Implementing Lightweight Intrusion Detection System on Resource Constrained Devices

Charles Stolz, Fuhao Li, and Jielun Zhang
Department of Computer Science, School of Electric Engineering & Computer Science
University of North Dakota
Grand Forks, ND 58202, USA
{charles.stolz, fuhao.li, jielun.zhang}@und.edu

2024 Cyber Awareness and Research Symposium (CARS) | 979-8-3503-8641-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/CARS61786.2024.10778716

***Abstract*—The rapid growth of Internet of Things (IoT) devices has increased the risk of network intrusions, which need effective security solutions for devices with low computational capability. However, Deep Learning based Intrusion Detection Systems (IDS) often demand substantial computational resources, which are unsuitable for the resource constrained IoT devices. To tackle this issue, we propose a lightweight IDS for Raspberry Pi operation. The proposed architecture includes traffic capture, threat detection, and alerting modules, utilizing signature and anomaly-based techniques. The signature-based module detects known attacks using predefined patterns, while the machine learning-based anomaly detection module identifies new threats by monitoring deviations from normal network behavior. The evaluation results show that our proposed scheme can detect a wide range of threats with minimal computational overhead.**

***Index Terms*—Intrusion Detection, IoT, Snort, Raspberry PI, Machine Learning**

I. Introduction

Internet of Things (IoT) devices are crucial in diverse applications, ranging from smart homes to industrial automation [1], [2]. Despite their benefits, the widespread use of IoT devices brings significant security challenges due to their operation in resource-limited environments, making them susceptible to various cyber threats [3]. Intrusion Detection Systems (IDS) are effective at protecting traditional networks from threats, but they require a lot of computational power, which is unsuitable for use on those IoT devices [4]–[6].

Recently, researchers have focused on deploying IDS solutions in resource-constrained environments [7]–[9]. These proposed schemes demonstrated that signature-based detection can be effectively deployed on those resource contrained devices, e.g., Raspberry Pi. However, these methods are inherently limited to detecting only known threats. They often fail to identify novel attack patterns, which is challenging in dynamic IoT environments. In this case, recent research has shifted to machine learning techniques for robust anomaly detection. For instance, Sivanandam *et al.* [10] and Katonová *et al.* [11] have employed various classifiers to improve detection accuracy and adaptability. However, while these models demonstrate high accuracy, they often require substantial computational resources, which can be challenging for resource-limited IoT devices. Therefore, there is a need for further optimization of these models to reduce their complexity and improve their computational efficiency. Moreover, most existing studies primarily rely on static datasets for evaluation, which does not fully capture the complexities and dynamics of real-world network environments. To ensure the practical applicability and reliability of IDS solutions, designing and testing a comprehensive detection process that can be seamlessly deployed and evaluated in live network scenarios is crucial.

In this paper, a lightweight IDS framework is proposed that combines machine learning techniques with signature-based and anomaly-based detection methods. This proposed system can detect both known and unknown threats by identifying deviations from normal network behavior. Additionally, our system is optimized for IoT networks and can be easily integrated into the existing IoT infrastructures. The proposed system consists of three main components: traffic capture, threat detection, and alerting modules. The signature-based detection module identifies known attacks using predefined patterns, and the anomaly-based detection module utilizes machine learning algorithms to detect novel threats by monitoring deviations in network behavior. This dual approach ensures comprehensive security coverage for IoT networks while maintaining minimal computational overhead. The proposed algorithm is evaluated using the CICIDS2017 dataset to demonstrate its detection performance. Moreover, we conducted real-world network attack simulations to evaluate the proposed algorithm in detecting various attack types and demonstrated the stability of the system under high-load conditions. The main contributions of this paper are: i) We proposed a lightweight IDS architecture that combines signature-based and anomaly-based detection methods, which can achieve high detection accuracy with low computational overhead. ii) We developed a comprehensive detection process, i.e., traffic capture, threat detection, and alerting modules, and the system was deployed on a Raspberry Pi platform to demonstrate the performance in resource-constrained environments. iii) We validate the system using the CICIDS2017 dataset along with the simulated attacks, and the evaluation results show its effectiveness in attack detection and maintaining stability under high-load conditions.

The remainder of this paper is organized as follows: Section II discusses the related work. Section III details our proposed methodology. Section IV presents the experimental setup and evaluation results. Section V concludes the paper and suggests possible future research directions.

