Internet of Things 27 (2024) 101308

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

A forensic tool for the identification, acquisition and analysis of sources of evidence in IoT investigations

Sergio Ruiz-Villafranca [a,*], Juan Manuel Castelo Gómez [b], José Roldán-Gómez [c]

[a] *University of Castilla-La Mancha, Avda. de España s/n, Albacete, 02071, Spain*
[b] *Technical University of Madrid, Alan Turing s/n, Madrid, 28031, Spain*
[c] *University of Oviedo, Federico García Lorca 18, Gijón, 33007, Spain*

ARTICLE INFO

Keywords:
Digital forensics
IoT forensics
Evidence identification
Traffic auditing
Evidence collection

ABSTRACT

The emergence of the Internet of Things (IoT) has posed a new challenge for forensic investigators, who find themselves carrying out examinations in a very heterogeneous and novel scenario. Aspects such as the high number of devices, the unlikelihood of having physical access to them, the short lifetime of the data, or the difficulty of acquiring it, demand changes in some of the key processes of forensic investigations. In this regard, the identification, acquisition, and analysis phases call for an IoT-centred approach that can fulfil the requirements of the environment. Due to the interoperability of the IoT, and the way in which the data is handled and exchanged, the network traffic becomes a very useful source of evidence. In view of this, this paper presents an automatic procedure for identifying, analysing, and acquiring IoT network traffic and using it as a basis for forensic examinations by employing an edge node capable of performing real-time traffic monitoring and analysis on the most popular IoT protocols. Furthermore, by pairing it with an Intrusion Detection System (IDS) based on Machine Learning (ML) algorithms, the proposal is capable of following a proactive approach, detecting threats and taking the corresponding measures to assure the correct initiation of a forensic process.

1. Introduction

With more than 15 billion units connected to the Internet, the Internet of Things (IoT) has become the largest environment in the digital world [1], bringing technology to contexts in which there was none, such as smart cities, eHealth, wearables, or the Industrial Internet of Things (IIoT).

With every advancement, new challenges arise, and, in the case of the IoT, it is its security measures. With numerous devices being used in this environment, the need for strong security measures is great. In contrast, recent reports evidence its poor state. Not only did the number of attacks increased in 2023 in comparison with the previous year, but 83.85% of them operated by exploiting the outdated Teletype Network (Telnet) service [2].

Given this situation, the forensic community has been looking for processes that can ensure the effective retrieval and analysis of IoT data. However, when studying their characteristics and requirements, several aspects are present that make their examination a more challenging task than simply adapting conventional forensic solutions to this new environment. Firstly, IoT networks are usually made up of several devices that constantly exchange data and interact with each other. Besides, this aspect having an impact on the scope of an investigation, it also affects how the pieces of evidence are distributed over the network, as well as their lifetime. Secondly, acquiring the sources of evidence is more difficult than in other conventional scenarios due to three elements: having

[*] Corresponding author.
E-mail addresses: sergio.rvillafranca@uclm.es (S. Ruiz-Villafranca), juanmanuel.castelo@upm.es (J.M.C. Gómez), roldangjose@uniovi.es (J. Roldán-Gómez).

https://doi.org/10.1016/j.iot.2024.101308
Received 29 January 2024; Received in revised form 10 June 2024; Accepted 28 July 2024
Available online 6 August 2024

physical access to a device is not always guaranteed; the sources of evidence are presented in different ways due to the dissimilarity between IoT boards; and the lack of IoT-centred tools hinders the chances of successfully performing an acquisition. Finally, IoT devices normally delegate the data storing operation to cloud services, so little evidence can be found in the actual device.

Under these circumstances, forensic examiners find themselves working in a heterogeneous environment in which the data are very volatile. Additionally, the existing techniques do not guarantee the acquisition of the sources of evidence. This increases the importance of responding rapidly when an incident arises so that the smallest amount of data is lost during the time it takes to initiate the forensic investigation. In addition, since non-volatile and volatile memory data is difficult to obtain, the network traffic offers possibly the best source of evidence that may be crucial in an examination.

Unfortunately, similarly to what occurs with the types of IoT devices that can be found in this environment, IoT network traffic is not heterogeneity-free. Several protocols can be found that have been specifically designed to be used by IoT devices, with Zigbee, Z-Wave, 6LoWPAN and S7COMM protocols being some examples [3]. Furthermore, they are also compatible with widely-used ones such as Wi-Fi or Bluetooth, which enables them to interact with conventional devices too. In addition, its acquisition or analysis has to be done in real-time and by a device inside the IoT network, otherwise the data will be lost. This means that the investigator needs to have control over a device in the network that can perform the sniffing task and that has the corresponding tools to perform the operation.

In order to address these issues, this article proposes the use of an edge node as a tool to assist in the forensic process. It does so by acting as an intermediary between IoT endpoints and monitoring the network traffic. Tailored to study the most common protocols in the IoT and IIoT, this node can collect and analyse packet-by-packet in real time. This facilitates the investigator's tasks in several ways. Firstly, it enables constant access to a device inside the IoT network in which they can perform manual forensic tasks if needed. Secondly, it ensures access to a source of evidence in an examination, in this case, the network traffic. Thirdly, it allows the automation of some key phases of the forensic process, namely identification, acquisition, and analysis, without having an impact on the network. And finally, it keeps track of the changes in behaviour in the network, thus being able to rapidly detect the number of devices in it and therefore limit the range of an investigation.

Additionally, in order to reduce the response time to a minimum and offer a proactive approach as well, this Edge node uses an Intrusion Detection System (IDS) based on Machine Learning (ML) algorithms to detect threats and notify the manager of the IoT network so that they can be aware of the existence of a possible incident.

1.1. Research questions formulated

With the goal of this research set, the following questions arise when evaluating its feasibility:

- **(RQ1)** Given the inability of IoT devices to be forensic-friendly, is it possible to develop a solution that can assist in making IoT scenarios forensic-ready to a certain extent?
- **(RQ2)** Since it is crucial to reduce the response time in IoT forensic investigations to a minimum due to the short lifetime of the data, can an approach be proposed that is able to detect threats and take measures to collect and preserve data if an examination is needed?
- **(RQ3)** Taking into account the difficulty of identifying the devices in an IoT forensic investigation, is it possible to track the behaviour of the IoT network and provide this data to the forensic investigator in order to facilitate the identification process?
- **(RQ4)** Considering that network traffic can be a useful source of evidence in IoT forensic examinations, and that it generates a great amount of data, can this data be studied, filtered, and collected following an IoT-centred approach in order for it to be used as a source of evidence?

1.2. Contributions

The contributions of this study are as follows:

- A forensic tool in the form of a node is presented to assist in the proactive and reactive phases of a forensic investigation.
- This forensic node, following the edge computing approach, is integrated in an IoT network and uses its data as a source of evidence to perform the identification, acquisition and analysis phases of the examination process.
- This node, which is composed by multiple services, has been designed with resource-constrained capabilities in mind, so that devices with characteristics similar to those of a Raspberry Pi 4 can run them.
- The proposal provides a dashboard that allows an investigator to interact with the sources of evidence collected. It also displays the information that has been extracted by the node using data filtering and anomaly detection techniques.
- A proof-of-concept is presented in which the proposal is deployed in an IIoT scenario, showing its capability to assist investigators in the forensic process.

The rest of the paper is organized as follows. Section 2 describes the related work regarding IoT forensic investigations. Section 3 presents the forensic edge node that automates the identification, acquisition, and analysis of IoT devices using real-time network traffic analysis. A proof of concept is carried out in Section 4, in which the proposal is tested to determine its functionality and feasibility. Finally, we discuss the experiment's results in Section 5, and present our conclusions in Section 6.

