electronics

Article

Distributed Deep Neural-Network-Based Middleware for Cyber-Attacks Detection in Smart IoT Ecosystem: A Novel Framework and Performance Evaluation Approach

Guru Bhandari *, Andreas Lyth, Andrii Shalaginov and Tor-Morten Grønli †

Department of Technology, School of Economics, Innovation, and Technology, Kristiania University College, 0107 Oslo, Norway

* Correspondence: guruprasad.bhandari@kristiania.no

† Current address: Mobile Technology Lab (MOTEL), Department of Technology, Kristiania University College, Kirkegata 24-26, 0153 Oslo, Norway.

Abstract: Cyberattacks always remain the major threats and challenging issues in the modern digital world. With the increase in the number of internet of things (IoT) devices, security challenges in these devices, such as lack of encryption, malware, ransomware, and IoT botnets, leave the devices vulnerable to attackers that can access and manipulate the important data, threaten the system, and demand ransom. The lessons from the earlier experiences of cyberattacks demand the development of the best-practices benchmark of cybersecurity, especially in modern Smart Environments. In this study, we propose an approach with a framework to discover malware attacks by using artificial intelligence (AI) methods to cover diverse and distributed scenarios. The new method facilitates proactively tracking network traffic data to detect malware and attacks in the IoT ecosystem. Moreover, the novel approach makes Smart Environments more secure and aware of possible future threats. The performance and concurrency testing of the deep neural network (DNN) model deployed in IoT devices are computed to validate the possibility of in-production implementation. By deploying the DNN model on two selected IoT gateways, we observed very promising results, with less than 30 kb/s increase in network bandwidth on average, and just a 2% increase in CPU consumption. Similarly, we noticed minimal physical memory and power consumption, with 0.42 GB and 0.2 GB memory usage for NVIDIA Jetson and Raspberry Pi devices, respectively, and an average 13.5% increase in power consumption per device with the deployed model. The ML models were able to demonstrate nearly 93% of detection accuracy and 92% f1-score on both utilized datasets. The result of the models shows that our framework detects malware and attacks in Smart Environments accurately and efficiently.

Keywords: cybersecurity; machine learning; malware and attacks; internet of things; IoT security; artificial neural network

check for updates

Citation: Bhandari, G.; Lyth, A.; Shalaginov, A.; Grønli, T.-M. Distributed Deep Neural-Network-Based Middleware for Cyber-Attacks Detection in Smart IoT Ecosystem: A Novel Framework and Performance Evaluation Approach. *Electronics* **2023**, *12*, 298. https://doi.org/10.3390/electronics12020298

Academic Editor: Paris Kitsos

Received: 30 November 2022
Revised: 29 December 2022
Accepted: 3 January 2023
Published: 6 January 2023

1. Introduction

Nowadays, industries and individuals are increasingly adopting IoT devices and smart technologies for the convenience of day-to-day life, enabling the optimization of the process which makes the environment more efficient and greener as a side effect. The Smart Environment is a technology-enabled phenomenon that offers better, user-friendly and efficient IoT-based environments with a specific focus on sustainable future both in rural and city-wise contexts [1]. The IoT Analytics (https://iot-analytics.com/number-connected-iot-devices/; accessed on 20 August 2022) reports that the market growth of IoT connections has increased to 18% in 2022 till May, which is 14.4 billion globally, despite the impact of the global chip shortage and supply disruptions due to COVID-19 and the war in Ukraine. However, ensuring the security of these middleware devices is becoming a major challenge.

The AT&T Alien Labs™ recently discovered BotenaGo malware that exposed millions of IoT devices [2]. In March 2021, a group of hackers accessed and controlled thousands of Verkada security cameras and exposed user credentials publicly on the internet [3]. The 2022 Cyber Threat Report of the *SonicWall* (https://www.sonicwall.com; accessed on 14 August 2022) the cybersecurity research lab reported a continuously increasing trend of IoT malware threats, with more than 60 million attacks recorded in 2021, which is the highest ever recorded in a single year. IoT malware attacks in particular increased by 6%, with routers being the most targeted devices [4]. Unfortunately, more than 80% of the top 10 vulnerabilities of the last year have repeated this year. These security issues put pressure on enterprises, organizations, and governments to acquire effective threat intelligence to protect the systems against malware and attacks [5].

Various cybersecurity-related challenges are observed in Smart Environments with IoT infrastructure, such as the following: (i) it is hard to implement end-point protection (antivirus, intrusion detection systems, and firewalls) in IoT resource-constrained environments, especially in an energy-efficient way, (ii) it is impossible to have synchronous real-time communication, and (iii) the diversity of devices makes the applicability of a single system generally limited and far from a universal design. Several existing studies [5–8] have proposed AI models for cybersecurity; however, the majority of them have considered only a portion of the dataset or targeted only a few attacks. Therefore, in this study, we have proposed an approach with a framework to discover malware attacks on IoT devices using AI-enabled approaches covering diverse and distributed scenarios in Smart Environments. In our work, the choice of hardware for setting up the IoT network is representative of typical industrial use and is available off the shelf. Our approach will utilize a multi-agent network of AI models, where the most cumbersome will be trained in the Cloud environment, and the rest can be trained in Fog/Dew and subsequently deployed on Edge devices.

The main contributions of the article are as follows: (a) the new approach discovers attacks and malware on the IoT devices using an AI-enabled approach; (b) it facilitates live tracking of the streamed network traffic for the detection of malware and attacks; (c) the approach helps to locate the security issues and the concerned affected devices that assists in reducing the maintenance effort; and (d) the approach presents the performance and concurrency testing of the IoT devices. The observation suggests that the proposed method is feasible for efficient implementation in real-world in-production Smart Environments.

The remainder of the paper is organized in the following manner: Section 2 discusses the related work to the domain and how our work contributes differently than existing AI-enabled cybersecurity solutions. Section 3 presents the proposed AI-enabled detection method along with the utilized test cases, IoT gateways, and AI model transfer approach. After that, Section 4 presents the experimental setup and results of both AI-specific performance measures and hardware consumption measures to benchmark the proposed approach. Next, we discuss our observations, limitations, and future directions in Section 5. Finally, the work is concluded in Section 6.

2. Related Work

Smart Environment is a technology-enabled circumstance that offers better, user-friendly and efficient IoT infrastructure with a focus on greener and more sustainable future [9]. Used devices, components, and generated data are subject to the user's needs with sustainability and adaptation as major targets [10,11]. To defend the IoT infrastructure against known cyber-attacks, various open-source and commercial software solutions, such as anti-viruses, firewalls, anti-pattern detection approaches, and security protocols, help to enhance cybersecurity.

In the existing literature, IoT attack and malware detection methods are often proposed for specific use cases—intrusion detection [12–14], malicious traffic detection [15], anomaly detection [16], and botnet detection [17]—using various AI methods: deep neural network (DNN) [14,18–21], cyberthreat intelligence [22,23], and knowledge graph [24–26]. The centralized deep learning methods are also considerably used for attack detection in IoT network traffic data [12,15,19,20,27].

A considerable number of studies and related datasets for cybersecurity of IoTs are proposed by the community [6–8]. The specific characteristics of each layer of IoT system and network are subject to many critical vulnerabilities reported in several surveys studies [6–8,28–30]. So far, adequate cybersecurity countermeasures are not well-established for IoT devices due to their constraint in terms of limited power, memory, and processing capabilities.

The demand for internet data traffic is rapidly increasing for different data-driven Smart Environment applications. The network traffic predictions focus on anticipating future traffic, utilizing previous traffic data [31]. Using IoT malware network traffic data, Bendiab et al. [32] proposed an AI-enabled detection approach at the package level, reducing the time of detection using deep learning methods. Their network data consist of 1000 pcap files of normal and malware traffic collected from different network traffic sources.

A general IoT ecosystem normally includes *IoT nodes*, end-point devices with limited computational capabilities (CPU $\approx$ MHz) that are used to collect data, send measurements and often work using batteries or solar panels. The *IoT gateways* are portable devices, having the functionality of low-end personal computers (CPU $\approx$ GHz), performing data processing and aggregation tasks. Moreover, the devices follow different proprietary and open communication protocols, unique data storage standards, operational logic [9], different operating systems, and dependencies. From the cybersecurity perspective, data can be protected on the Linux-based IoT gateway using tools available for Unix such as ClamAV (Clam AntiVirus) for malicious software detection, encryption available for Linux [33] and RPiDS (Raspberry Pi IDS) [34] for an intrusion detection system (IDS). However, the application of such measures on *IoT end-nodes* is extremely limited. There is no OS, yet rather firmware that defines a strict routine of initialization function SETUP() and the iterative function LOOP() [35]. The only cybersecurity solution that is available and being tested for AVR is the Arduino Crypto library, designed to protect the information by application of various standard encryption methods [36]. Therefore, it is necessary to establish an understanding of what kind of data analytics for security can be run on IoT nodes and what should be moved to the IoT gateway for the sake of ensuring primary services availability and data protection [37]. Our previous work [38] provides a framework for the data-driven cyberattack prediction method using several intelligent methods. The method analyzes the complexity of power consumption and bandwidth in deploying AI models to IoT devices.

