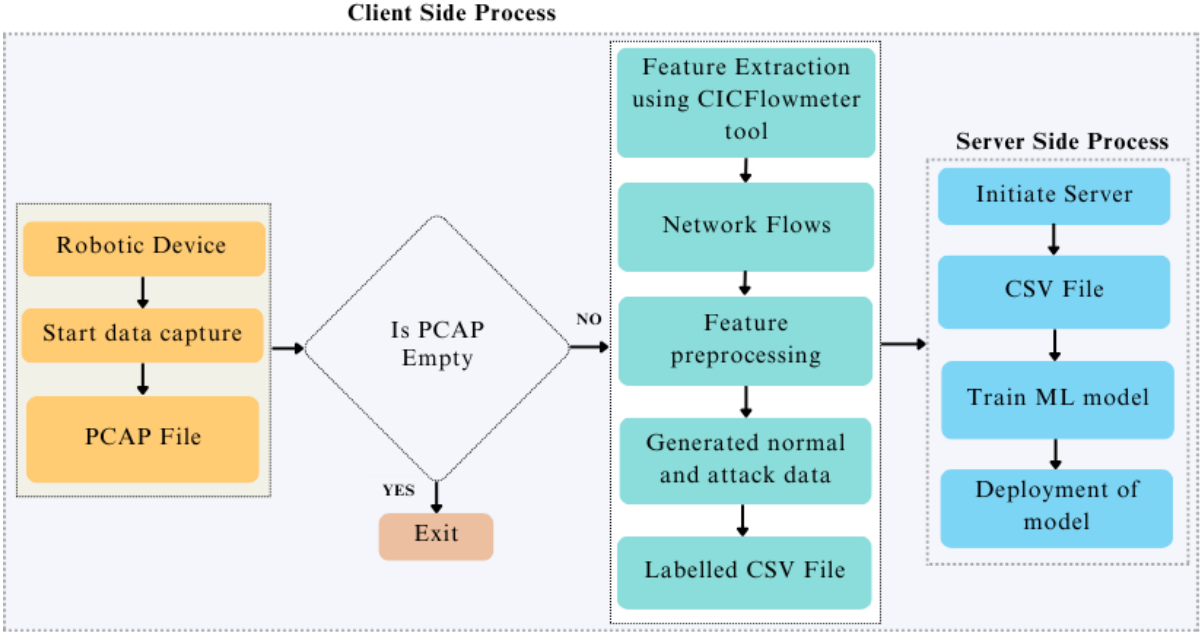

Figure 1. Flow diagram of the data collection and proposed work

The robotic device serves as the client in the network communication setup, generating traffic through its routine operations. This device is configured to communicate with a server (PC), and during this communication, both normal and attack traffic are generated. Normal traffic represents typical operations such as sensor data transmission, command execution, and status updates. In contrast, attack traffic is generated by simulating cyber-attacks, such as Denial of Service (DoS) utilizing scapy packages and python scripts. The differentiation between normal and attack traffic is critical for creating a labeled dataset, which is essential for training the IDS.

Figure 2 depicts the total number of instances taken into consideration for training the model. To collect the network traffic data, packet sniffing tools such as Wireshark, tcpdump, or Scapy are employed. These tools are instrumental in capturing detailed information about the network communication between the robotic device and the server. The captured data includes various parameters, such as IP addresses, port numbers, protocols, timestamps, packet lengths, and payload data as presented in Figure 3. The sample features are presented in figure. This comprehensive capture of network traffic provides the raw data needed for feature extraction and subsequent analysis.

Traffic capture is conducted in two distinct phases: one during the normal operation of the robotic device to gather standard communication patterns, and the other during simulated attack scenarios to capture malicious behavior. This dual-phase approach ensures that the dataset includes a diverse range of traffic types, which is essential for training a robust IDS.