2. Related work

In this section, the proposals from the research community regarding IoT forensics are analysed, detailing the challenges that can be found in this environment and how the related work proposes solutions to deal with them.

Firstly, in order to understand the characteristics and requirements of IoT forensic investigations, we find in Oriwoh et al. [4] a good source of information on which aspects are crucial when examining an IoT device, and in which way this environment differs from the conventional one, highlighting elements such as the number of devices, the quantity, type of data, and its location. Similarly, [5] rises concerns regarding data location and legal jurisdictions, and the difficulty of maintaining the chain of custody in this environment. The authors in Yaqoob et al. [6] introduce a key feature of the IoT, namely its heterogeneity, and explain the multiple scenarios that a forensic investigator may find when examining IoT devices. Finally, [7] addresses the lack of IoT-oriented forensic solutions, and how this issue hinders the process of extracting and analysing the data that IoT devices contain.

When it comes to using external solutions to assist in forensic investigations, the research community has proven that it is a very promising approach, and that it is a matter that should be studied in depth. One of the first significant attempts is that of Perumal et al. [8], which use an external Hadoop server to store and preserve the data that can be collected in IoT examinations. In addition, many relevant concerns regarding IoT investigations are raised in their work, such as warrant obtention, triage examination and the chain of custody.

Focusing on the privacy aspects of IoT forensics and complying with the requirements of ISO/IEC 29100:2011, [9] deploy a piece of software that is in charge of collecting and storing the data generated by IoT devices. Unfortunately, the tool needs to be installed on the device prior to the beginning of the examination, limiting its practicality.

With regards on the Internet of Vehicles (IoV), [10] introduce a very detailed framework that uses a distributed infrastructure for the acquisition and secure storage of IoT data. Additionally, they propose an algorithm for the verification of the pieces of evidence collected, which are tested together with the framework, obtaining valuable results.

Finally, [11] present an investigation framework that uses a fog node that filters and analyses the data generated by an IoT device. In addition, this framework complies with the principles of the Digital Forensic Research Workshop (DFRWS) [12]. Although the concept is quite promising, it is only a theoretical experiment, so there is no information on how the scheme would be implemented from a practical standpoint in real scenarios.

Despite not being centred on using external solutions to assist in the forensic process, there are many frameworks, methodologies, methods, and models that are useful in understanding how to approach the crucial phases of the forensic process. This is especially when addressing the identification, acquisition, and analysis of sources of evidence.

Regarding the identification phase, it is a widely-shared idea that dividing the IoT network into zones based on their physical characteristics is a useful method, with proposals such as [4,13,15–17]. Others authors such as [8,21] use the communications that are made by the IoT devices in order to determine which may be useful sources of evidence.

With respect to acquisition, three approaches can be found. The first one only considers an offline acquisition as the technique to use for collecting the data generated by IoT devices, and this is adopted by Feng et al. [14], Bharadwaj and Singh [19], Kasukurti and Patil [20]. The second approach is opting for an online/live/remote acquisition as the only plausible method, with proposals such as [8,10,11], which have been mentioned above. These all use an external solution to assist in the investigation process. Finally, [18,22] believe that both techniques can be used and should be considered by the investigator.

Lastly, focusing on the analysis, there are not many proposals that detail or provide guidelines on how to approach this phase. However, when this process is addressed, the only technique that is considered is the offline one, with [20,22] providing two examples of this.

2.1. Critical analysis

In this section, we present a critical comparison between the proposals made by the research community and the one presented in this manuscript. The reasoning behind this approach is that performing a practical evaluation is quite difficult due to the characteristics of the forensic field and the related work. Firstly, there are many pieces of research that are more process-modelling-oriented rather than being tool-oriented. This means that the external piece of hardware or software upon which they rely on is either a concept or has not been made publicly available, even there are some that do not present any tool at all. Secondly, there are no forensic testbeds that can be used to make a fair practical comparison between proposals, as investigations are highly shaped by the context in which they are performed, and it varies from one to another. Finally, some of the proposals cover specific IoT scenarios, which means that not all of them can be used under the same circumstances.

As a result, a critical evaluation is the only approach that provides a fair comparison. This way, we are capable of comparing the approach followed by each proposal, how they tackle the most important phases of the forensic process, namely identification, acquisition, and analysis, and their limitations.

A summary of the main characteristics of the pieces of research evaluated can be seen in Table 1.

In conclusion, there are many interesting ideas that can be extracted after studying the related work regarding IoT forensic investigations. The main ones are the following:

- Using an external solution to assist in the forensic process is a promising and potentially effective method, but still needs to be improved in order to become a feasible option.

Table 1
Evaluation of the related work.

Proposal	Approach	Identification	Acquisition	Analysis	Limitations
[4]	Traditional (without relying on external solutions) and follows a network-based zone model	Based on the zone which the device is in	Traditional approach	Traditional	One of the first approaches to model IoT forensics, so it lacks detail and is based on traditional techniques
[8]	Uses a Hadoop server	Based on device communication	Live extraction	Traditional approach	Does not consider live analysis scenarios
[13]	Traditional-based framework that complies with the ISO/IEC 27043	Divided into cloud, network, and device level	Not specified	Not specified	There are crucial aspects of the investigation process that are not detailed
[14]	Traditional	Not specified	Offline	Not addressed	Focused on the Autonomous Automated Vehicle environment
[15]	Traditional	Based on network zones	Traditional	Not addressed	Designed to be combined with a platform to assist in the investigation that was not developed
[10]	Uses a service installed in each device together with a distributed central platform	Not addressed	Using the platform	Not addressed	Focused on the IoV
[16]	Traditional. It is an extension of [13]	Divided into cloud, network, and device level	Not specified	Not specified	Does not solve the deficiencies of its predecessor
[17]	Traditional framework	Network zone-based	Traditional	Not specified	Centred on tool use rather than modelling the forensic process
[11]	Uses a fog node	Using the fog node	Remote	Not addressed	It is only tested from a theoretical viewpoint
[18]	Traditional model that complies with the ISO/IEC 27043	Not specified	Traditional	Not specified	It is not tested or implemented
[19]	Traditional model	Not specified	Offline	Not specified	It is focused on IoT Prototyping Hardware Platform
[20]	Traditional methodology	Physical	Offline	Offline	Focused on the wearable environment and it is not structured
[21]	Traditional framework	By studying radio signals	Not specified	Not specified	Focused on working only with radio frequency signals
[22]	Traditional model that complies with the Industrial Internet Reference Architecture (IIRA)	Through pre-incident manual inventory	Physical and online	Offline	Focused on Industrial Control Systems (ICS)
Proposed research	Uses an edge node	By monitoring real-time network traffic	Online	Offline and online	Only uses network traffic as a source of evidence

- The heterogeneity of the environment is leading researchers to focus on a specific context of the IoT instead of developing generic solutions.
- There are many approaches when it comes to performing the identification phase. Some proposals follow a zone division method, others rely on logical communications, and a few opt for developing a specific solution to carry out this task.
- Contrary to what occurred in conventional forensics, the online/remote/live acquisition has gained popularity in the forensic community due to the difficulty of performing a physical acquisition in the IoT.
- There are few proposals that address the analysis phase, which could be due to the lack of IoT-centred tools that can assist in the process. As a result, investigators must rely on conventional ones, and, therefore, the actions to be followed during the analysis do not vary greatly. However, what does change, and in a drastic way, are the circumstances that surround this phase.