It can be seen that there are only a few relevant solutions available for the IoT nodes, such as those available for Arduino, when it comes to the implementation of machine learning (ML) models [39,40]. ML is a domain of AI ensembling a set of models that can be trained from previous data to be able to classify, detect or find unseen patterns. There is also Q-behave [41], an ML library for Arduino, dedicated to training an Arduino to learn simple patterns from the user and is not exactly an implementation of community-accepted ML models. On the other hand, there exist implementations of artificial neural networks (ANNs), such as ArduinoANN [42] or Neurona [43]. So, there can be seen a few mostly experimental implementations, yet no widely used software products, where ML is used heavily on IoT nodes, rather than IoT gateways. Therefore, there is a need for evaluating the concurrency and resource utilization of the IoT devices to measure the effectiveness of the AI model on these tiny devices. It can be clearly seen that the resource utilization measures of the ML model are missing in the existing studies.

The method analyzes multi-level network traffic data from various IoT devices such as Raspberry Pi, Arduino Nano, Arduino UNO, NVIDIA Jetson, Orange Pi, and several other sensors and actuators. The collected data are labeled as benign samples and attack samples. The AI methods are known to be good at predicting the binary and multi-class output (y = dependent variable) from the given features of the sample (X = independent variables).

The ML prediction is finding y based on X, such as $y = f(X)$, where $X = x_1, x_2, x_3, \ldots$ are the multiple features. In the case of network traffic data, the output depends on whether the sample packet belongs to one of the attack types or is benign based on the features of the packets, i.e, protocol, packet size, depth, duration, seen_bytes, and total_bytes. Furthermore, deep neural networks (DNNs) are multi-layer neural networks that are better for self-learning capabilities. The multi-layering allows models to become more efficient at learning complex features and performing more intensive computation tasks executing many complex operations simultaneously [44]. Similarly, for attacks and malware prediction, DNN can perform better in understanding the characteristics of the incoming and outgoing networking traffic information of the network.

Our proposed approach not only presents the identifying the suitable ML model to the given data, but also the effectiveness of the model while deploying to widely used IoT community devices, such as Rasberry Pi, NVIDIA Jetson, etc. The performance and concurrency measurement of the IoT devices with the ML model checks how efficiently the AI applications perform in IoT cybersecurity.

3. AI-Based Framework for IoT Cybersecurity

The Smart Environments consist of the implementation of interconnected IoT devices, such as Arduino, Raspberry Pi, Banana Pi, and NVIDIA Jetson, etc. The proposed framework method as shown in Figure 1 resembles the data collection process, feature engineering, inference of the AI model, and deployment of the trained model with real-world test cases. The first step consists of data collection from the IoT ecosystem and composing the dataset. After that, the preprocessing of the dataset to make it into a feedable form for the ML the method takes place. We apply various ML methods on different splits of the dataset to construct the trained model which is then used for the prediction of malware and attack types.

The following subsections describe brief information on four major areas of focus. Firstly, an overview of AI-based middle architecture (Section 3.1), and then AI techniques used for experiments on the network traffic data (Section 3.2) and then different test cases (Section 3.3), finally an IoT gateway, and an AI model transfer approach (Section 3.4. The framework workflow for IoT cybersecurity is associated with different software steps and hardware components.

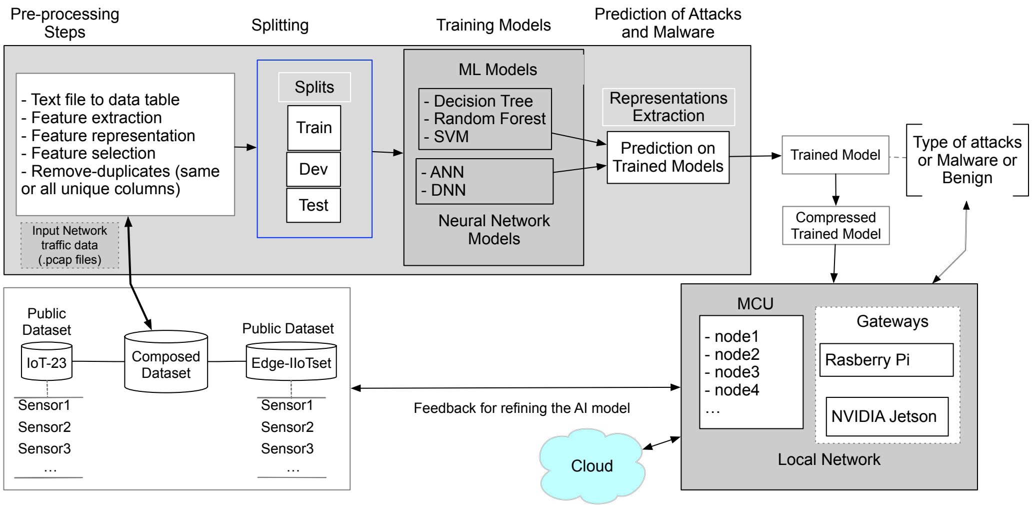


Figure 1. The framework workflow of the proposed method for IoT cybersecurity interconnecting different software and hardware components.

3.1. *AI-Based Middleware Architecture*

The experimental platform includes the emulation of real-world applications to be found in modern Smart Environments and uses open-source components. Raspberry Pi and

NVIDIA Jetson are taken as the gateways for logging the information that constantly listens to the router connected with the deployed IoT devices. The IoT network uses the MQTT (MQ Telemetry Transport) (https://mqtt.org/; accessed on 5 July 2022) message protocol with Raspberry Pi4 Mod B set up as a gateway and MQTT-broker, relaying messages between the nodes. Arduino UNO WiFi R2 (ATmega4809), Arduino NANO 33 IoT (Cortex-M0 32-bit SAMD21) and ESP32 Huzzah (Tensilica LX6) are used as sensor and actuator nodes, publishing and subscribing messages to topics on the broker. The model deployed on the Raspberry Pi gateway will monitor this traffic. Publishers (sensors) in the network will publish topics in one of two ways. Either by (i) publishing messages on a strictly timed interval or (ii) publishing messages to the topic in case of an event triggered by the algorithm. Subscribers (actuators) will query the broker for updates on topics in strictly timed intervals.

Figure 2 shows the interoperability of the Smart Environment and IoT cybersecurity [9], contributes as a basis of real-world use-case scenarios in this study. As shown in the figure, there are several test cases, such as water management, waste management, etc., and all are in the IoT ecosystem. The advanced data analytics ML implementation works as an indicator of compromise in malware and attack scenarios. The compromised information will be forwarded to the incident response system to implement appropriate control measures.

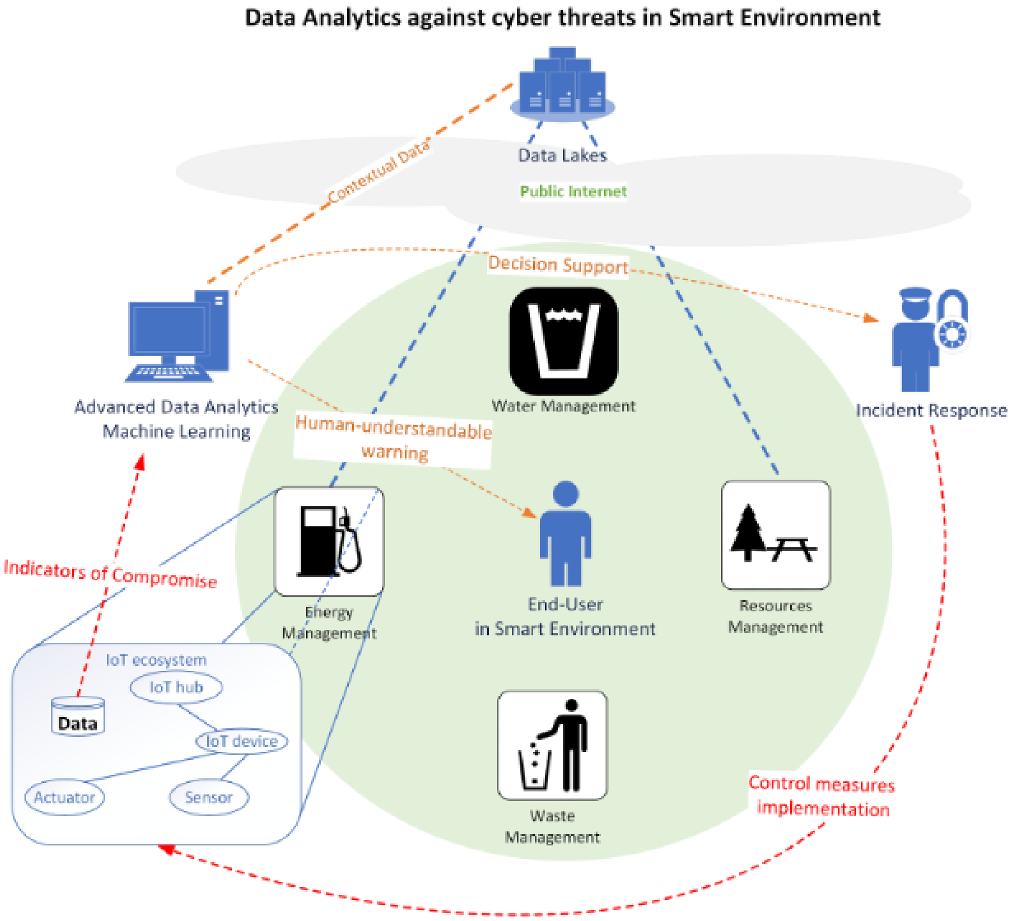


Figure 2. The interoperability of Smart Environment and IoT cybersecurity.