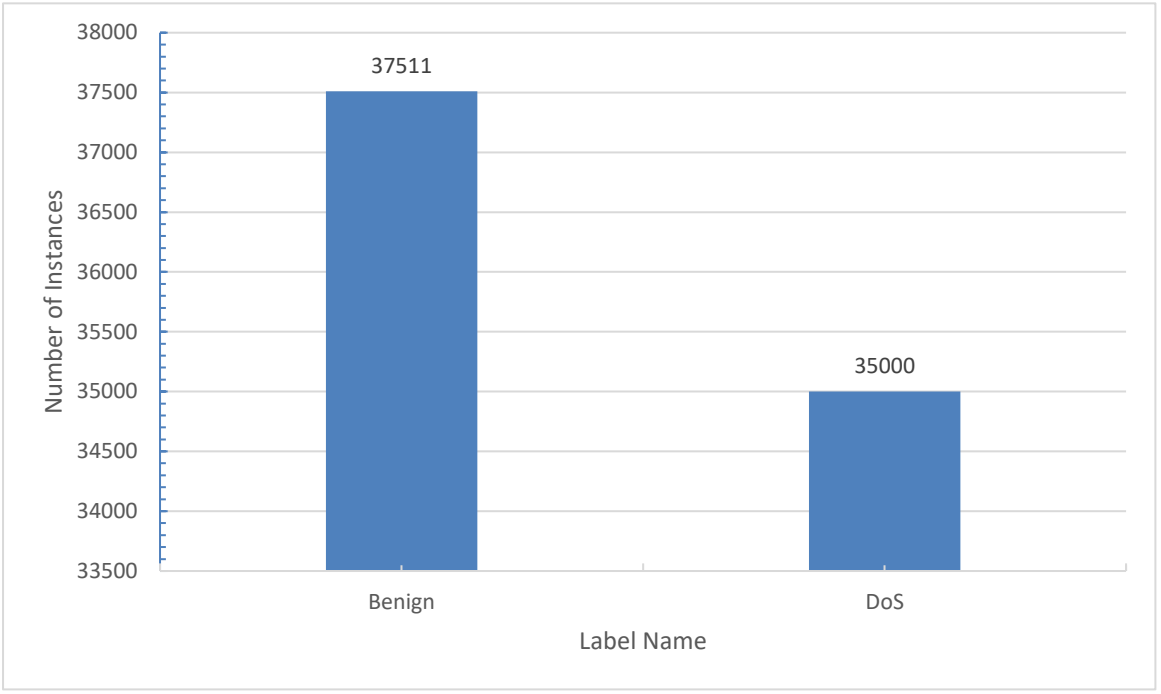

Figure 2. Proportion of normal and attack instances in the captured dataset.

S.No No	Feature Name	Description	Data Type	S.No	Feature Name	Description	Data Type
1	Flow ID	Unique identifier for the flow of packets.	Integer	11	Fwd Pkts/s	Number of packets in the forward direction per second.	Float
2	Src IP	Source IP address from which the packet originated.	Categorical	12	Bwd Pkts/s	Number of packets in the backward direction per second.	Float
3	Src Port	Source port of the traffic.	Integer	13	Pkt Len Min	Minimum length of a packet.	Float
4	Dst IP	Destination IP address to which the packet is sent.	Categorical	14	Pkt Len Max	Maximum length of a packet.	Float
5	Dst Port	Destination port of the traffic.	Integer	15	Pkt Len Mean	Mean length of a packet.	Float
6	Protocol	Network protocol used in the packet (TCP/UDP etc.)	Categorical	16	Pkt Len Std	Standard deviation of packet lengths.	Float
7	Timestamp	Time at which the packet is transmitted.	DateTime	17	Pkt Len Var	Variance of packet lengths.	float
8	Flow Duration	Duration of the flow in microseconds.	Integer	18	FIN Flag Cnt	Count of TCP FIN flags.	Integer
9	Tot Fwd Pkts	Total packets in the forward direction.	Integer	19	SYN Flag Cnt	Count of TCP SYN flags.	Integer
10	Tot Bwd Pkts	Total packets in the backward direction.	Integer	20	RST Flag Cnt	Count of TCP RST flags.	Integer

Figure 3. Sample features in the captured dataset

The final step in the data collection model is structuring the labeled data into a format suitable for machine learning. The data is organized into CSV files, where each row represents a captured flow or packet, and columns represent

the various features, such as IP addresses, ports, protocols, packet lengths, and labels. The feature extraction is performed utilizing the CICflowmeter tool. This structured format is essential for the subsequent training and evaluation of the IDS, as it ensures that the data is consistent, well-organized, and ready for use in machine learning algorithms.

4. Proposed framework

The proposed framework aims to ensure the secure and efficient operation of robotic devices by continuously monitoring network traffic, identifying potential threats, and differentiating between normal and malicious activities. An overview of the framework is illustrated in Figure 4. Building on this framework, algorithm 4.1 is developed and implemented through a series of steps that include data collection, preprocessing, model training, deployment, and real-time prediction.