3. Proposed forensic edge node for the identification, acquisition and analysis of sources of evidence in IoT forensic investigations

This section presents the implementation details of the Forensic Edge Node, as outlined in Section 1. The Edge Node is a forensic tool that is designed to perform the various functionalities required in any forencic process when it is deployed into an IoT network. This deployment is described in detail, including the design considerations taken during its development. Secondly, the specifics of

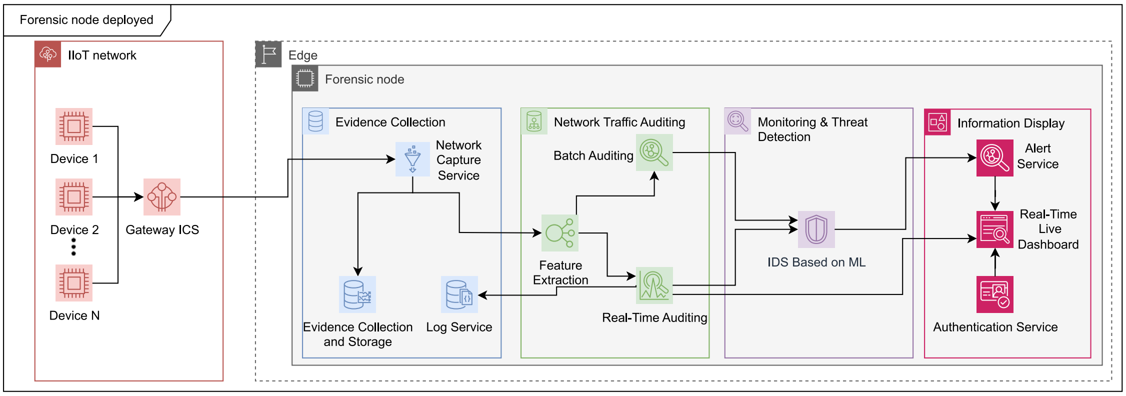


Fig. 1. Forensic edge node's architecture.

the deployment of the architecture related to the forensic edge node are outlined, including the manner in which it can be deployed within an IoT network using an edge layer. Finally, each functionality of the proposed forensic edge node is presented, considering all the stages which it is divided and illustrating their purpose. Furthermore, the manner in which the forensic edge node interacts with the information provided by the network evidence is discussed.

3.1. Forensic edge node fundamentals

During the development of our Forensic Edge Node, we considered several fundamental aspects. These aspects were crucial in shaping the design and functionality of our Edge Node and ensuring its effectiveness in addressing the challenges of IoT forensic examinations. Here are the key considerations that guided our development process:

- **Flexible Deployment.** The Edge Node can work in different IoT contexts, taking into account different network topologies. It can provide benefits such as mobility and resilience to network interruptions, and can operate through edge deployments or as part of the IoT network.
- **Micro-service oriented development.** To ensure overall node application functionality even if one service fails, the Edge Node's functionality is divided into three independent tiers. This construction allows for flexibility in the use of different levels.
- **Multi-instance communication.** By aggregating network traffic information from each node, the Edge Node is able to work with multiple nodes instances simultaneously. The use of containerized technology allows for deployment in multiple containers, providing better management and scalability.
- **Data Integrity and Confidentiality.** The Edge Node maintains data security and supports chain-of-custody through a hardened configuration that restricts access to certain functionalities without authentication and preservation techniques such as hashes, to ensure the integrity of the collected data.

3.2. Description of the proposal

The forensic edge node has been designed to be divided into multiple stages and services, which are detailed in this section. In order to better show its structure and characteristics, in Fig. 1 a graphical representation of the node's architecture is shown.

3.2.1. Data collection

This is the main stage of the forensic edge node, which includes the collection of the network traffic received by the forensic node and, therefore, of the source of evidence. In this stage, the node captures the mirrored traffic, stores it, and sends it to the following stage so that it can be analysed. The result of the acquisition is stored in a log file. The services that comprise this stage are the following:

- **Network capture service.** This service listens on the interfaces of the node to capture the mirrored traffic and process it together with the packet information. This functionality is implemented through the use of the Python framework *Scapy* [23] together with the firewall functionality *iptables* [24], and the module *netfilterqueue* [25], which allows us to implement a filter that determines whether the packet is a useful one to capture, and, if so, filter it using *Scapy* and send it to the sniffer. In addition, the use of the *Scapy* framework facilitates the future real-time analysis of the traffic received. The protocols captured by the node are:

Table 2
Features extracted from each IoT network protocol.

Feature name	Protocol	Description
Source Address	Common to all protocols	Physical address of the source device
Destination Address	Common to all protocols	Physical address of the destination device
Source IP Address	IP based protocols	IP address of the source device
Destination IP Address	IP based protocols	IP address of the destination device
Source Port	IP based protocols	Port used by the source device to send the packet
Destination Port	IP based protocols	Port used by the destination device to receive the packet
Extended Source Address	Zigbee	Extended physical address from the Zigbee protocol communications
Command ID	Zigbee	Indicates the function that the packet makes
Destination PAN ID	6lowpan	Short address that indicates the destination device in the personal area network
Source IPv6 Address	6lowpan	IPv6 address of the source device
Destination IPv6 Address	6lowpan	IPv6 address of the destination device
FCS	6lowpan	Corresponds to a sequence for checking the correct format of the packet
Data	6lowpan	Data encapsulated in the packet
Message Type	MQTT/CoAP/AMQP	Indicates what kind of packet it is and its functionality
Message Content	MQTT	Shows the packet information
Version	MQTT/CoAP	Indicates the version of the protocol used
Message Code	CoAP/AMQP	Identifies the code of the functionality
Message ID	CoAP	Identifies the message in the communication
Message Class	AMQP	Indicates the functionality of the packet
Message Code	AMQP	Shows the identification of the communication
Transition ID	Modbus	Identifies the communication of the packet
Protocol ID	Modbus/S7COMM	Indicates the role in the communication
Function Code	Modbus/S7COMM	Indicates the functionality of the packet
Output Value	Modbus	Shows the values obtained from the master node
SSID	WiFi	Shows the name of the access point network
Address Meaning	WiFi	Indicates the meaning of each interface in the communication
Vendor	WiFi	Show the vendor's name
HCI Packet Type	BLE	Indicates the type of the BLE packet
Command opcode	BLE	Indicates the operational code of the command
BD_ADDR	BLE	Indicates the Bluetooth address of the device that sends the packet

  - *IoT protocols.* The most commonly used IoT protocols can be acquired and analysed by the node, these being: Zigbee, 6lowpan, Message Queuing Telemetry Transport (MQTT), Advanced Message Queuing protocol (AMQP), Bluetooth Low Energy (BLE) and Constrained Application Protocol (CoAP) [26,27].
  - *OT protocols.* The node is able to handle the Transmission Control Protocol (TCP) version of the most popular industrial communication, namely S7COMM [28], which is the proprietary protocol used by Siemens industrial devices, and Modbus/TCP [29].
  - *IT protocols.* In this case, the protocol under study is Wi-Fi, both in its decrypted and encrypted version.

- **Data collection and storage service.** Once the network traffic is captured, the packets are classified per transport or application protocol detected and per day. The data is stored in two formats. The first one is PCAP, so that the source of evidence is collected in its raw state and, therefore, it is available for the investigator to analyse if they deem it necessary. For each traffic capture, the following associated data are stored as well: the file generated, its name, its size, the start and end date-time of the acquisition, and its SHA1 and MD5 hash codes. The other retrievable format is CSV, which facilitates the analysis of the filtered fields.
- **Log Service.** Even though the analysis of each packet is carried out in the following stage, the results of these analyses are stored and updated accordingly. The service stores the most relevant information obtained from each packet, which varies depending on the captured protocol, and it does the same with the information collected in the log files, but usually the filtered fields are: source and destination address of the packet, source and destination port, the source and destination ID, the vendor of the device, and which specific operation the packet is performing.