3.2. Artificial Intelligent Methods

As the predictions are made on a number of malware and attack types, all the AI methods used in this study are of multi-class types. Different AI models—deep neural networks (DNNs), support vector machine (SVM) random forest (RF), decision tree (DT), gradient boosting (GB), and naive Bayes (NB)—are used to train the dataset for the classification of malware and attacks. We used several imbalance handling methods to improve the performance, making the training process more stable and easy, which considerably reduces the model training time. The input data of IoT network traffic and the data generated by our IoT devices go through certain pre-processing steps. For each of the algorithms, the processed data were split into an 80:20 ratio between the training and testing sets.

3.3. Test-Cases for the Realization of Smart Environments

As a test bed for the project, six typical IoT use cases were developed for demonstration purposes (smart waste bin, temperature meter, passenger counter, luminance meter, weather station, and indoor air quality meter). Due to their frequent use in the field, Wi-Fi was chosen as the wireless interface, and MQTT was chosen as the messaging protocol. Devices commonly found in the industry were chosen for nodes and gateways in these use cases. The devices and their specification can be found in Table 1. A specific test case was designed for electrical testing and cross-platform testing. In this test case, 10 AUWR2 devices were set up to send messages to an MQTT broker every ten seconds. Figure 3 shows the information on hardware platforms implemented in the study.

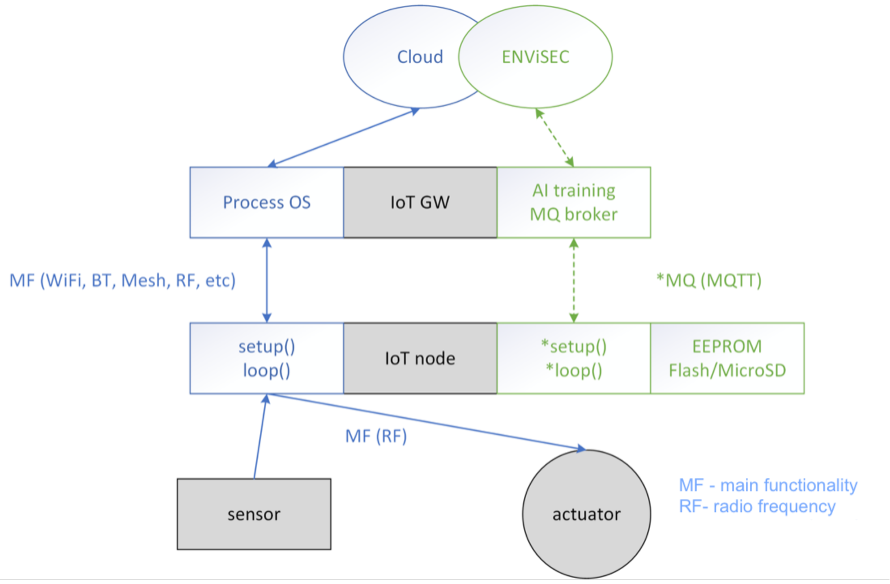


Figure 3. The proposed middleware architecture for Smart Environment and IoT cybersecurity (*-MQTT)

Table 1. Hardware platforms information.

Device Name	Arduino UNO WiFi R2	Arduino Nano 33 IoT	ESP32 Huzzah	Raspberry Pi mod 4
Used abbreviation	AUWR2	Nano	ESP32	RPi
Chip	ATmega4809	SAMD21 Cortex M0+	Tensilica LX6 microcontroller	ARM Cortex-A72 processor
Memory	Flash: 48 KB, SRAM: 6144 Bytes, EEPROM: 256 Bytes	FLASH: 256 KB, SRAM: 32 KB	448 KB ROM, 520 KB SRAM, 16 KB SRAM in RTC, QSPI	8 GB RAM
Operating voltage	5 V	3.3 V	2.3 V to 3.6 V	5 V
Connectivity	WiFi, BLE	WiFi, BLE	WiFi, BLE	WiFi, BLE

3.4. IoT Gateway and AI Model Transfer Approach

The IoT nodes in the network should belong to the same network and connect to an IoT gateway. The gateway is an access point to retrieve traffic data to implement the AI-enabled model. All the sensors and actuators are linked to a common IoT interface and the access point. Figure 4 shows the AI-enabled model transfer approach to perform prediction on IoT devices, where the high inference tasks take place in the server host or cloud, which minimizes the burden of huge data processing in IoT-node, thus reducing energy consumption. The two procedures, (A) publishing the processed JSON data to the localhost, and (B) fetching the published data in a IoT gateway, are as presented in the following Algorithm 1.

Algorithm 1 AI-enabled model transfer approach.

```
procedure (A): PUBLISH THE JSON TO THE LOCALHOST:
    generate the network traffic log file (.log) by running the Zeek tool
    convert .log file into the tabular format
    Refine the data filtering duplicate entries
    Convert it into the JSON file
    avail the JSON file in the localhost.
end procedure
procedure (B): PREDICT ON THE FETCHED JSON FROM THE LOCALHOST:
    Fetch the trained model
    Transfer the trained model to the IoT node
    Fetch the processed prediction data from the localhost in the network
    Apply ML model on the data
    Predict the multi-level attacks based on the data
    Produces the result with the devices infected by the malware or attacks
    Overview information to suggest an appropriate action to mitigate the attack
end procedure
```

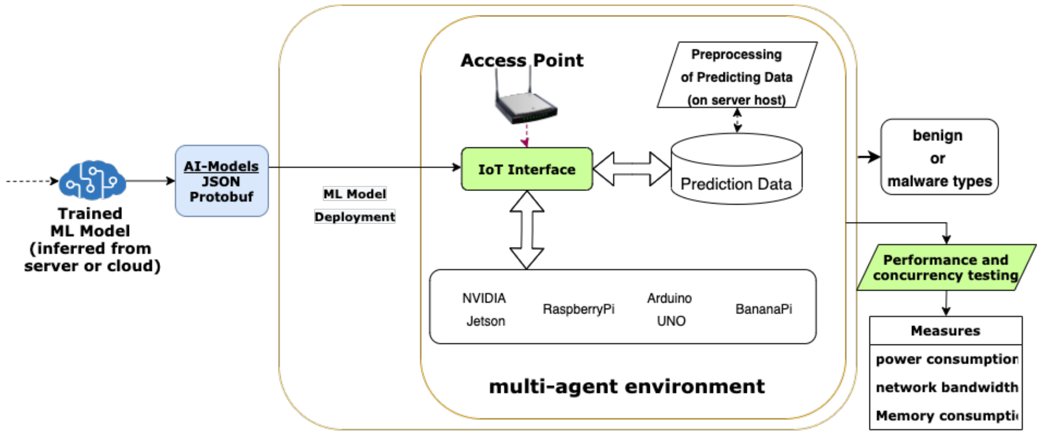


Figure 4. AI model transfer approach performing high inference on server and prediction on IoT devices.

4. Experimental Setup and Results

Preprocessing of the dataset is conducted to represent it so that the ML can be appropriately utilized for attack detection in IoT devices. In this section, a brief introduction of the used datasets and results of the ML models with the performance and concurrency testing on the hardware devices are presented with graphical representations.

4.1. Overview of the Datasets

In this study, the *Aposemat IoT-23* dataset (https://www.stratosphereips.org/datasets-iot23; accessed on 15 July 2022) and *Edge-IIoTset* [45] IoT network traffic are used for the training and testing of ML models. The *IoT-23* dataset is a semi-structured log of information, labeled as malicious or benign IoT network traffic packets. It was created by Avast AIC laboratory collected from different IoT devices. The dataset contains a total of 325,307,990 captures, of which 294,449,255 are malicious samples. Several studies [46,47] have also used this dataset for network traffic analysis, malware, and attack detection applications. The network traffic information is extracted using *Wireshark*, and *tcpdump* in *.pcap* files which are semi-structured textual files. The second dataset, *Edge-IIoTset* contains 1,363,998 normal and 545,673 attack samples. The feature extraction and selection processes are carried out on the converted structured data

4.2. AI-Specific Performance Measures

To check the practical implication of the ML approach for malware detection and attack prediction, we applied several ML and DNN models on the same set of processed data. Resulting in the performance of the DNN model looks promising for IoT security. The given Table 2 presents performance measures, accuracy, precision, recall, and f1-score of the ML models on both datasets for training and testing. Since the f1-score of the deep neural network (DNN) model is observed as being the highest, therefore, we applied a trained DNN model for the prediction of malware and attacks in our project environments.