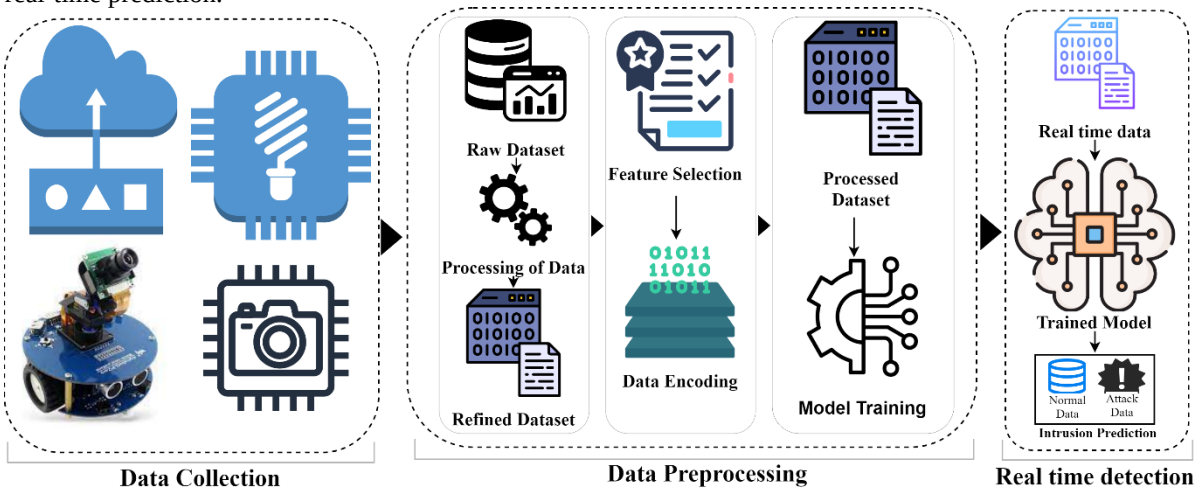

Figure 4. Proposed framework

The data collection process begins with the robotic device, which is configured as a client in the network, engaging in routine communication with a PC acting as the server. Network sniffing tools, such as Scapy, record the traffic in real-time and store it in Packet Capture (PCAP) files, which serve as the raw data source for the IDS.
During the preprocessing phase, relevant features are extracted from the raw data, such as packet length, inter-arrival time, flow duration, and the number of packets sent and received. The data is then cleaned and normalized to ensure consistency and scale. The labeled dataset is split into training and testing sets, and then ML learning algorithms is trained on the training set to learn the patterns that differentiate normal traffic from attacks.
After training and validation, the model is deployed into the robotic system for real-time monitoring. The deployed model continuously extracts features from incoming packets and uses the trained model to classify the traffic as either normal or attack. If the traffic is identified as malicious, the IDS triggers an alert.

4.1. Algorithm: Real-Time Intrusion Detection System

Input:

• Raw Dataset D_{raw}

Output:

• Real-time predictions $\hat{y}$ for captured data

Step 1: Load and Preprocess Data

1. Load Data

df ← read_csv(file_path)

2. Preprocess Data

a. Convert Labels:

y ← LabelEncoder.transform(categorical_labels)

b. Drop Columns:

df ← df.drop({'Flow ID', 'Src IP', 'Dst IP', 'Timestamp'})

c. Handle Infinite Values:

df.replace({∞, -∞}, NaN, inplace=True)

d. Handle Missing Values:

df ← df.fillna(0)

e. Standardize Features:

If training is True:

scaler ← StandardScaler.fit(X)

Else:

X_scaled ← scaler.transform(X)

f. Return:

X_scaled, y, scaler, LabelEncoder, feature_names

Step 2: Train-Test Split

3. Split Data

X_train, X_test, y_train, y_test ← train_test_split(X, y, test_size=0.3)

Step 3: Train the Model

4. Train Model

model ← RandomForestClassifier(n_estimators=100)

model.fit(X_train, y_train)

Step 4: Evaluate the Model

5. Evaluate Model

ŷ ← model.predict(X_test)

Print accuracy_score(ŷ, y_test)

Print classification_report(ŷ, y_test)