3.2.2. Network traffic auditing

This stage uses the data collected and stored in the previous phase as an input. Firstly, the network capture is preprocessed to extract different features directly from the network packet in real-time with the goal of obtaining the useful data as soon as possible. These filtered data are later used to present information regarding the behaviour of the network and the devices in it. Apart from the real time analysis, the node is capable of performing an offline one multiple times per day or upon request, using all the data collected. This can be done since IoT protocols are not encrypted, and the ones that are, such as Z-Wave or Zigbee, can be easily decrypted, as seen in Badenhop et al. [30], Yassein et al. [31]. To make this stage possible, the following services have been developed:

- **Feature Extraction**. Using the *Scapy* framework and the data obtained in the previous stage, the features selected as relevant from a forensic standpoint for each protocol, which are presented in Table 2, are extracted automatically. The aim is to gather and filter the information that is relevant to perform the identification and analysis phases of the forensic process.

Table 3
Performance metrics of machine learning models.

Model	Edge-IIoTset				Custom dataset [44]			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
Random Forest	95,46%	90,33%	87,19%	87,76%	98,54%	98,36%	97,98%	97,98%
XGBoost	**98,17%**	94,57%	94,50%	94,47%	**99,91%**	**99,41%**	**99,08%**	**99,24%**
LightGBM	97,94%	**95,24%**	**95,07%**	**95,11%**	99,90%	98,89%	99,13%	99,01%

- **Real-Time Network Traffic Auditing**. This service processes the features extracted and gathers knowledge based on them in real time. Given the interoperability and the high number of devices in the environment, the goal of this analysis is to identify them and determine their behaviour. As a result, the study is based on the communication between the devices. One of the main activities performed in this process is the identification of the connected devices and their characteristics, such as their vendor. To extract this information, we use the Wireshark Manufacture database [32] to identify them. In addition, this service is also capable of extracting the features that are present in the packet information in hexadecimal value. It also translates them to their meaning in terms of the packet field that they represent to facilitate the reading for the investigator.
- **Batch Network Traffic Auditing**. This is the service that carries out the periodic analysis using the information collected. During this offline analysis, useful forensic information is obtained about the devices and their context so that the investigator can keep track of the changes in the network. In addition, this analysis is performed once that it has been confirmed, using the hash code of the captured files, that the data have not been altered, and, therefore, the integrity has been maintained.

3.2.3. Monitoring & threat detection

As previously stated, our monitoring and alert service employs a robust IDS fortified with ML algorithms to attain high performance in the tabular classification of packets, thereby providing a solution for real-time threat detection and incident response. This is achieved through the advantages inherent to the Edge Computing environment in conjunction with ML techniques [33]. This integration fortifies the network's security and improves the efficiency and accuracy of forensic investigations, enabling proactive identification and mitigation of potential security incidents.

Our IDS leverages ML techniques, taking advantage of their proven effectiveness in environments with unknown and evolving threats, and bearing in mind that the manual formulation of attack-specific rules is challenging in diverse and dynamic environments. The effectiveness of ML in these scenarios is well-documented in the literature, with several studies such as those of Roldán et al. [34], Suthishni and Kumar [35] highlighting its superiority over traditional methods in handling complex and variable threat landscapes. The advantages of using a machine-learning-based IDS are numerous, and include:

- *Enhanced Accuracy*. ML algorithms are highly effective at processing large datasets and identifying complex patterns and anomalies that may be difficult for humans to detect or define explicitly.
- *Adaptive Learning*. IDS can continuously evolve and adapt to new attack vectors and changes in network behaviour through the use of ML.
- *Robust Anomaly Detection*. ML enables the identification of anomalies by modelling what is considered "normal" within a system. This is a difficult task for rule-based systems.

IDS based on ML implementation. The hardware setup used to train and test these models has been a Raspberry Pi 4 with 2 GB of RAM memory. This aspect has been decisive in the choice of algorithms to be considered.

The ML techniques that our IDS implementation considers are: (RF) [36], XGBoost [37], and LightGBM [38]. These algorithms are excellent for tabular classification, as demonstrated in Shwartz-Ziv and Armon [39]. Fortunately, there is scientific evidence in the literature that demonstrates the feasibility of these algorithms under similar memory and processing circumstances [40–42]. To determine the most effective technique, we conducted experiments using two datasets: the Edge-IIoTset [43], and a dataset generated by the authors of Ruiz-Villafranca et al. [44], with both representing similar environments with IoT and OT protocols and communications with an edge layer. These datasets are used as benchmarks to evaluate the performance of algorithms, aiding in informed decisions about their suitability for our specific use case.

Both datasets, namely Edge-IIoTset, and the dataset obtained from Ruiz-Villafranca et al. [44], underwent a standardized preprocessing procedure to make them compatible with the ML techniques employed. This section outlines the key steps involved in the preprocessing stage:

- **Data Cleaning**. Identifying and handling missing or erroneous data to ensure the quality and reliability of the datasets.
- **Data Transformation**. Converting and standardizing data types, addressing outliers, and transforming variables as necessary for compatibility with the chosen ML algorithms.
- **Feature Engineering**. Creating new features or modifying existing ones to enhance the predictive power of the models.
- **Data Optimization**. Ensuring that the datasets are efficiently structured and formatted for optimal ML performance.

After preprocessing, the datasets were divided into training (70%) and testing (30%) sets. During the training stage, Random Search [45] was used for the parameter tuning of each model, (the selected parameters are detailed in Table 4), and Extremely Randomized Trees [46] were used for feature selection to extract the most relevant information.

Table 4
List of best parameters found using Random Search for both datasets.

Algorithm	Best hyperparameters EdgeIIoTset	Best hyperparameters [44]
RF	max_depth: 12 n_estimators: 174 criterion: gini max_features: auto random_state: 42	max_depth: 17 n_estimators: 151 criterion: gini max_features: auto random_state: 42
Xgboost	learning_rate: 0.1 eta: 0.1 max_depth: 6 subsample: 0.8 seed: 42	learning_rate: 0.0993 eta: 0.12 max_depth: 9 subsample: 0.5 seed: 42
LightGBM	learning_rate: 0.0952 max_bin: 20 max_depth: 15 num_leaves: 80 subsample:0.75	learning_rate: 0.104 max_bin: 16 max_depth: 20 num_leaves: 66 subsample:0.81

To evaluate the models, a cross-validation process was conducted using a 10 k-fold approach. The model with the best score during validation was selected for further analysis. For comprehensive model assessment, metrics such as accuracy, precision, recall, and F1-score were employed. These metrics provide valuable insights into the models' performance and aid in selecting the most suitable option for our specific use case.

Algorithms Evaluation. The results in Table 3 shows the performance of each Machine Learning model. Random Forest exhibits the poorest performance metrics out of the three models, indicating a tendency to fail to classify certain types of network traffic correctly. In contrast, XGBoost demonstrates exceptional performance, particularly in the second dataset, achieving near-perfect accuracy, precision, recall, and f1-score with an overall performance of 98%–99% in every metric. This indicates an excellent ability to generalize across various network environments and attack vectors. However, the lower performance on the Edge-IIoTset dataset suggests a potential overfitting issue or less effectiveness in handling the specific challenges presented by that dataset. LightGBM, while slightly trailing behind XGBoost in some metrics, demonstrates remarkable consistency across both datasets. Its strength lies in balancing high accuracy with computational efficiency. This aspect is crucial in a real-time IoT forensic environment, where processing speed and model responsiveness are critical.