Table 2. Performance measures of different ML and DNN models.

Model	**Dataset**	**Accuracy**		**Precision**		**Recall**		**F1-Score**	
		Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
DNN	IoT-23	0.93	0.93	**0.95**	**0.97**	**0.92**	**0.92**	**0.93**	**0.94**
	EdgeIIoTset	0.94	0.94	0.97	**0.98**	0.87	**0.89**	0.92	**0.93**
SVM	IoT-23	**0.95**	**0.95**	0.55	0.54	0.42	0.43	0.48	0.48
	EdgeIIoTset	0.96	**0.96**	0.86	0.88	0.84	0.84	0.85	0.86
RF	IoT-23	**0.95**	**0.95**	0.74	0.59	0.45	0.44	0.56	0.5
	EdgeIIoTset	0.98	0.95	0.94	0.87	0.92	0.84	0.93	0.85
DT	IoT-23	**0.95**	**0.95**	0.72	0.56	0.47	0.45	0.57	0.5
	EdgeIIoTset	**0.99**	**0.96**	**0.99**	0.86	**0.99**	0.85	**0.99**	0.85
GB	IoT-23	**0.95**	**0.95**	0.55	0.54	0.42	0.43	0.48	0.48
	EdgeIIoTset	0.96	**0.96**	0.86	0.88	0.84	0.84	0.85	0.86
NB	IoT-23	0.82	0.82	0.38	0.29	0.5	0.49	0.43	0.36
	EdgeIIoTse	0.92	0.92	0.71	0.71	0.7	0.7	0.7	0.7

4.3. Concurrency and Hardware Performance Testing

Hardware performance testing is a form of resource utilization testing that focuses on how a system running the model performs under a particular load and environment. Different hardware performance measures contribute to benchmarking and standardizing the deployment model to IoT systems.

In this study, we used PeakTech TrueRMS BENCH-Type multimeter 4090 (https://www.peaktech.de/uk/PeakTech-P-4090-Graphical-bench-multimeter-22.000-counts-with-USB/P-4090; accessed on 10 September 2022) for measuring precise power consumption. The power is supplied from the PeakTech 6227 power supplier (https://peaktech-rce.com/en/laboratory-power-supplies/644-peaktech-6227-laboratory-switching-power-supply-dc-0-60v-0-6a-max-150w-multifunction-digital-lcd-display.html; accessed on 10 September 2022) to the IoT devices in which the AI model was deployed. Figure 5 shows the physical view of our hardware setup for the measurement of power and current consumption. Additionally, Figures 6–8 show the circuit diagrams of both publishers and subscribers on three different use-cases: (i) *waste bin*, (ii) *Luminance*, and (iii) *temperature* sensing respectively.

For other measures, such as CPU consumption, memory usage, network bandwidth, etc., Netdata cloud tool (https://app.netdata.cloud/; accessed on 8 May 2022) is used. Netdata is an open-source and real-time infrastructure monitoring and troubleshooting tool. It collects several metrics from systems, hardware, and applications and supports visualizing them.

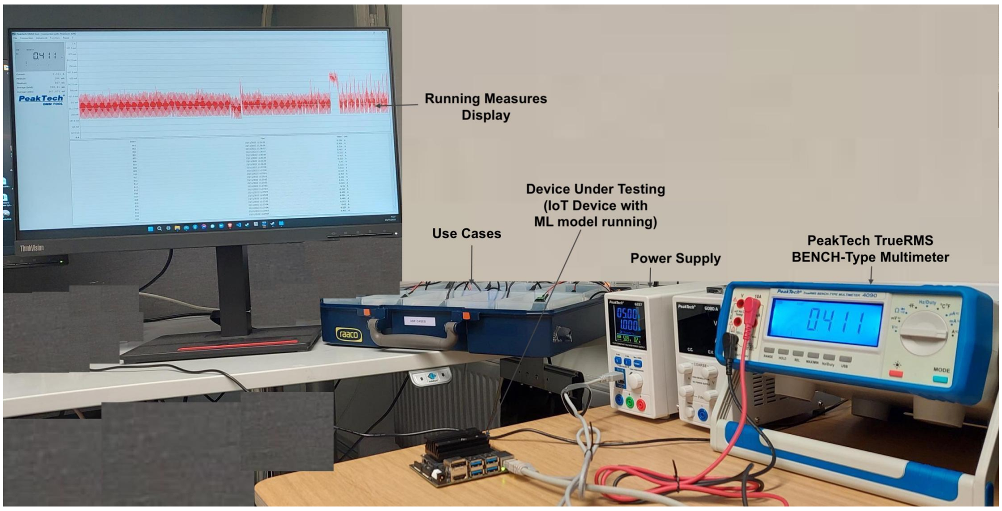
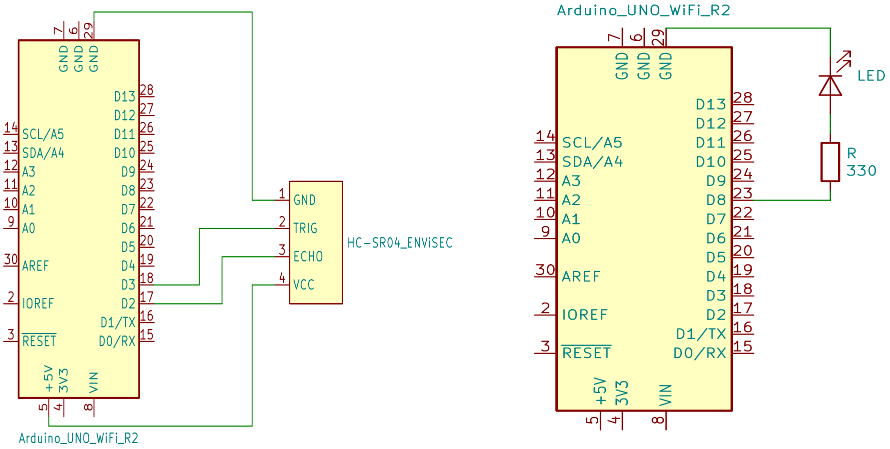
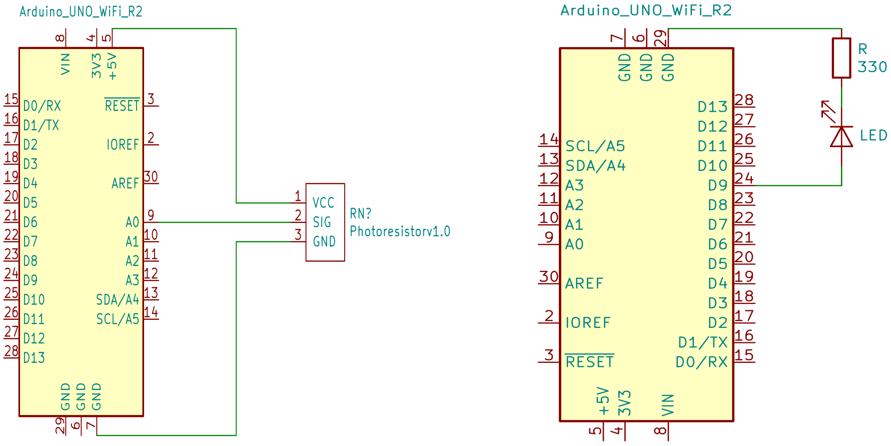

Running Measures Display
Device Under Testing (IoT Device with ML model running)
Use Cases
Power Supply
PeakTech TrueRMS BENCH-Type Multimeter

Figure 5. Hardware setup for the measurement of power and current consumption.



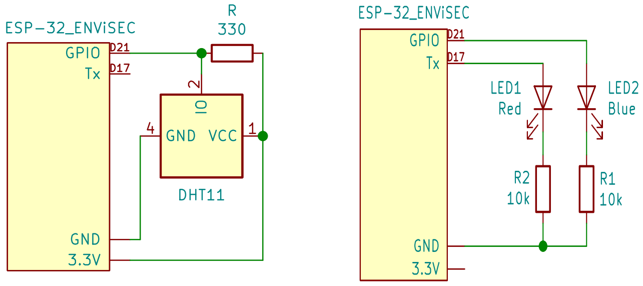
(**a**) (**b**)

Figure 6. Circuit diagram of the *waste bin* use-case, (**a**) Publisher, and (**b**) Subscriber.