TABLE I
SUMMARY OF STUDIES ON IDS IMPLEMENTATIONS

Study	Platform	Method	Machine Learning Techniques
A. Sforzin *et al.* [7]	Raspberry Pi 2	Signature-based	None
S. Tripathi *et al.* [8]	Raspberry Pi 3	Signature-based	None
P. A. M. Hambali *et al.* [9]	Raspberry Pi 3	Signature-based	None
N. Sivanandam *et al.* [10]	Raspberry Pi	Anomaly-based	Decision Tree (DT), Random Forest (RF)
E. A. Katonová *et al.* [11]	Raspberry Pi 4 Model B	Anomaly-based	Random Forest (RF), Multi-Layer Perceptron (MLP), Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Extreme Gradient Boosting (XGB)
P. R. Agbedanu *et al.* [12]	Raspberry Pi 2	Anomaly-based	Incremental Principal Component Analysis (IPCA), Self-Adjusting Memory K-Nearest Neighbors (SAM-KNN)
R. Sumanth *et al.* [13]	Raspberry Pi 3	Anomaly-based	K-Means Clustering
G. A. Morales *et al.* [14]	Raspberry Pi 2	Anomaly-based	Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbors (KNN), Random Forest (RF)
E. Ciklabakkal *et al.* [15]	Not specified	Anomaly-based	Autoencoder (AE), Self-Organizing Generative Adversarial Active Learning (SO-GAAL), Random Forest (RF), K-Means Clustering, One-Class Support Vector Machine (OCSVM), Isolation Forest (IF)

II. Related Work

This section provides an overview of existing research on IDS for IoT networks, focusing on signature-based and anomaly-based detection methods. We introduce the strengths and limitations of each approach to identify the gaps that our proposed solution addresses. The existing works are summarized in Table I.

A. Signature-Based IDS

Efforts have been made in Signature-based methods for detecting threats. For instance, A. Sforzin *et al.* [7] proposed a lightweight IDS based on Raspberry Pi 2 named RPiDS which utilizes Snort for signature-based detection. Their work has shown the feasibility of deploying the IDS on those resource-constrained devices, but the CPU utilization of their system reached 100% under the high data rates, which is challenging in high-traffic environments. S. Tripathi and R. Kumar [8] implemented an IDS using Snort and Tshark on a Raspberry Pi 3, showcasing effective deployment without memory or processing constraints. However, it lacked the capability to detect new threats. P. A. M. Hambali *et al.* [9] proposed a monitoring system for IoT networks using Raspberry Pi 3. While these works illustrate the potential for using lightweight IDS on Raspberry Pi, they rely solely on predefined signatures, which is ineffective against unknown attacks and new variants of known threats. This limitation has driven researchers to propose the machine learning based algorithms to adapt to dynamic and evolving attack patterns.

B. Machine Learning based IDS

To tackle the issues of signature-based systems, researchers have focused on machine learning based techniques for designing IDS [10]–[13], [16]–[18]. Additionally, some existing works have been focusing on applying them on resource-constrained devices.For instance, N. Sivanandam *et al.* [10] developed a machine learning-based IDS using Python on a Raspberry Pi, employing classifiers, i.e., Random Forest, Decision Tree, and Gradient Boosting Decision Trees to detect black hole and grey hole attacks in Bluetooth Low Energy (BLE) networks. E. A. Katonová *et al.* [11] extended this work by using a range of classifiers, including Random Forest, Multi-layer Perceptron, Support Vector Classifier, K-Nearest Neighbors, and XGBoost, on a Raspberry Pi 4 B. R. Sumanth *et al.* [13] utilized K-Means clustering to build IDS on Raspberry Pi 3, integrating rule-based mechanisms for IP blocking and demonstrate the potential of unsupervised learning in IDS. P. R. Agbedanu *et al.* [12] proposed an IDS using a combination of Incremental Principal Component Analysis (IPCA) and Self-Adjusting Memory K-Nearest Neighbors (SAM-KNN) on Raspberry Pi 2 and improved the detection performance. Nevertheless, the deep learning models can be further accelerated to enhance their efficiency to be more suitable for those resource-limited IoT devices. Additionally, designing and deploying a comprehensive real-world detection process is needed to validate the system under the practical conditions.