In summary, based on its overall performance and resource efficiency, LightGBM is the most suitable model for our forensic edge node proposal. Its consistently high performance across diverse datasets together with its computational efficiency make it ideal for real-time applications in varied and dynamic network environments. LightGBM's ability to offer quick and reliable analyses aligns well with the needs of forensic investigators, who require rapid insights without compromising on accuracy. The adoption of LightGBM could enhance the adaptability and effectiveness of our cybersecurity infrastructure, making it a valuable tool in the evolving landscape of digital forensics.

3.2.4. Information display

Once we have collected and analysed the data obtained from the network, it is important to display the information gathered. Therefore, this last stage is in charge of presenting the data in a clear and intuitive way to the investigator, avoiding the need for them to access the node and read any log, file, or raw data, although they can do so if they deem it. The services also provide confidentiality, as it is necessary to be authenticated in order to access the information that it is shown using a web front-end. Furthermore, thanks to the threat detection system, the alerts generated are shown in this stage as well. To implement these functionalities, we have used the Streamlit framework [47] to develop the following services:

- **Alert service**. This service is a critical component of network security. It is launched the instant that a potential threat is detected and serves as a sentinel that monitors the network's activity. Its primary function is to identify irregularities, anomalies, or suspicious patterns that deviate from established norms. Upon detection, the "Alert Service" notifies the investigator promptly, indicating the presence of a security incident. Its effectiveness is bound to its relationship with an IDS powered by ML algorithms.
- **Authentication service**. This service controls access to the information that is generated and stored on the forensic node. It is the first service that the forensic investigator must access in order to see the information. A authentication is implemented via a personal user ID and a secure password that is generated during the deployment of the forensic node. In addition, extra hardening measures are applied, such as an anti-brute-force systems, which temporally blocks the IP address which tries to log in more than five consecutive times using an invalid user and password combination.
- **Real-time live dashboard**. This shows the most relevant information obtained from the analysis. The dashboard updates the information charts and tables automatically when new results are ready to be shown. The dashboard is divided into two pages:

- *Live Dashboard*. This is the main page of the service and the landing page that the investigator sees after logging in. The page is divided into different sections: an area chart displaying the total number of packets captured per minute and the number of packets per protocol, a table providing access to the information regarding the last ten captured packets, and a table showing the vendors of the devices connected to the network. Additionally, the page features a dynamic graph node display. This graph node presents a visual representation of the network, displaying the connections and interactions between different nodes in real-time. It is a crucial functionality for investigators to comprehend the network's structure and communication flow. Additionally, the page includes a table with alerts, their importance, and a brief description of each. Thus, this comprehensive dashboard serves as an essential tool for investigators, offering both detailed data and visual insights into the network's activities.
- *Daily Report*. This page presents a detailed report generated by the offline analysis service, providing forensic investigators with valuable insights into network activities over the past ten days. The interface is user-friendly and offers a comprehensive overview of key metrics, including traffic patterns, anomalies, and notable events. Daily reports can be accessed as well, allowing for a deeper understanding of historical network behaviour. Furthermore, the investigator has the option to download traffic captures collected during these periods in either of the two formats in which they are stored. This ensures accessibility and compatibility with various forensic tools and workflows.

Algorithm 1 is presented in pseudocode format and provides a comprehensive overview of its functionalities and workflow. The algorithm serves as a representation of the operations and decision-making processes carried out by the forensic node, offering an intuitive approach to understanding its workflow.

Algorithm 1 The functionality of the Forensic Edge Node.

```
Input: Network traffic flows from the IoT gateways and devices
Machine learning model previously trained and deployed as IDS on the node
Output: Network traffic processed
Evidence Collection:
Edge node captures the traffic received, then it sends the traffic to the Network Traffic Auditing service
The node stores the traffic received into PCAP and CSV files, and it obtains its SHA1 and MD5 hash codes for integrity purposes
Network Traffic Auditing:
Receives the traffic capture and extracts the most relevant features depending on the protocol
Sends the traffic to the real-time and batch auditing service to extract information of the topology and devices connected
The data from both services is sent to the ML-based-IDS deployed as a monitoring and threat detection service
Monitoring and Threat Detection:
Receives the traffic information and packets captured
for packet in traffic do
    classification = detect_anomalies(packet)
    if classification is normal then
        return
    else
        packet.tag ← classification
        Sends the packet to Alert Service to display it
    end if
end for
Information Display:
Receives the traffic that has been analysed
if packet contains tag then
    Alert Service analyzes the tag and notifies the investigator
end if
if Investigator is authenticated then
    Shows the dashboard with the network information and the possible alerts
end if
```

3.3. Deployment workflow

To guarantee that the node operates optimally, it is essential to establish a well-defined deployment workflow. The process of setting up the node and ensuring its seamless integration into the network is guided by this structured roadmap. The steps of the workflow are illustrated in Fig. 2, which provides a visual guide to complement the textual descriptions and facilitates a comprehensive understanding of the deployment process.

The initial step for a forensic researcher is to study the environment's characteristics in which the forensic node will be deployed. This is an analytical phase where the characteristics are examined in detail. It is not only about comprehending the network but also determining whether deploying a forensic edge node is necessary and advantageous. This phase serves as a preliminary measure to

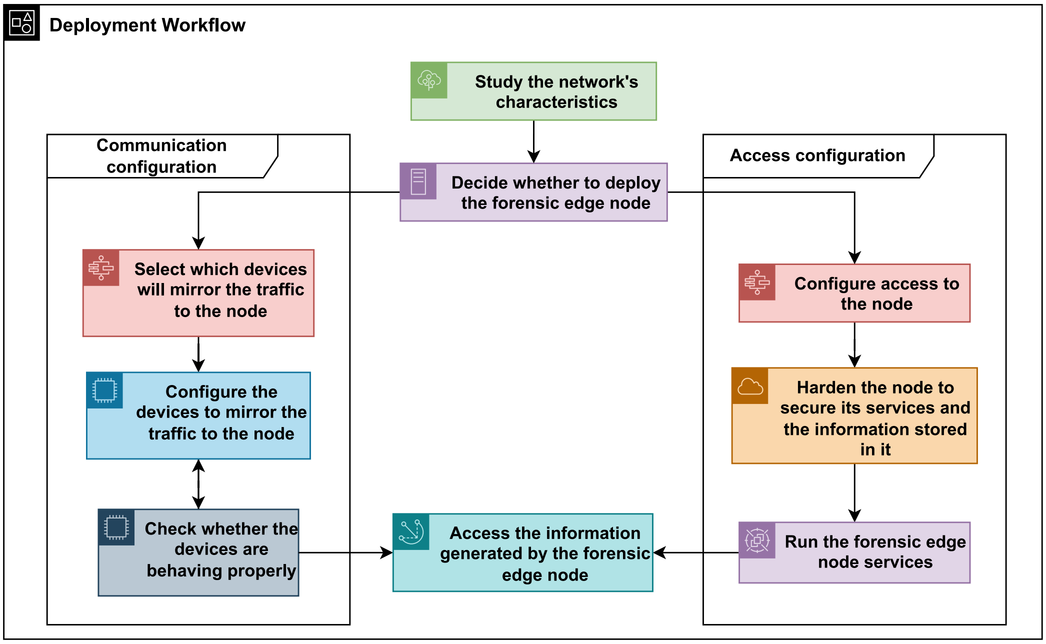

Deployment Workflow
Study the network's characteristics
Decide whether to deploy the forensic edge node
Communication configuration
Select which devices will mirror the traffic to the node
Configure the devices to mirror the traffic to the node
Check whether the devices are behaving properly
Access configuration
Configure access to the node
Harden the node to secure its services and the information stored in it
Run the forensic edge node services
Access the information generated by the forensic edge node

Fig. 2. Forensic edge node's deployment workflow.

evaluate the compatibility and potential effectiveness of introducing a forensic edge node into the current network infrastructure. It entails a comprehensive analysis of the network's structure, traffic patterns, and potential vulnerabilities.