(**a**) (**b**)

Figure 7. Circuit diagram of the *Luminance sensing* use-case, (**a**) Publisher, and (**b**) Subscriber.

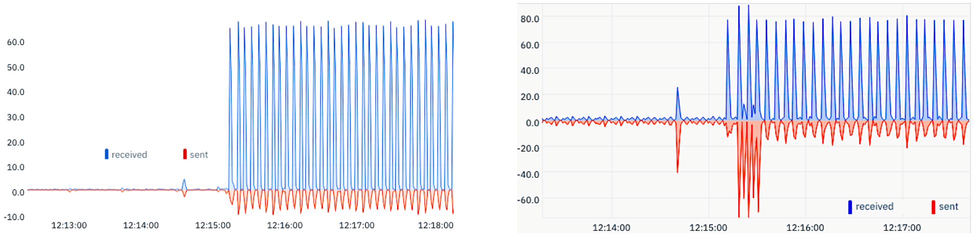


(**a**) (**b**)

Figure 8. Circuit diagram of the *temperature* sensing use-case, (**a**) Publisher, and (**b**) Subscriber.

4.3.1. Network/Concurrency Measures

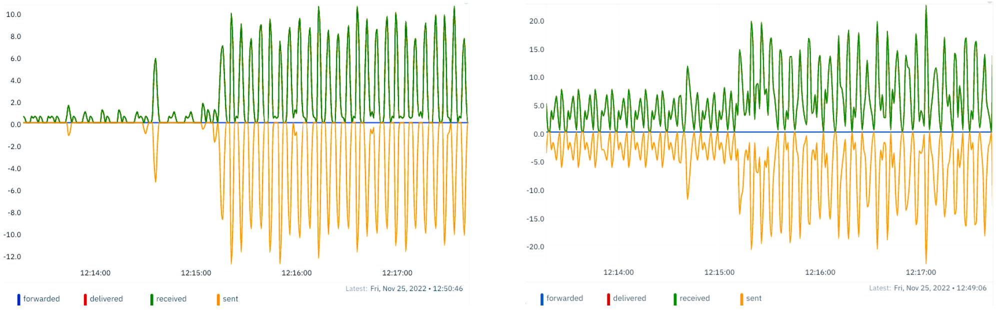
(a) *Nework Bandwidth:* This measure refers to the aggregated bandwidth of all physical network interfaces of the IoT devices. The measure does not include lo, VPNs (virtual private networks), network bridges, IFB (intermediate functional block) devices, bond interfaces, etc. As shown in Figure 9a as a model running on the Jetson device at time 12:15, the inbound traffic fluctuates between 0 and 60 kb/s and outbound traffic lies between 0 and −10.0 kb/s. Similarly, on Raspberry Pi (Figure 9b), before running the ML model, the sent and received bandwidth fluctuates from 0 to 4.7 kb/s and 0 to −4.0 kb/s. With the ML model running, the receiving bandwidth lies between 0 and 80 kb/s, and the sending bandwidth is between 0 and −80 kb/s.



(**a**) (**b**)

Figure 9. Network bandwidth consumption by the devices; (y → bandwidth in kb/s and x → time), (**a**) Jetson, (**b**) Rasberry Pi.

(b) *Packets Statistics:* Another important measure related to network architecture is packet statistics. This measure for the host shows the received packets by the internet protocol (IP) layer and sent packets via the IP layer. The measure does not include the forwarded packet count. Figure 10 presents the variation in statistics of internet protocol version 4 (IPv4) network packets by running the ML model initiated at time 12:15.



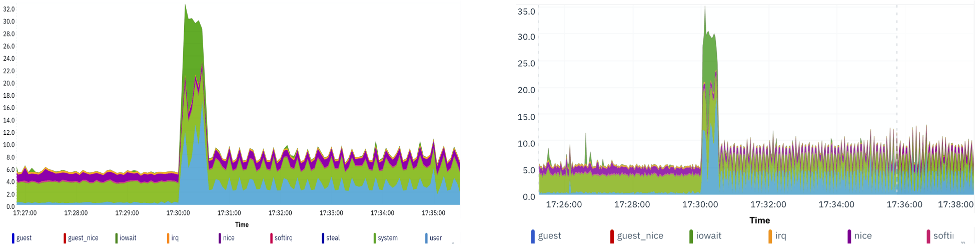
(**a**) (**b**)

Figure 10. The variation in IPV4 Network Packets by ML model (**a**) Jetson, (**b**) Rasberry Pi (y → packets & x → time).

4.3.2. System Load

The system load is a measure of the amount of computational work that an IoT device performs in a certain condition over some time. In this study, we calculated several system load measures; CPU consumption, system processes, memory consumption, and disk usage to check the in-production implementation of the proposed model. The variations in these sub-measures in the idle mode of the device (baseline with normal functioning with no ML model deployed) and ML running modes (baseline + ML) are demonstrated with the help of pictorial representations.

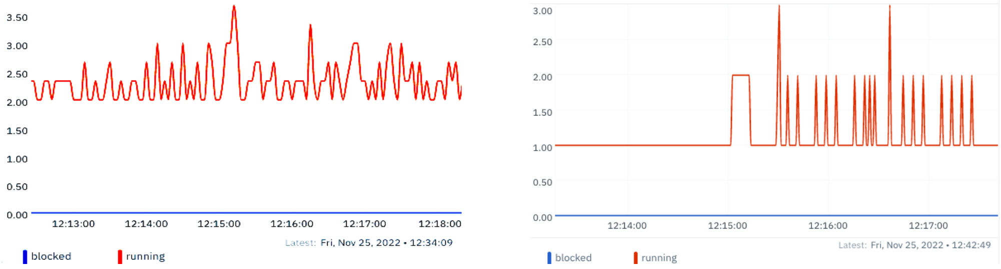
(a) *CPU Consumption (%):* This measure corresponds with the total CPU utilization (100%) of all cores of the device. Figure 11 presents the CPU utilization of (a) NVIDIA Jetson, and (b) Raspberry Pi devices where the ML model was initiated to run at time 17:30. The running ML model required up to ≈32% CPU of Jetson and ≈35% of Raspberry Pi for initiating the ML model for a couple of seconds and then it behaves normally consuming only up to ≈8% on average which is just ≈2% increased than idle (baseline) mode on both devices.



(**a**) (**b**)

Figure 11. CPU utilization on IoT devices (**a**) Jetson, and (**b**) Raspberry Pi (y-axis refers CPU% used).

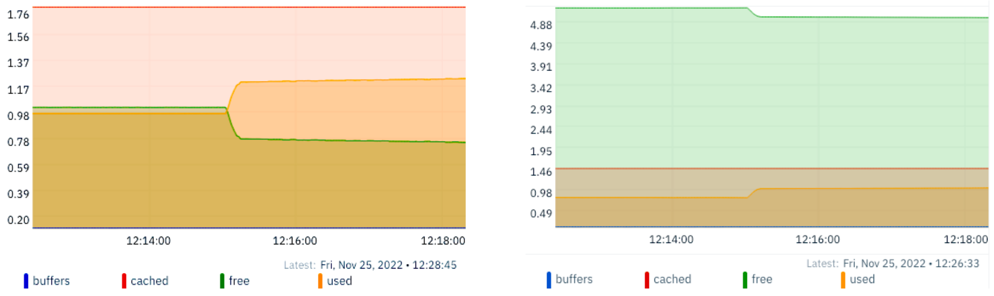
(b) *System Processes:* This measure indicates the average of total system processes consumed by the device. It consists of both runnable (running or ready to run) and blocked (or waited for I/O to complete) system processes. Figure 12 shows the status of system processes before and after running the model on both (a) NVIDIA Jetson and (b) Raspberry Pi devices. There is a little increase at time 12:15 when the ML model was started, and then it behaves normally as before in Figure 12a Jetson, where the Raspberry Pi device (Figreu 12b) shows a small fluctuation after deployment of the ML model.



(**a**) (**b**)

Figure 12. Consumption of the system processes on devices (**a**) Jetson, and (**b**) Raspberry Pi, at time 12:15 (y → average system processes and x → time).