To address these issues, we propose a lightweight intrusion detection system framework that combines signature-based and anomaly-based detection methods, optimizing model complexity and algorithm design to achieve more efficient threat detection in resource-constrained environments. Our system is adaptable to various IoT scenarios, offering a novel solution to enhance the security of IoT devices. We also conducted real-world network attack simulations to validate the effectiveness of the proposed system in detecting multiple attack types and demonstrated its stability under high-load conditions.

III. Proposed System

The proposed IDS for IoT networks is designed to provide real-time monitoring and detection of suspicious activities using the computational capabilities of Raspberry Pi 4 Model B. The architecture integrates both signature-based and anomaly-based detection techniques to ensure comprehensive threat identification.

A. System Overview

The IDS architecture consists of the following primary components:

- Traffic Capture: Responsible for capturing network traffic data from IoT devices.
- Threat Detection: Implements the core detection mechanisms, employing both signature-based and machine learning-based anomaly detection techniques.
- Alerting: Generates alerts and notifications for identified threats, providing actionable intelligence to network administrators.

The system utilizes the following hardware and software configuration:

- Hardware:
  - Raspberry Pi 4 Model B with 4GB RAM and a 1.5GHz 64-bit quad-core CPU.
  - 64GB microSD card for storage.
  - Gigabit Ethernet network interface with a LAN cable.
- Software:
  - Operating System: Debian-based Linux (aarch64).
  - Traffic capture tools: 'tcpdump' for packet capture and 'PyShark' for packet analysis.
  - System monitoring tools: 'top' and 'vmstat' for real-time system resource utilization.
  - Programming language: Python 3.11.2 for implementing machine learning models and data processing scripts.
  - Machine Learning framework: scikit-learn 1.3.2

B. Traffic Capture

The traffic capture leverages 'tcpdump' to collect packet-level data from the network and 'PyShark' to process and analyze this data. This data serves as the input for the threat detection, which utilizes both Snort for signature-based detection and machine learning models for anomaly-based detection.

Algorithm 1 Traffic Capture with tcpdump and PyShark

```
Require: Network interface $interface$, Output file $output\_file$
Ensure: Packet capture and analysis
0: Initialize 'tcpdump' capture on $interface$ and write to $output\_file$
0: Capture live traffic from $interface$ and store in $output\_file$
0: Initialize PyShark to read from $output\_file$
0: for each packet in $output\_file$ do
0:     Process and analyze packet
0: end for
0: Apply filter to capture specific traffic type (e.g., HTTP)
0: Initialize filtered PyShark capture on $output\_file$ with filter $bpf\_filter$
0: for each packet in filtered capture from $output\_file$ do
0:     Process and analyze packet
0: end for=0
```

C. Signature-Based Threat Detection

The IDS incorporates signature-based techniques to identify suspicious activities. The system utilizes Snort, a widely-used open-source intrusion detection system, to match predefined threat patterns against network traffic. Snort employs a comprehensive database of known attack signatures to detect and alert on well-documented threats [7] [8].

D. Anomaly-Based Threat Detection

The anomaly-based threat detection focuses on identifying deviations from normal network behavior to detect novel or unknown threats. We propose using two machine learning models:

- Random Forest: Random Forest has been demonstrated to be effective in various IDS applications, as shown in the works of E. A. Katonová *et al.* [11], who implemented it in a Raspberry Pi-based IDS environment, and N. Sivanandam *et al.* [10], who utilized it for Bluetooth Mesh Networks.
- K-Nearest Neighbors (KNN): An effective algorithm for anomaly detection, particularly suitable for smaller datasets. KNN has been successfully applied in IDS by G. A. Morales *et al.* [14] in a collaborative inferencing system and E. A. Katonová *et al.* [11], who implemented KNN in a Raspberry Pi-based IDS.