Step 5: Save Model and Scaler

6. Save Artifacts

joblib.dump(model, 'intrusion_detection_model.pkl')

joblib.dump(scaler, 'scaler.pkl')

joblib.dump(LabelEncoder, 'label_encoder.pkl')

joblib.dump(feature_names, 'feature_names.pkl')

Step 6: Real-Time Prediction

7. Load Model and Scaler

model, scaler, LabelEncoder, feature_names ← joblib.load(saved_files)

8. Make Predictions

Iterate through rows:

For each row in new_data_scaled do:

ŷ_i ← model.predict(row)

predicted_class_i ← LabelEncoder.inverse_transform(ŷ_i)

Print predicted_class_i

5. Results and discussion

In this study, the performance of three distinct ML algorithms KNN, DT, and RF for the real-time detection of intrusions within a robotic system. The metrics used to assess these algorithms include accuracy, precision, recall, and F1-score. The results are summarized in Table 2. The DT algorithm demonstrated the highest overall performance, achieving an accuracy of 96.61%. The precision was 96.76%, indicating that the model was highly effective in minimizing false positives, which is critical in an intrusion detection system (IDS).

The KNN algorithm also performed well, with an accuracy of 95.48%. The Random Forest (RF) algorithm, while still effective, exhibited the lowest performance among the three algorithms. It achieved an accuracy of 92.12%, which is notably lower than both DT and KNN. The analysis indicates that the DT algorithm is the most suitable for real-time IDS in a robotic environment, given its superior balance between precision and recall, which are critical for minimizing both false positives and false negatives. The KNN algorithm, while slightly less accurate, provides a simpler alternative that still offers high reliability. The Random Forest algorithm, although commonly effective in a wide range of applications, appears to be less suitable for this specific task due to its comparatively lower performance.

Also, Figure 5 (b), (c), and (d) depicts training and testing time for different ML techniques. Among the classifiers, KNN showed the shortest training time, indicating efficiency in building the model as depicted in Figure 5(c). DT and RF had relatively longer training times due to the internal complexity, with RF taking the most time among the three classifiers. DT exhibited the shortest testing time, making DT a fast option for classifying instances in the testing dataset.

Table 2. Prediction results

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
KNN	95.48%	95.4%	94.255%	95.45%
DT	96.61%	96.76%	95.67%	96.41%
Random Forest	92.12%	91.25%	90%	90.10%