After deploying the node, it is necessary to follow two configurations that can run in parallel. Each configuration consists of three steps:

Communication configuration. This configuration segment is a crucial phase that establishes the foundation for the entire workflow. It starts with selecting devices to mirror traffic to the node. This process is critical to ensure accurate data reflection for analysis. The devices are then configured to effectively mirror this traffic. This involves fine-tuning device settings to ensure maximum efficiency and accuracy in data mirroring. The last step in this segment is to check whether the configured devices are behaving as intended. This step involves validation and verification to ensure reliability and consistency in data mirroring.

Access configuration. This configuration focuses on establishing and securing access to the node. The first step involves configuring access, which includes setting up authentication mechanisms and access controls to prevent unauthorized access. The emphasis then shifts to hardening the node, which involves securing its services, and safeguarding stored information. The task is to implement security measures such as encryption, intrusion detection systems and regular audits. This highlights the inherent focus on security measures that are integral in maintaining data integrity and confidentiality. The segment concludes with running forensic edge node services to allow the investigator to remotely access them.

After successfully completing and finalizing these configurations, the investigator can establish a connection with the forensic node. This connection enables the investigator to access and scrutinize the information generated by the node, which is a product of the node's analysis and processing, providing valuable insights that aid the investigator in their examination. Therefore, completing the configurations is crucial in the deployment process, as it signifies the investigator's transition from the setup phase to the use phase of the forensic node.

4. Proof of concept

In this section, an example of the deployment of the forensic node is presented using a Multi-access Edge Computing (MEC)-IIoT scenario as a testing environment with the aim of demonstrating the functionality and effectiveness of the proposal, as well as showing the workflow that the investigator can follow when using it. Firstly, the resources used to set up the testing scenario are described. Secondly, the details of the proof of concept are presented, explaining how the deployment is carried out, and how the different services of the forensic edge node work in the set-up environment. And, finally, we explain how the investigator can access and interact with the interface provided by the node to study and retrieve the information gathered. In addition, a cyberattack is introduced in this experiment in order to test the validity of the threat detection service, and show how the IDS and the alert service operate.

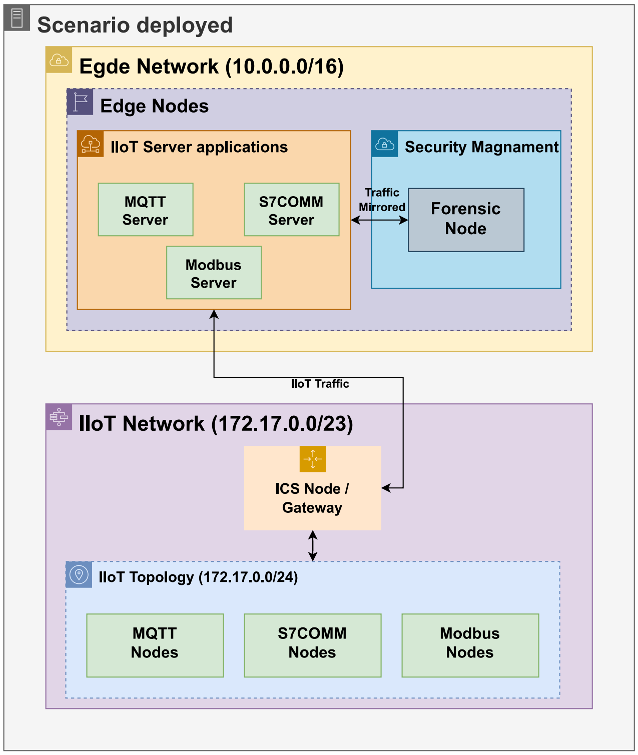


Fig. 3. IIoT scenario deployed.

4.1. *Experimental setup*

To develop and implement the proof of concept described above, we used a workstation with 32 GB RAM and an Intel i7-10875H processor, which runs Ubuntu 22.04 LTS. The hypervisor, namely VirtualBox 7.0.6, was used to run the emulator deployed that implements the MEC-IIoT scenario. The forensic edge node was deployed on the MEC emulated network, which is detailed in the following section. As a Mininet host, this type of host shares the computational and memory resources of the host machine. To ensure a realistic deployment of our proposal, we limited the Mininet host to the specifications of a Raspberry Pi 4, with a CPU speed of 1.80 GHz and a maximum of 2 GB of RAM. These specifications are similar to the hardware used during the implementation of the IDS-based ML described in Section 3.2.3.

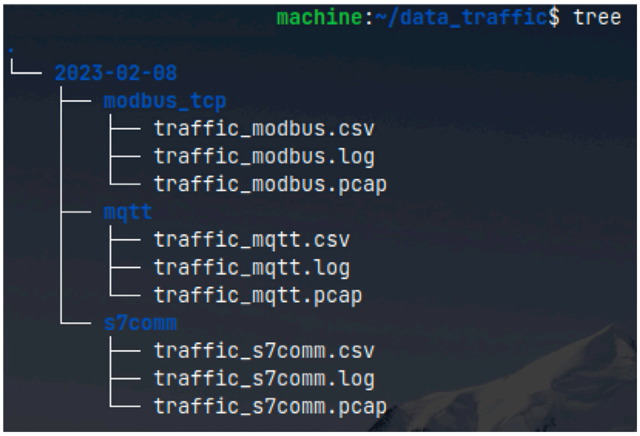
4.2. *Scenario definition*

With the aim of showing the functionality of our proposal, a scenario was designed that presents an IIoT topology that operates together with an MEC one. This allows the use of our forensic edge node in an edge deployment approach. Consequently, the environment is comprised of two enterprise networks with the following features:

IIoT network. This network is divided into two topologies: one focused on the communication between IoT devices which use the MQTT protocol and operate as temperature sensors that publish information to a topic, with the messages being sent to a mosquitto broker: the other being an Operational Technologies (OT) topology that is deployed with three devices that use the Modbus/TCP protocol allocated to it, one of which operates as a server that receives the data from the clients. One client writes a sequence of values that can be either “true” or “false”, while the second reads the sequence stored on the server. In addition, the S7COMM protocol can be found in the topology as well, and this is used by three devices that have a similar structure to Modbus/TCP applications. For this protocol, the writer sends a random string that is stored on the server, and the reader client periodically reads it. Finally, an ICS node is deployed, that will mirror the IIoT traffic to the forensic edge node.

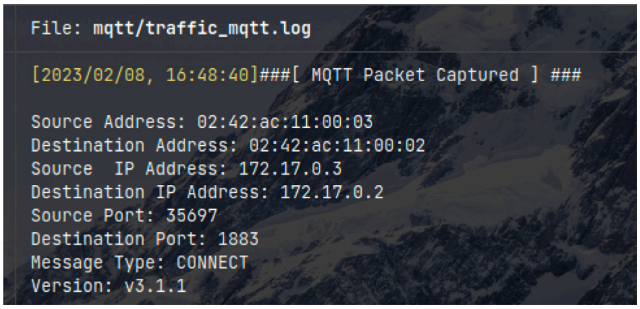
MEC network. The edge topology deployed has 64 hosts connected and communicated which each other in a 3-tier network architecture. This means that the topology allocates two core switches and two aggregation switches for each one. Also, redundancy is considered in the topology to ensure availability by implementing a Software Define Network (SDN) controller, which uses RYU [48], and the spanning-tree protocol [49]. Our forensic edge node is deployed in this topology. Furthermore, it includes a Long Term Evolution (LTE) station, offering the possibility of connecting to the node using a mobile communication protocol.