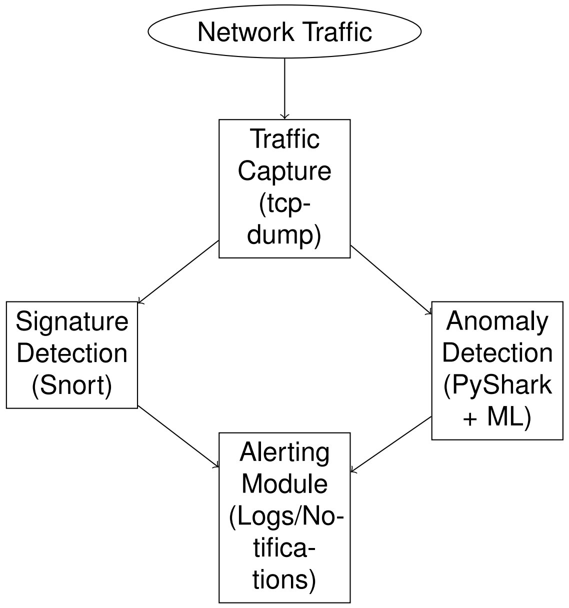
Network Traffic
Traffic Capture (tcp-dump)
Signature Detection (Snort)
Anomaly Detection (PyShark + ML)
Alerting Module (Logs/Notifications)

Fig. 1. Proposed IDS architecture.

E. Detection Capabilities

The IDS is designed to detect a range of suspicious activities, leveraging both the capabilities of Snort and machine learning techniques. This includes, but is not limited to, indicators of compromise (IoCs), unusual network traffic patterns, and specific attack signatures. Using Snort's extensive rule set, the IDS can identify known malicious IP addresses, domain names, and file hashes. Snort rules are regularly updated to include the latest IoCs, enabling the system to detect threats based on predefined patterns. Snort's effectiveness in detecting such indicators is well-documented in the literature [7] [8].

The proposed machine learning capabilities enhance the IDS by detecting anomalies in network traffic. This includes significant deviations in traffic volume, unexpected protocol usage, and abnormal communication patterns that may indicate novel or previously unknown threats. Machine learning models, such as Random Forest and KNN, are trained to recognize these deviations by analyzing historical network data. Studies by E. A. Katonová *et al.* [11] and G. A. Morales *et al.* [14] demonstrate the effectiveness of these models in similar contexts.

F. Alerting System

The alerting system combines Snort for signature-based detection and machine learning for anomaly detection on a Raspberry Pi. Detected anomalies are logged in '/var/log/attack alerts.log'. Future enhancements will add real-time notifications and a database to store alerts.

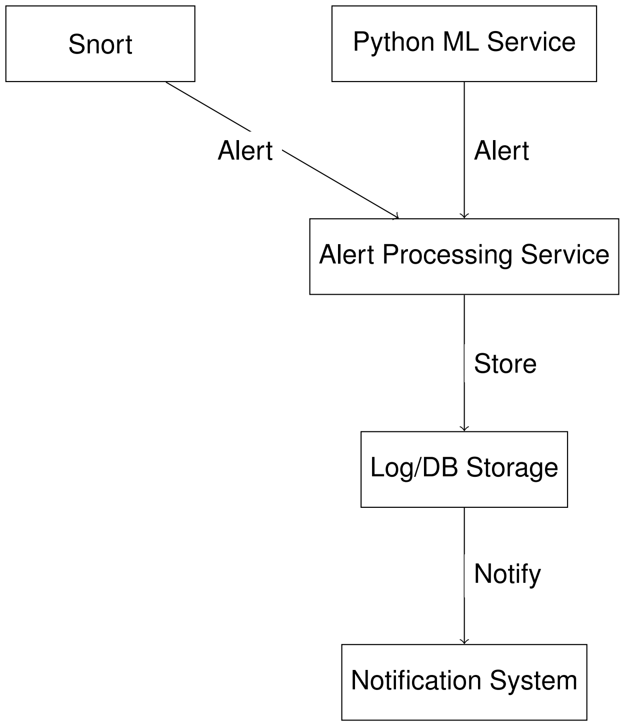

Fig. 2. Alert system architecture.

IV. Implementation and Results

A. Anomaly-Based Threat Detection Results

In this subsection, we present the results of our anomaly-based threat detection models trained on the CICIDS2017 dataset [19]. The primary focus is on the performance evaluation of the Random Forest and KNN models. The dataset includes a variety of attack types, which were used to train and evaluate the models, covering different categories such as brute-force attacks (FTP-Patator and SSH-Patator), Denial of Service (DoS) attacks (slowloris, Slowhttptest, Hulk, and GoldenEye), and web-based attacks (XSS and SQL injection). Additionally, it includes other significant threats like DDoS, port scanning, and botnet activities.