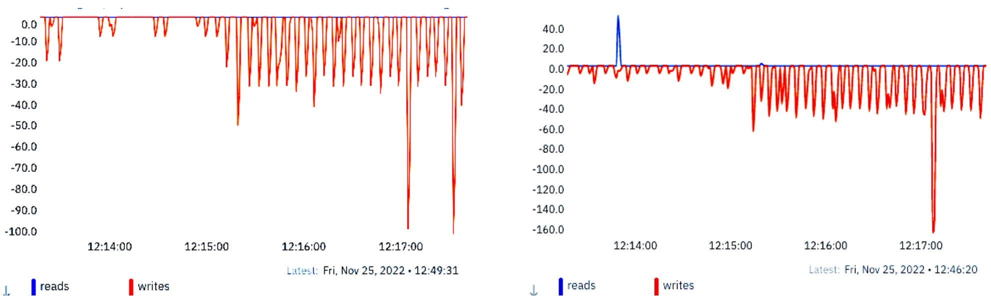
(c) *Memory Consumption (RAM Usage):* Similar to CPU usage, running an ML model on these devices consumes a small amount of physical memory—random access memory (RAM)—as well. Figure 13 shows the amount of memory usage (RAM) on the IoT devices, (a) Jetson and (b) Raspberry Pi. The memory allocation increases by 0.42 GB in the Jetson, where the buffer memory remains constant at 0.06 GB even after running the ML model. Likewise, the memory allocated in Raspberry Pi is slightly increased just by 0.2 GB. Therefore, our proposed ML model with much less physical memory consumption suggests that it can be deployed on these tiny IoT devices for in-production real-world applications for the detection of malware and cyberattacks.



(**a**) (**b**)

Figure 13. The variation in the RAM utilization on devices (**a**) Jetson, (**b**) Raspberry Pi (y → memory in GB and x → time).

(d) *Disk Usage (MicroSSD Card):* In the hardware setup for the experimentation, both device-under-testing (DUT) gateways are supported by the MicroSD card for the operating system and secondary storage. The disk usage measure indicates the I/O overhead to the storage by the device. Figure 14 shows the disk bandwidth consumed by both experimented devices, NVIDIA Jetson and Raspberry Pi. Once the program is loaded into the memory, it seems it rarely reads the data from storage; however it writes the predicted output to the storage. So, the software engineer can look at it for reference and take further action to mitigate the detected malware and attacks.



(**a**) (**b**)

Figure 14. Disk bandwidth consumed by the IoT devices (**a**) Jetson, (**b**) Raspberry Pi (y → disk bandwidth in kb/s and x → time).

4.3.3. Electricity

The energy consumption prediction suggests whether the AI model to the IoT devices is feasible to adapt to real-world environments. The objective of the prediction is to make the model optimized in terms of low energy levels so that battery power can last longer. In this study, we computed electricity measures in both idle and AI model running cases. Here, the AI model running mode indicates the prediction of malware and attacks by analyzing the proactive network traffic using the trained model that was produced from the training phase. The current drawn by the devices is measured both in the idle and ML model running mode, by connecting a PeakTech 4090 digital ammeter in series with the devices. Based on these measurements, the power consumption is calculated using $P = U \times I$. The current measurement, calculated power consumption, and increase in percentage are presented in Table 3. Figures 15 and 16 show the graphical representations of the variation in current usage by the NVIDIA Jetson and Raspberry Pi devices, respectively. Similarly, Figures 17 and 18 depict the variation in the calculated power consumption in NVIDIA Jetson and Raspberry Pi, respectively. In an attempt to calculate the theoretical power consumption of the ML model, we consider the devices running the model powered by an average Li-ion battery with about 8 [Wh] (≈4.0 V · 2.0 Ah, type 18,650 for the sake of example).

Using the Equation (1), it yields a theoretical 8/(1.353–1.167) [Wh/W] ≈ 43 [h] model running time on the NVIDIA Jetson, and 8/(2.346–2.109) ≈ 33.8 [h] model running time on Raspberry Pi. A more practical approximation gives us 8/1.353 [Wh/W] ≈ 5.9 [h] for the Jetson in model running mode, compared to 8/1.168 [Wh/W] ≈ 6.8 [h] in idle mode. For the Raspberry Pi we get 8/2.346 [Wh/W] ≈ 3.4 [h] in model running mode, as opposed to 8/2.109 [Wh/W] ≈ 3.8 [h] in idle mode.

$$\text{Total possible time to run model [h]} = \frac{\text{Battery capacity [Wh]}}{\text{ML running mode} - \text{idle mode [W]}}. \tag{1}$$

Table 3. Power performance of AI model.

Device	**Idle Mode**		**Model Running Mode**		**Percentage Increase**
	Current [A]	***Power [W]***	***Current [A]***	***Power [W]***	
Nvidia Jetson	0.229	1.167	0.265	1.353	∼15.9%
Raspberry Pi	0.421	2.109	0.469	2.346	∼11.2%

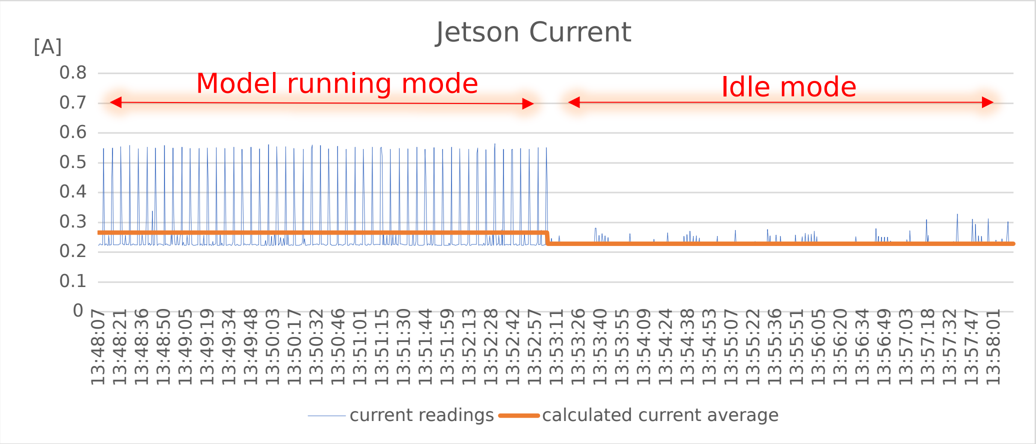


Figure 15. Variation in current consumption for ML model in Jetson; (y → current [A], x → time).

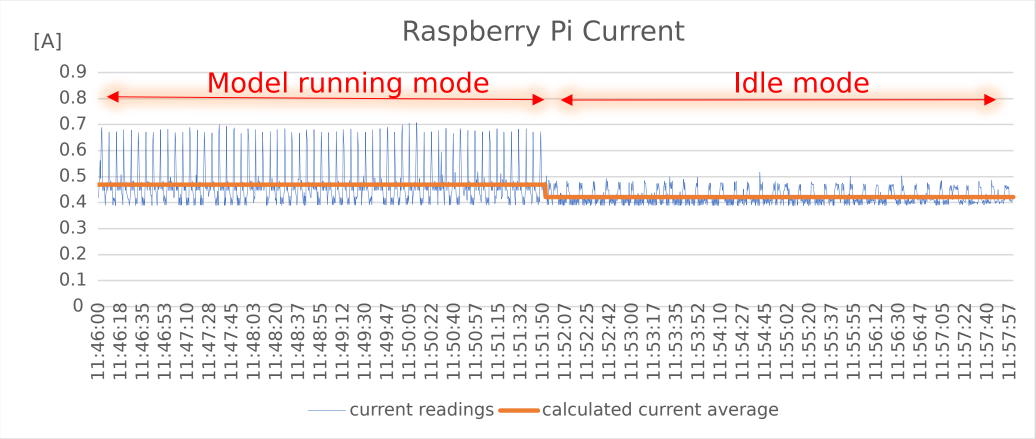


Figure 16. Variation in current consumption for ML model in Raspberry Pi; (y → current [A], x → time).

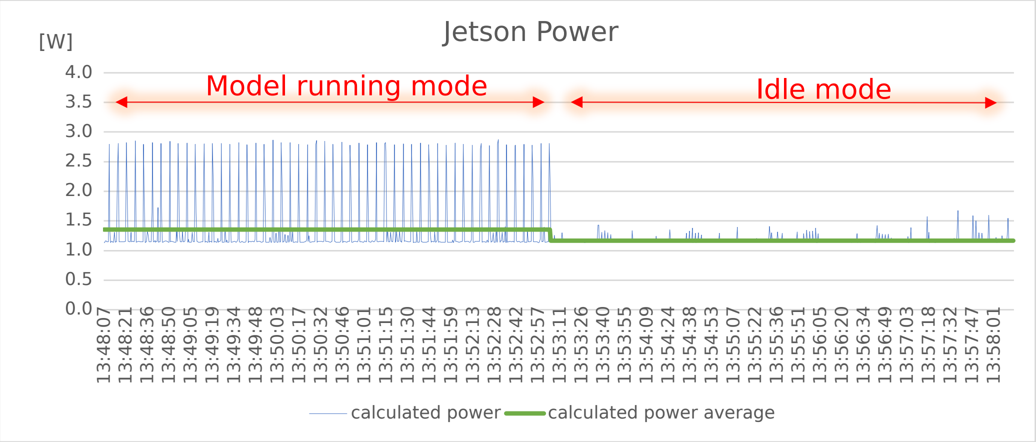


Figure 17. Variation in current consumption for ML model in Jetson; (y → power [W], x → time).