For the design, development, and implementation of this scenario, the MECInOT emulator was used [50]. This emulator allows the implementation of the scenario described above thanks to the use of virtualization technologies, using Docker networks [51]



```
machine:~/data_traffic$ tree
.
└── 2023-02-08
    ├── modbus_tcp
    │   ├── traffic_modbus.csv
    │   ├── traffic_modbus.log
    │   └── traffic_modbus.pcap
    ├── mqtt
    │   ├── traffic_mqtt.csv
    │   ├── traffic_mqtt.log
    │   └── traffic_mqtt.pcap
    └── s7comm
        ├── traffic_s7comm.csv
        ├── traffic_s7comm.log
        └── traffic_s7comm.pcap
```

Fig. 4. Storage structure.



```
File: mqtt/traffic_mqtt.log

[2023/02/08, 16:48:40]###[ MQTT Packet Captured ] ###

Source Address: 02:42:ac:11:00:03
Destination Address: 02:42:ac:11:00:02
Source  IP Address: 172.17.0.3
Destination IP Address: 172.17.0.2
Source Port: 35697
Destination Port: 1883
Message Type: CONNECT
Version: v3.1.1
```

Fig. 5. Log file example.

and containers to deploy the IIoT network, and openLEON [52] to implement the MEC topology. Fig. 3 shows the scenario deployed for the proof of concept and gives more detail about the IP addresses of each topology defined.

4.3. Experiment definition

Now the scenario has been described, we can detail how the experiment is defined in order to demonstrate the functionality of our proposed forensic edge node and which process is followed to do so. The goal is to show how the forensic node captures and analyzes the network data, as well as how the investigator can see the information gathered using the live dashboard. In this way, it is possible to have a global vision of how the different services exchange data, and how they can be accessed. Additionally, a malicious node will be introduced to perform various scanning techniques. With this goal in mind, the experiment is divided into two main workflows.

4.3.1. Internal workflow

This workflow corresponds to the tasks carried out in the “Data Collection” and “Identification & Analysis” stages. When the edge node is deployed in the topology, the firewall tool *iptables* is configured to send the traffic directly to a queue which the capture service is listening to. In this way, the traffic is captured and stored on the forensic edge node. The storing operation is performed in a structured way so that the data are easy to access. Specifically, the network captures are stored in the following way: */<**data_traffic**>/<**capture_date**>/<**protocol**>/*. The directory structure for our proof of concept can be seen in Fig. 4, which also presents the three formats in which the data is stored. In each directory, the traffic captures in PCAP and CSV can be found, together with the result from the analysis, which is stored in a log file. The format of this log, together with its content is shown in Fig. 5.

4.3.2. Investigator workflow

This workflow is strongly related to the “Information Display” stage, in which the main services that the investigator can check in order to study the information gathered by the forensic edge node are allocated. Through this workflow, which is graphically depicted in Fig. 6, they are able to understand the network’s current and past states and behaviours, and to receive notifications regarding possible attacks or anomalous activities occurring in it. The first step is to introduce the credentials provided in the login

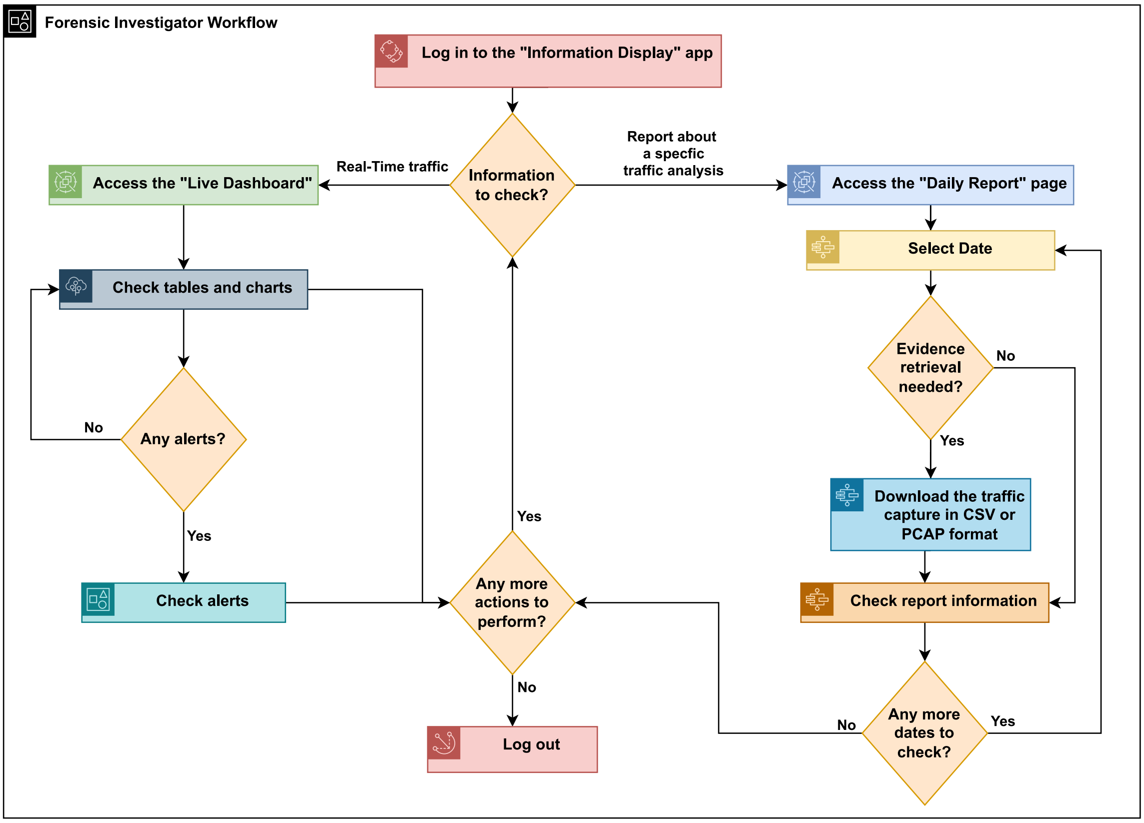

Forensic Investigator Workflow
Log in to the "Information Display" app
Information to check?
Real-Time traffic
Report about a specfic traffic analysis
Access the "Live Dashboard"
Access the "Daily Report" page
Select Date
Check tables and charts
Evidence retrieval needed?
No
Yes
Any alerts?
No
Yes
Download the traffic capture in CSV or PCAP format
Check alerts
Any more actions to perform?
Yes
No
Check report information
Log out
Any more dates to check?
No
Yes

Fig. 6. Forensic researcher workflow for using the "Information Display" service.