B. Evaluation Metrics

The models were evaluated based on Accuracy, Precision, Recall, and F1-Score. Accuracy measures the overall correctness of the predictions, and Precision reflects the proportion of true positive predictions out of all positive predictions made. Recall assesses the model's ability to identify true positive instances among all actual positives. The F1-Score is calculated based on Precision and Recall. It provides a balanced measure of both correctness and completeness of the model's predictions.

1) Top Features Used: The models were trained using the top 20 features selected from the dataset based on their importance. These features are shown in Table II.

TABLE II
Top 20 Features Used for Model Training.

Feature Name	Description
Fwd Packet Length Mean	Mean length of forward packets
Bwd Packet Length Std	Standard deviation of backward packet length
Bwd Packet Length Min	Minimum length of backward packets
Fwd Packet Length Min	Minimum length of forward packets
Packet Length Variance	Variance of packet length
Destination Port	Port number of the destination
Min Packet Length	Minimum packet length
Packet Length Mean	Mean packet length
Average Packet Size	Average size of packets
Packet Length Std	Standard deviation of packet length
Bwd Packets/s	Backward packets per second
URG Flag Count	Count of URG flags in packets
Avg Bwd Segment Size	Average size of backward segments
Max Packet Length	Maximum length of packets
Avg Fwd Segment Size	Average size of forward segments
Init Win bytes forward	Initial window bytes in forward direction
Fwd IAT Std	Standard deviation of forward inter-arrival time
Flow Packets/s	Flow packets per second
ACK Flag Count	Count of ACK flags in packets
PSH Flag Count	Count of PSH flags in packets

These features were selected based on their contribution to the overall model performance using a Random Forest Classifier to rank feature importance according to the average decrease in node impurity [20].

2) Random Forest Model: The Random Forest model was evaluated on the CICIDS2017 dataset using the top 20 selected features [19]. The performance of the model is summarized in Table III, showcasing the precision, recall, F1-score, and support for both benign traffic and attack traffic. The confusion matrix for the Random Forest model is presented in Table IV.

TABLE III
Evaluation performance of Random Forest based IDS.

	Precision	Recall	F1-Score	Support
Benign Traffic	0.9906	0.9717	0.9811	557646
Attack Traffic	0.9722	0.9908	0.9814	557646
Accuracy	0.9813			
Macro Avg	0.9814	0.9813	0.9813	1115292
Weighted Avg	0.9814	0.9813	0.9813	1115292

TABLE IV
Confusion Matrix of Random Forest based IDS.

	Predicted: Benign	Predicted: Attack
Actual: Benign	541870 (TN)	15776 (FP)
Actual: Attack	5118 (FN)	552528 (TP)

3) KNN Model: The KNN model was also evaluated on the CICIDS2017 dataset using the same top 20 features [19]. The model's performance is summarized in Table V, with the confusion matrix presented in Table VI.

TABLE V
Evaluation performance of KNN based IDS.

Class	**Precision**	**Recall**	**F1-Score**
Benign Traffic	0.9880	0.9732	0.9806
Attack Traffic	0.9736	0.9882	0.9809
Accuracy	0.9807		
Macro Avg	0.9808	0.9807	0.9807
Weighted Avg	0.9808	0.9807	0.9807

TABLE VI
Confusion Matrix of KNN based IDS.

	Predicted: Benign	Predicted: Attack
Actual: Benign	542728 (TN)	14918 (FP)
Actual: Attack	6588 (FN)	551058 (TP)

C. Signature-Based Threat Detection Results

The detection system was tested against three common network attacks using hping3 and Hydra. Table VII summarizes the attacks, tools, and verification methods.

TABLE VII
Threat Detection Test Results.

Attack Type	**Tool**	**Verification**
Ping of Death	hping3	Monitored ICMP alerts
SYN Flood	hping3	Monitored SYN alerts
SSH Brute-Force	Hydra	Monitored failed login alerts

The details of the detection system are shown in the analysis of the logged alert file, which can be found in Lightweight-IDS/results/snort-experiment/alert in our GitHub [21]. Listing 1 shows examples of network traffic alert logs. For the Ping of Death attack, the log recorded a total of 22 ICMP ECHO requests and replies, each accompanied by "Ping detected" alerts. During the SYN Flood attack, the system logged ten instances of repeated SYN packets, generating corresponding "SYN Flood detected" alerts for each. Finally, the SSH Brute-Force attack was captured with one alert, where multiple failed login attempts were detected, with a corresponding "SSH Brute-force attempt detected" alert.