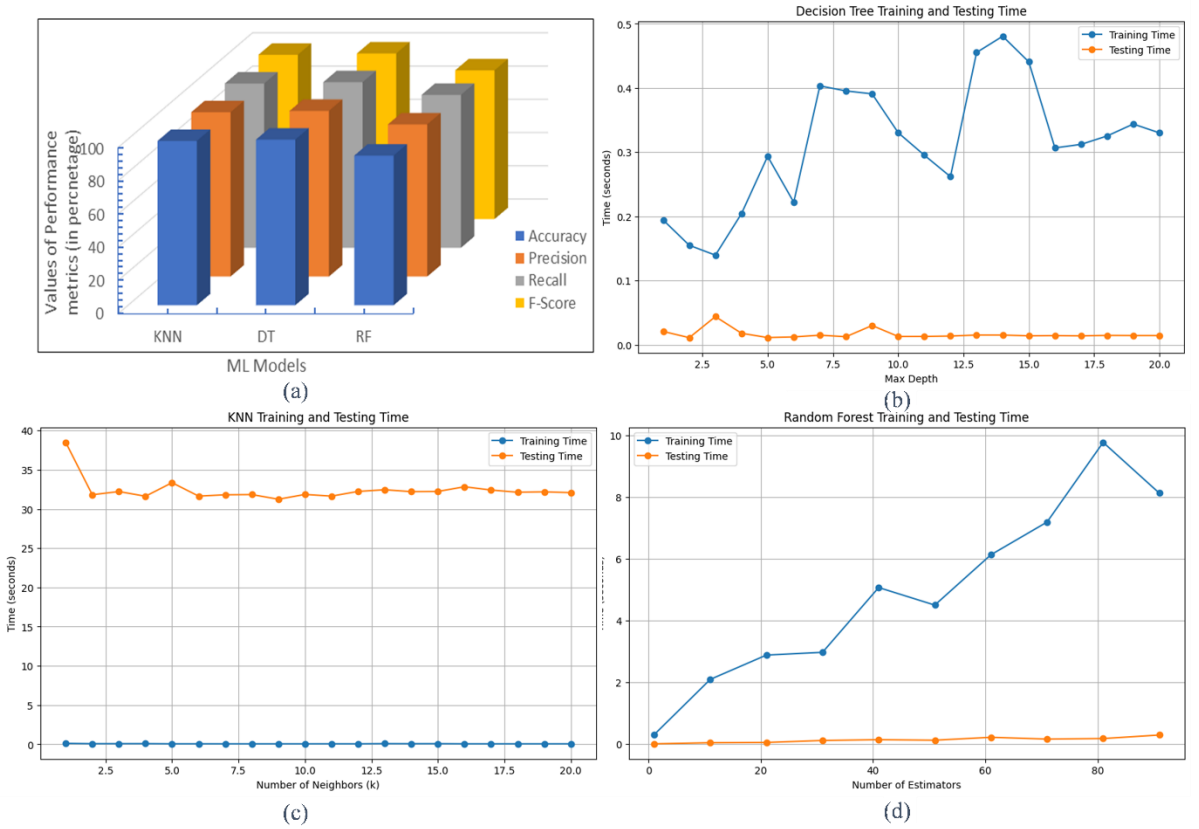

Figure 5. Performance evaluation (a) Multiple machine learning models in binary classification (b) DT training and testing time

6. Conclusion

As robotic systems become increasingly integral to multiple sectors ensuring the security against cyber threats has become a paramount concern. In this study, development and implementation of a Real-Time Intrusion Detection System (IDS) tailored for robotic systems operating in interconnected network environments is presented. The proposed IDS is designed to monitor network traffic continuously, detect potential threats, and distinguish between normal and malicious activities in real time. A comprehensive approach is employed, beginning with the creation of a testbed to simulate network traffic under both normal and attack scenarios. Data collection involved capturing and preprocessing this traffic, which included feature extraction, data cleaning, and normalization. Various machine learning models, including Decision Tree (DT), K-Nearest Neighbors (KNN), and Random Forest, are trained on the processed data. Among these, the Decision Tree classifier demonstrated superior performance, achieving an accuracy of 96.61% in distinguishing between normal and attack traffic. The deployed IDS proved effective in real-

time monitoring, providing real-time detection of anomalies with minimal latency. The results underscore the importance of implementing a lightweight and adaptable IDS framework that can secure robotic systems without imposing significant computational overhead. Future work will explore the integration of more advanced machine learning techniques and encryption methods to further enhance the system's detection capabilities and data security.

References

[1] J. P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab, "Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations," *Int J Inf Secur*, vol. 21, no. 1, pp. 115–158, Feb. 2022, doi: 10.1007/S10207-021-00545-8/FIGURES/7.

[2] D. A. G. S. M. N. Neerendra Kumar, "SECURITY ANALYSIS OF VULNERABILITIES IN ROBOTS," *Design Engineering*, pp. 4689–4700, Oct. 2021, Accessed: May 24, 2023. [Online]. Available: http://thedesignengineering.com/index.php/DE/article/view/5422

[3] K. Cottrell, D. B. Bose, H. Shahriar, and A. Rahman, "An empirical study of vulnerabilities in robotics," *Proceedings - 2021 IEEE 45th Annual Computers, Software, and Applications Conference, COMPSAC 2021*, pp. 735–744, Jul. 2021, doi: 10.1109/COMPSAC51774.2021.00105/VIDEO.

[4] A. Sayeed, C. Verma, N. Kumar, N. Koul, and Z. Illés, "Approaches and Challenges in Internet of Robotic Things," *Future Internet 2022, Vol. 14, Page 265*, vol. 14, no. 9, p. 265, Sep. 2022, doi: 10.3390/FI14090265.

[5] M. Pogliani *et al.*, "Security of Controlled Manufacturing Systems in the Connected Factory: The Case of Industrial Robots," *Journal of Computer Virology and Hacking Tech*, doi: 10.1007/s11416-019-00329-8.

[6] T. Bonaci, J. Herron, T. Yusuf, J. Yan, T. Kohno, and H. J. Chizeck, "To Make a Robot Secure: An Experimental Analysis of Cyber Security Threats Against Teleoperated Surgical Robots," Apr. 2015, Accessed: Jan. 04, 2023. [Online]. Available: https://www.researchgate.net/publication/329012011_Semi-Quantitative_Security_Risk_Assessment_of_Robotic_Systems

[7] S. Neupane *et al.*, "Security Considerations in AI-Robotics: A Survey of Current Methods, Challenges, and Opportunities," *IEEE Access*, vol. 12, pp. 22072–22097, 2024, doi: 10.1109/ACCESS.2024.3363657.