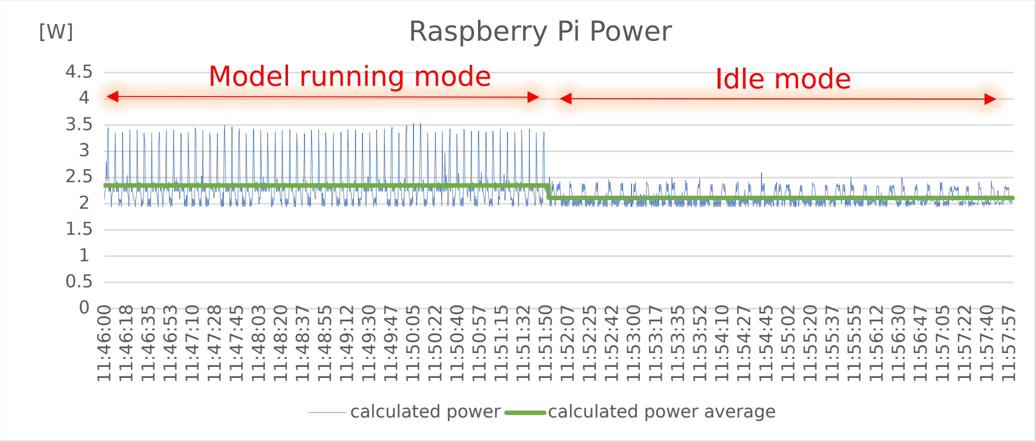


Figure 18. Variation in power consumption for ML model in Raspberry Pi; (y → power [W], x → time).

5. Discussion and Future Directions

This section briefly describes the discussion on challenges and considerations inspecting AI-enabled methods for IoT cybersecurity. Additionally, our ongoing efforts and future directions to further optimize the proposed approach are also covered in this section.

5.1. Discussion

Due to the lack of ground-truth data on cyberattacks, the trustworthiness of the AI model relies on data labeling. The public datasets available on the internet are specific to the data extraction tool and IoT environments deployed to extract the network traffic data. However, our approach ignored these infrastructure-specific identifiers-based columns, such as IPs, ports, timestamps, IDs, etc. We believe that the other remaining features than identifier-based features of packets depict the general scenarios and attack patterns. The publicly available datasets *IoT-23* and *Edge-IIoTset* [45] are based on artificially generating malware attack samples. However, the real-world scenario would be different than the situation of artificially injecting the attacks from a few machines with only a few IP addresses. The ground truth data from the actual malware attacks reflect the actual real-world scenarios, but that may suffer from data imbalance because the normal sample would be very high, and the attack samples can be much less.

The distribution of the number of samples is not balanced in some categories and it is too small to apply supervised machine learning. For example, in *IoT-23* dataset, *C-Mirai* has only two samples, *Okiru-Attack* has three samples. Therefore, overfitting may persist even with new deep-learning models. A type of ML method where the training dataset contains limited information, named few-shot learning (FSL) [48] can be a possible direction to investigate for better results with imbalance data [15]. Our open-source repository of the project is hosted in GitHub (https://github.com/SmartSecLab/ENViSEC) which offers further contributors to extend the research work.

The IoT system designers and practitioners are taking care of security issues in advance designing new smart systems or architectures. In the technological market, anti-virus and anti-malware tools are being widely used by the users [49]. However, there are many cyber-attacks and ransomware, such as WannaCry [50] and Morris Worm [49], which were of high severity impact on multiple infrastructures at a time. Additionally, with the advancement of quantum computers, cybersecurity can become the most critical problem for the internet in the near future [51]. With ML in cybersecurity, the systems can analyze cyberattacks and malware patterns and respond to their changing behaviors [52]. The adaptation of ML in IoT applications enables cybersecurity to be more proactive in defending systems and preventing threats as well as responding to active attacks in real-time with the appropriate action.

5.2. Future Directions

We aim to further improve the proposed method and make the model more efficient in terms of performance and energy consumption. In the following, we list the ongoing efforts to carry out the research domain.

- Implementation using C programming language with a lightweight model in multi-agent systems which can utilize minimal resources to run the prediction;
- Use more advanced types of deep learning techniques and see whether they obtain better performance;
- Consider penetration testing in the constructed Smart Environment;
- In addition to network traffic as input data, we are interested in using the other data obtained from sensors/actuators, source code, reports, texts, logs, and other relevant information related to IoT devices to enable better identification of device-specific (sensor-specific) security-related issues, i.e., anomalies and errors.

6. Conclusions and Future Work

In summary, our AI-enabled detection method discovers multi-level attacks and malware in Smart Environments. The novel method proactively monitors the streamed network traffic data to detect malware and attacks. In general, the deep neural network (DNN) is the best choice with high-performance scores for malware detection and classification in the context of both *IoT-23* and *Edge-IIoTset* datasets considering the complete samples. The presented precise measurement of the power consumption and concurrency testing support hardware engineers in efficiently deploying AI-model in their Smart Environments. With good accuracy, precision, and f1-score, and only a small variation in network bandwidth (30 kb/s on average), CPU utilization (2% increase), and current and power consumption while deploying the AI model to the IoT devices suggests that the new method is efficient for in-production deployment. Moreover, the result of the study suggests that the model detects malware and attacks accurately and efficiently in IoT devices. The use of the tool assists in pinpointing the infected IoT devices, and minimizing malware assessment costs and intensive manpower automating the detection process.

Author Contributions: Conceptualization, A.S. and T.-M.G.; Methodology, G.B. and A.L.; Software, G.B. and A.L.; Writing—original draft, G.B. and A.L.; Writing—review & editing, A.S. and T.-M.G.; Supervision, A.S.; Project administration, A.S. and T.-M.G.; Funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Union's Horizon 2020 research and innovation program within the framework of the NGI POINTER Project under grant agreement# 871528.