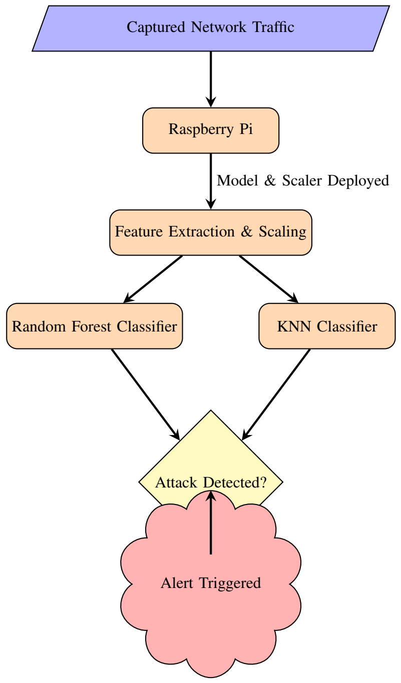

Fig. 3. Overview of Model Deployment and Attack Detection Process on Raspberry Pi.

```
[**] [1:1000001:1] Ping detected [**]
[Priority: 0]
08/11-18:48:52.607618 192.168.0.18 ->
    192.168.0.45
ICMP TTL:64 TOS:0x0 ID:28197 IpLen:20 DgmLen:84
     DF
Type:8 Code:0 ID:2 Seq:2 ECHO

[**] [1:1000002:1] SYN Flood detected [**]
[Priority: 0]
08/11-18:49:36.421284 192.168.0.18:2459 ->
    192.168.0.45:80
TCP TTL:64 TOS:0x0 ID:35723 IpLen:20 DgmLen:40
******S* Seq: 0x2AD81783  Ack: 0x2338A8EF  Win:
     0x200  TcpLen: 20

[**] [1:1000007:1] SSH Brute-force attempt
    detected [**]
[Priority: 0]
08/11-19:07:28.398231 192.168.0.18:39432 ->
    192.168.0.45:22
TCP TTL:64 TOS:0x0 ID:40965 IpLen:20 DgmLen:74
     DF
***AP*** Seq: 0x7F2BD8C8  Ack: 0x10148946  Win:
     0x1F6  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2691466795
    3149052523
```

Listing 1. Network traffic alert logs example.

D. Computational Efficiency

The computational efficiency of the system was assessed during a load test designed to simulate high traffic conditions and evaluate the performance of key security and machine learning processes. The test was conducted over an extended period, with the system handling a continuous SYN Flood attack to stress the network stack.

During the load test, two primary processes were monitored for their computational resource consumption:

- `snort` (PID: 64034) – This process is responsible for network intrusion detection and was actively monitored for malicious activities, including the SYN Flood attack. It consumed approximately 54.3% of the CPU, indicating its intensive processing of network packets.
- `python3` (PID: 65489) – This process is running a machine learning tool designed for anomaly detection. It utilized 73.3% of the CPU, reflecting the computational demands of real-time data processing and analysis.

TABLE VIII
CPU USAGE DURING LOAD TEST.

Process	CPU Usage (%)
`snort` (PID: 64034)	54.3
`python3` (PID: 65489)	73.3

Memory usage was also monitored during the load test. Although free memory remained stable at 124,136 KB and cached memory was 2,175,088 KB, these values did not impact the ability of the system to manage the SYN Flood attack. The SYN Flood attack generated a high volume of TCP SYN packets, which placed significant strain on the system's network and processing capabilities. Despite the heavy load, the system maintained operational stability, with key processes effectively managing the incoming traffic and performing real-time analysis. The code used for the experiments in this paper is available on GitHub. The repository also includes logs and reports from the experiments [21].

V. CONCLUSIONS

Accurate and fast security solutions are needed with the fast growth of IoT devices. Traditional IDSs often require substantial computational resources, making them unsuitable for resource-constrained IoT devices. To tackle this issue, we proposed a lightweight IDS for the Raspberry Pi environment. The system combined signature-based detection with Snort and anomaly-based detection utilizing machine learning models, and the experimental results demonstrated that our implementation achieved high accuracy in detecting common attacks such as Ping of Death, SYN Flood, and SSH brute-force attempts with decent processing efficiency.