[8] H. Kabir, M. L. Tham, and Y. C. Chang, "Internet of robotic things for mobile robots: Concepts, technologies, challenges, applications, and future directions," *Digital Communications and Networks*, vol. 9, no. 6, pp. 1265–1290, Dec. 2023, doi: 10.1016/J.DCAN.2023.05.006.

[9] G. Lacava *et al.*, "Cybsersecurity issues in robotics," *J Wirel Mob Netw Ubiquitous Comput Dependable Appl*, vol. 12, no. 3, pp. 1–28, Sep. 2021, doi: 10.22667/JOWUA.2021.09.30.001.

[10] A. Botta, S. Rotbei, S. Zinno, and G. Ventre, "Cyber Security of Robots: a Comprehensive Survey," *Intelligent Systems with Applications*, p. 200237, May 2023, doi: 10.1016/J.ISWA.2023.200237.

[11] "Wireshark · Go Deep." Accessed: Apr. 20, 2024. [Online]. Available: https://www.wireshark.org/

[12] "GitHub - cometa/CICFlowMeter: CICFlowmeter-V3.0 (formerly known as ISCXFlowMeter) is a network traffic Bi-flow generator and analyzer for anomaly detection." Accessed: Aug. 27, 2024. [Online]. Available: https://github.com/cometa/CICFlowMeter

[13] T. Bonaci, J. Herron, T. Yusuf, J. Yan, T. Kohno, and H. J. Chizeck, "To Make a Robot Secure: An Experimental Analysis of Cyber Security Threats Against Teleoperated Surgical Robots," Apr. 2015, doi: 10.48550/arxiv.1504.04339.

[14] L. N. K. Duong *et al.*, "A review of robotics and autonomous systems in the food industry: From the supply chains perspective," *Trends Food Sci Technol*, vol. 106, pp. 355–364, Dec. 2020, doi: 10.1016/J.TIFS.2020.10.028.

[15] P. Kulshrestha and T. V. Vijay Kumar, "Machine learning based intrusion detection system for IoMT," *International Journal of System Assurance Engineering and Management*, vol. 1, pp. 1–13, Sep. 2023, doi: 10.1007/S13198-023-02119-4/FIGURES/12.

[16] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," *IEEE Access*, vol. 7, pp. 46717–46738, 2019, doi: 10.1109/ACCESS.2019.2906934.

[17] X. H. Nguyen, X. D. Nguyen, H. H. Huynh, and K. H. Le, "Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways," *Sensors 2022, Vol. 22, Page 432*, vol. 22, no. 2, p. 432, Jan. 2022, doi: 10.3390/S22020432.

[18] A. Meleshko, A. Shulepov, V. Desnitsky, E. Novikova, and I. Kotenko, "Visualization Assisted Approach to Anomaly and Attack Detection in Water Treatment Systems," *Water 2022, Vol. 14, Page 2342*, vol. 14, no.

15, p. 2342, Jul. 2022, doi: 10.3390/W14152342.
[19] H. Siddharthan and D. Thangavel, “A novel framework approach for intrusion detection based on improved critical feature selection in Internet of Things networks,” *Concurr Comput*, vol. 35, no. 1, p. e7445, Jan. 2023, doi: 10.1002/CPE.7445.
[20] H. Ghasemi and S. Babaie, “A new intrusion detection system based on SVM–GWO algorithms for Internet of Things,” *Wireless Networks*, pp. 1–13, Feb. 2024, doi: 10.1007/S11276-023-03637-6/TABLES/7.
[21] A. A. Alsulami, Q. Abu Al-Haija, A. Tayeb, and A. Alqahtani, “An Intrusion Detection and Classification System for IoT Traffic with Improved Data Engineering,” *Applied Sciences 2022, Vol. 12, Page 12336*, vol. 12, no. 23, p. 12336, Dec. 2022, doi: 10.3390/APP122312336.
[22] P. Kulshrestha and T. V. Vijay Kumar, “Machine learning based intrusion detection system for IoMT,” *International Journal of System Assurance Engineering and Management*, vol. 1, pp. 1–13, Sep. 2023, doi: 10.1007/S13198-023-02119-4/FIGURES/12.