Data Availability Statement: Our project repository is hosted in GitHub (https://github.com/SmartSecLab/ENViSEC) which provides the source code and instructions to replicate the experiment.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Belli, L.; Cilfone, A.; Davoli, L.; Ferrari, G.; Adorni, P.; Di Nocera, F.; Dall'Olio, A.; Pellegrini, C.; Mordacci, M.; Bertolotti, E. IoT-Enabled Smart Sustainable Cities: Challenges and Approaches. *Smart Cities* **2020**, *3*, 1039–1071. [CrossRef]
2. Cyrus, C. BotenaGo Malware Targets Millions of IoT Devices. Available online: https://www.iotworldtoday.com/2021/11/16/botenago-malware-targets-millions-of-iot-devices/ (accessed on 23 March 2022).
3. Shkolnik, M. 3 Steps: Cyber Breach Recovery Plan—Based on Verkada Breach. Available online: https://firedome.io/blog/cyber-breach-recovery-plan-based-on-verkada-breach/ (accessed on 23 March 2022).
4. Conner, B. 2022 SonicWall Cyber Threat Report. Technical Report. Available online: https://www.sonicwall.com/resources/white-papers/2022-sonicwall-cyber-threat-report/ (accessed on 23 March 2022).
5. Shalaginov, A.; Azad, M.A. Securing Resource-Constrained IoT Nodes: Towards Intelligent Microcontroller-Based Attack Detection in Distributed Smart Applications. *Future Internet* **2021**, *13*, 272. [CrossRef]
6. Bout, E.; Loscri, V.; Gallais, A. How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2021**, *24*, 248–279. [CrossRef]
7. Xenofontos, C.; Zografopoulos, I.; Konstantinou, C.; Jolfaei, A.; Khan, M.K.; Choo, K.K.R. Consumer, Commercial, and Industrial IoT (In)Security: Attack Taxonomy and Case Studies. *IEEE Internet Things J.* **2021**, *9*, 199–221. [CrossRef]
8. Rawat, D.B.; Doku, R.; Garuba, M. Cybersecurity in Big Data Era: From Securing Big Data to Data-Driven Security. *IEEE Trans. Serv. Comput.* **2019**, *14*, 2055–2072. [CrossRef]
9. Shalaginov, A.; Grønli, T.M. Securing Smart Future: Cyber Threats and Intelligent Means to Respond. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 2560–2564. [CrossRef]
10. Augusto, J.C. Past, Present and Future of Ambient Intelligence and Smart Environments. In *Proceedings of the Agents and Artificial Intelligence*; Filipe, J., Fred, A., Sharp, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 3–15. [CrossRef]
11. Augusto, J.C.; Nakashima, H.; Aghajan, H. Ambient Intelligence and Smart Environments: A State of the Art. In *Handbook of Ambient Intelligence and Smart Environments*; Nakashima, H., Aghajan, H., Augusto, J.C., Eds.; Springer: New York, NY, USA, 2010; pp. 3–31. [CrossRef]
12. Tait, K.A.; Khan, J.S.; Alqahtani, F.; Shah, A.A.; Ali Khan, F.; Rehman, M.U.; Boulila, W.; Ahmad, J. Intrusion Detection using Machine Learning Techniques: An Experimental Comparison. In Proceedings of the 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Taiz, Yemen, 4–5 July 2021; pp. 1–10. [CrossRef]
13. Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X. Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset). In *Proceedings of the 12th International Networking Conference*; Ghita, B., Shiaeles, S., Eds.; Lecture Notes in Networks and Systems; Springer: Berlin/Heidelberg, Germany, 2021; pp. 73–84. [CrossRef]
14. Khan, M.A.; Khan, M.A.; Jan, S.U.; Ahmad, J.; Jamal, S.S.; Shah, A.A.; Pitropakis, N.; Buchanan, W.J. A deep learning-based intrusion detection system for MQTT enabled IoT. *Sensors* **2021**, *21*, 7016. [CrossRef]
15. Lin, K.; Xu, X.; Xiao, F. MFFusion: A multi-level features fusion model for malicious traffic detection based on deep learning. *Comput. Netw.* **2022**, *202*, 108658. [CrossRef]
16. Ullah, I.; Mahmoud, Q.H. Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access* **2021**, *9*, 103906–103926. [CrossRef]
17. Popoola, S.I.; Ande, R.; Adebisi, B.; Gui, G.; Hammoudeh, M.; Jogunola, O. Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT-Edge Devices. *IEEE Internet Things J.* **2021**, *9*, 3930–3944. [CrossRef]
18. Dutta, V.; Choraś, M.; Pawlicki, M.; Kozik, R. A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection. *Sensors* **2020**, *20*, 4583. [CrossRef]
19. Popoola, S.I.; Adebisi, B.; Hammoudeh, M.; Gui, G.; Gacanin, H. Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks. *IEEE Internet Things J.* **2020**, *8*, 4944–4956. [CrossRef]
20. Abdalgawad, N.; Sajun, A.; Kaddoura, Y.; Zualkernan, I.A.; Aloul, F. Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset. *IEEE Access* **2021**, *10*, 6430–6441. [CrossRef]
21. Hu, X.; Gu, C.; Chen, Y.; Wei, F. CBD: A deep-learning-based scheme for encrypted traffic classification with a general pre-training method. *Sensors* **2021**, *21*, 8231. [CrossRef]
22. Sikos, L.F. Handling Uncertainty and Vagueness in Network Knowledge Representation for Cyberthreat Intelligence. In Proceedings of the 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6. [CrossRef]
23. Rahman, M.R.; Mahdavi-Hezaveh, R.; Williams, L. A Literature Review on Mining Cyberthreat Intelligence from Unstructured Texts. In Proceedings of the 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy, 17–20 November 2020; pp. 516–525. [CrossRef]
24. Zhang, S.; Zhang, M.; Li, H.; Bai, G. Threat Analysis of IoT Security Knowledge Graph Based on Confidence. In *Proceedings of the Emerging Technologies for Education*; Jia, W., Tang, Y., Lee, R.S.T., Herzog, M., Zhang, H., Hao, T., Wang, T., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2021; pp. 254–264. [CrossRef]
25. Mozzaquatro, B.A.; Agostinho, C.; Goncalves, D.; Martins, J.; Jardim-Goncalves, R. An Ontology-Based Cybersecurity Framework for the Internet of Things. *Sensors* **2018**, *18*, 3053. [CrossRef]

26. Choi, C.; Choi, J. Ontology-Based Security Context Reasoning for Power IoT-Cloud Security Service. *IEEE Access* **2019**, *7*, 110510–110517. [CrossRef]
27. Strecker, S.; Dave, R.; Siddiqui, N.; Seliya, N. A Modern Analysis of Aging Machine Learning Based IoT Cybersecurity Methods. *arXiv* **2021**, arXiv:2110.07832.
28. Andrade, R.O.; Yoo, S.G.; Tello-Oquendo, L.; Ortiz-Garcés, I. A Comprehensive Study of the IoT Cybersecurity in Smart Cities. *IEEE Access* **2020**, *8*, 228922–228941. [CrossRef]
29. Osborne, C. Remote Code Execution Flaw Allowed Hijack of Motorola Halo+ Baby Monitors. Available online: https://portswigger.net/daily-swig/remote-code-execution-flaw-allowed-hijack-of-motorola-halo-baby-monitors (accessed on 20 August 2022).
30. Lu, Y.; Xu, L.D. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet Things J.* **2018**, *6*, 2103–2115. [CrossRef]
31. Ismail, L.; Buyya, R. Artificial Intelligence Applications and Self-Learning 6G Networks for Smart Cities Digital Ecosystems: Taxonomy, Challenges, and Future Directions. *Sensors* **2022**, *22*, 5750. [CrossRef]
32. Bendiab, G.; Shiaeles, S.; Alruban, A.; Kolokotronis, N. IoT Malware Network Traffic Classification using Visual Representation and Deep Learning. In Proceedings of the 2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, 29 June–3 July 2020; pp. 444–449. [CrossRef]
33. Ward, J. Top 5 Raspberry Pi Network Security Tips for Beginners. Available online: https://www.raspberrypistarterkits.com/guide/top-raspberry-pi-network-security-tips-beginners/ (accessed on 29 November 2022).
34. Sforzin, A.; Mármol, F.G.; Conti, M.; Bohli, J.M. RPiDS: Raspberry Pi IDS—A Fruitful Intrusion Detection System for IoT. In Proceedings of the 2016 International 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), Toulouse, France, 18–21 July 2016; pp. 440–448. [CrossRef]
35. Arduino. Arduino Reference. Available online: https://www.arduino.cc/reference/en/ (accessed on 29 November 2022).
36. Arduino Cryptography Library: Arduino Cryptography Library. Available online: https://rweather.github.io/arduinolibs/crypto.html (accessed on 17 August 2022).
37. Shalaginov, A.; Semeniuta, O.; Alazab, M. MEML: Resource-aware MQTT-based Machine Learning for Network Attacks Detection on IoT Edge Devices. In Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion, ACM, UCC '19 Companion, Auckland, New Zealand, 2–5 December 2019; pp. 123–128. [CrossRef]
38. Bhandari, G.P.; Lyth, A.; Shalaginov, A.; Grønli, T.M. Artificial Intelligence Enabled Middleware for Distributed Cyberattacks Detection in IoT-based Smart Environments. In Proceedings of the IEEE International Conference on Big Data 2022 (Big Data), Osaka, Japan, 17–20 December 2022.
39. Mellis, D. Protecting the Three States of Data. Available online: https://blog.arduino.cc/2016/04/27/machine-learning-for-the-maker-community/ (accessed on 29 November 2022).
40. Mellis, D.A. ESP (Example-Based Sensor Predictions). Available online: https://github.com/damellis/ESP (accessed on 29 November 2022).
41. Śmigielski, M. Machine Learning Library for Arduino. Available online: https://github.com/smigielski/q-behave (accessed on 29 November 2022).
42. Heymsfeld, R. A Neural Network for Arduino. Available online: http://robotics.hobbizine.com/arduinoann.html (accessed on 29 November 2022).
43. Moretti, C.B. Neurona—Artificial Neural Networks for Arduino. Available online: https://github.com/moretticb/Neurona (accessed on 30 November 2022).
44. Biswas, S. Advantages of Deep Learning, Plus Use Cases and Examples. Available online: https://www.width.ai/post/advantages-of-deep-learning (accessed on 22 December 2022).
45. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access* **2022**, *10*, 40281–40306. [CrossRef]
46. Stoian, N.A. Machine Learning for Anomaly Detection in IoT Networks: Malware Analysis on the IoT-23 Data Set. Bachelor Thesis, University of Twente, Enschede, The Netherlands, 2020.
47. Liang, Y.; Vankayalapati, N. Machine Learning and Deep Learning Methods for Better Anomaly Detection in IoT-23 Dataset Cybersecurity. Preprint. Available online: https://github.com/yliang725/Anomaly-Detection-IoT23 (accessed on 22 December 2022).
48. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv.* **2020**, *53*, 63. [CrossRef]
49. Jajoo, A. A Study on the Morris Worm. Available online: http://xxx.lanl.gov/abs/2112.07647[cs] (accessed on 19 December 2022).
50. WannaCry Ransomware Attack. Available online: https://en.wikipedia.org/w/index.php?title=WannaCry_ransomware_attack&oldid=1128454751 (accessed on 21 December 2022).

51. Hossain Faruk, M.J.; Tahora, S.; Tasnim, M.; Shahriar, H.; Sakib, N. A Review of Quantum Cybersecurity: Threats, Risks and Opportunities. In Proceedings of the 2022 1st International Conference on AI in Cybersecurity (ICAIC), Victoria, TX, USA, 24–26 May 2022; pp. 1–8. [CrossRef]
52. Zhang, F. The Growing Role of Machine Learning in Cybersecurity. Available online: https://www.securityroundtable.org/the-growing-role-of-machine-learning-in-cybersecurity/ (accessed on 22 December 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.