REFERENCES

[1] S. L. Keoh, S. S. Kumar, and H. Tschofenig, "Securing the internet of things: A standardization perspective," *IEEE Internet of things Journal*, vol. 1, no. 3, pp. 265–275, 2014.

[2] L. Farhan, S. T. Shukur, A. E. Alissa, M. Alrweg, U. Raza, and R. Kharel, "A survey on the challenges and opportunities of the internet of things (iot)," in *2017 Eleventh International Conference on Sensing Technology (ICST)*. IEEE, 2017, pp. 1–5.

[3] A. Adnan, A. Muhammed, A. A. Abd Ghani, A. Abdullah, and F. Hakim, "An intrusion detection system for the internet of things based on machine learning: Review and challenges," *Symmetry*, vol. 13, no. 6, p. 1011, 2021.

[4] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the internet of things," *IEEE access*, vol. 7, pp. 42 450–42 471, 2019.

[5] J. Zhang, S. Liang, F. Ye, R. Q. Hu, and Y. Qian, "Towards detection of zero-day botnet attack in iot networks using federated learning," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 7–12.

[6] S. Roy, J. Li, B.-J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for iot networks," *Future Generation Computer Systems*, vol. 127, pp. 276–285, 2022.

[7] A. Sforzin, F. G. Mármol, M. Conti, and J.-M. Bohli, "Rpids: Raspberry pi ids — a fruitful intrusion detection system for iot," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, 2016, pp. 440–448.

[8] S. Tripathi and R. Kumar, "Raspberry pi as an intrusion detection system, a honeypot and a packet analyzer," in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*. IEEE, 2018, pp. 80–85.

[9] P. A. M. Hambali, M. R. Effendi, E. A. Z. Hamidi *et al.*, "Prototype design of monitoring system base tranceiver station (bts) base on internet of things," in *2020 6th International Conference on Wireless and Telematics (ICWT)*. IEEE, 2020, pp. 1–6.

[10] N. Sivanandam and T. Ananthan, "Intrusion detection system for bluetooth mesh networks using machine learning," in *2022 International Conference on Industry 4.0 Technology (I4Tech)*. IEEE, 2022, pp. 1–6.

[11] E. Katonová, P. Nehila, P. Fecil'Ak, O. Kainz, M. Michalko, F. Jakab, and R. Petija, "Implementation of ids functionality into iot environment using raspberry pi," in *2023 21st International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 2023, pp. 289–294.

[12] P. R. Agbedanu, R. Musabe, J. Rwigema, I. Gatare, and Y. Pavlidis, "Ipca-samknn: A novel network ids for resource constrained devices," in *2022 2nd International Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*. IEEE, 2022, pp. 540–545.

[13] R. Sumanth and K. Bhanu, "Raspberry pi based intrusion detection system using k-means clustering algorithm," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2020, pp. 221–229.

[14] G. A. Morales, J. Xu, D. Zhu, and R. Slavin, "Lightweight collaborative inferencing for real-time intrusion detection in iot networks," in *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*. IEEE, 2022, pp. 392–400.

[15] E. Ciklabakkal, A. Donmez, M. Erdemir, E. Suren, M. K. Yilmaz, and P. Angin, "Artemis: An intrusion detection system for mqtt attacks in internet of things," in *2019 38th Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2019, pp. 369–3692.

[16] F. Li, M. Ali, and J. Zhang, "Cyber attack detection in iot using enhanced stream classification algorithm," in *International Symposium on Intelligent Computing and Networking*. Springer, 2024, pp. 391–402.

[17] J. Zhang, F. Li, and F. Ye, "An ensemble-based network intrusion detection scheme with bayesian deep learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[18] M. Ali and J. Zhang, "Explainable artificial intelligence enabled intrusion detection in the internet of things," in *International Symposium on Intelligent Computing and Networking*. Springer, 2024, pp. 403–414.

[19] R. Panigrahi and S. Borah, "A detailed analysis of cicids2017 dataset for designing intrusion detection systems," *International Journal of Engineering & Technology*, vol. 7, no. 3.24, pp. 479–482, 2018.

[20] S. Raschka, Y. H. Liu, and V. Mirjalili, *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd, 2022.

[21] C. Stolz, "Lightweight ids," GitHub repository, 2024. [Online]. Available: https://github.com/rylandtikes/Lightweight-IDS.git