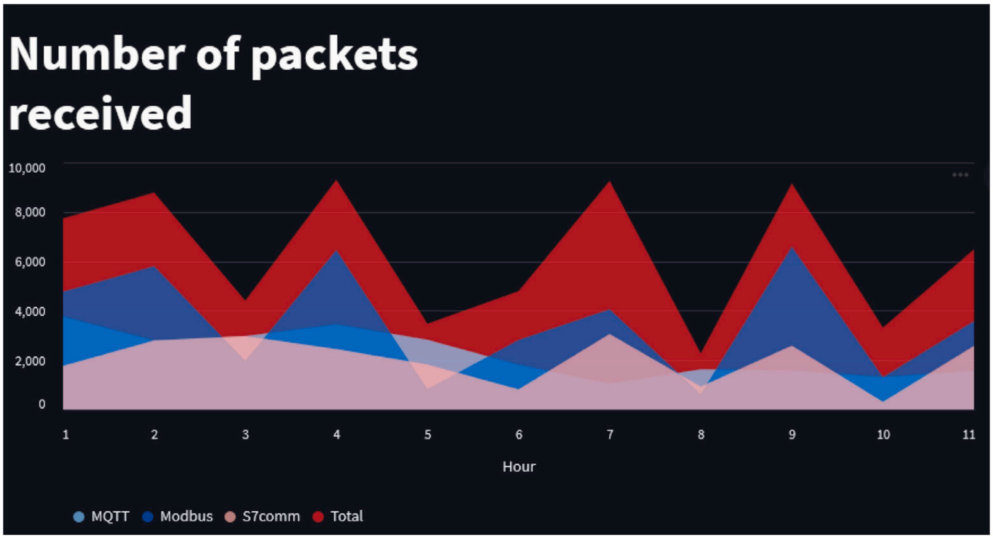
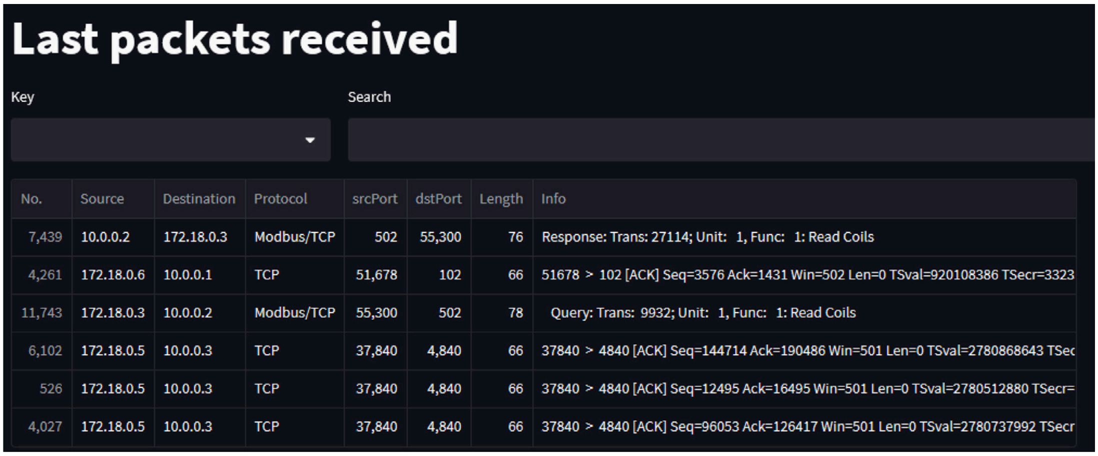


Fig. 7. Packets received in real-time by the forensic edge node.

form, which were generated during the deployment, as was detailed in Section 3.2.4. After that, the investigator should consider which information they would like to check. If they want to study the real-time traffic, Figs. 7, 8 and 9 show the relevant information that the investigator could find on the dashboard.

Also, if the IDS finds any malicious traffic, such as the scanning packets introduced by the malicious node, the system generates an alert, a process that can be observed in Fig. 10. This figure illustrates how alerts are presented to the investigator, with a table



Last packets received

Key

Search

No.	Source	Destination	Protocol	srcPort	dstPort	Length	Info
7,439	10.0.0.2	172.18.0.3	Modbus/TCP	502	55,300	76	Response: Trans: 27114; Unit: 1, Func: 1: Read Coils
4,261	172.18.0.6	10.0.0.1	TCP	51,678	102	66	51678 > 102 [ACK] Seq=3576 Ack=1431 Win=502 Len=0 TSval=920108386 TSecr=3323
11,743	172.18.0.3	10.0.0.2	Modbus/TCP	55,300	502	78	Query: Trans: 9932; Unit: 1, Func: 1: Read Coils
6,102	172.18.0.5	10.0.0.3	TCP	37,840	4,840	66	37840 > 4840 [ACK] Seq=144714 Ack=190486 Win=501 Len=0 TSval=2780868643 TSec
526	172.18.0.5	10.0.0.3	TCP	37,840	4,840	66	37840 > 4840 [ACK] Seq=12495 Ack=16495 Win=501 Len=0 TSval=2780512880 TSecr=
4,027	172.18.0.5	10.0.0.3	TCP	37,840	4,840	66	37840 > 4840 [ACK] Seq=96053 Ack=126417 Win=501 Len=0 TSval=2780737992 TSecr

Fig. 8. Last packets sniffed by the forensic edge node and their information.

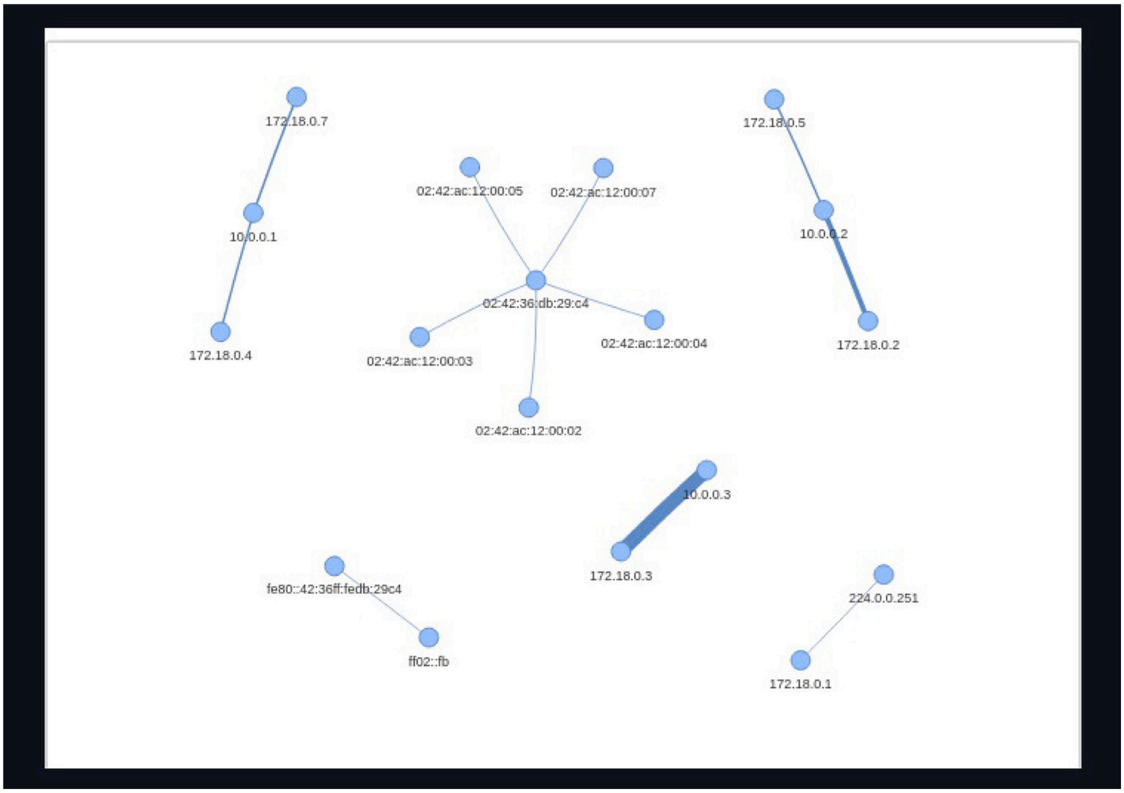
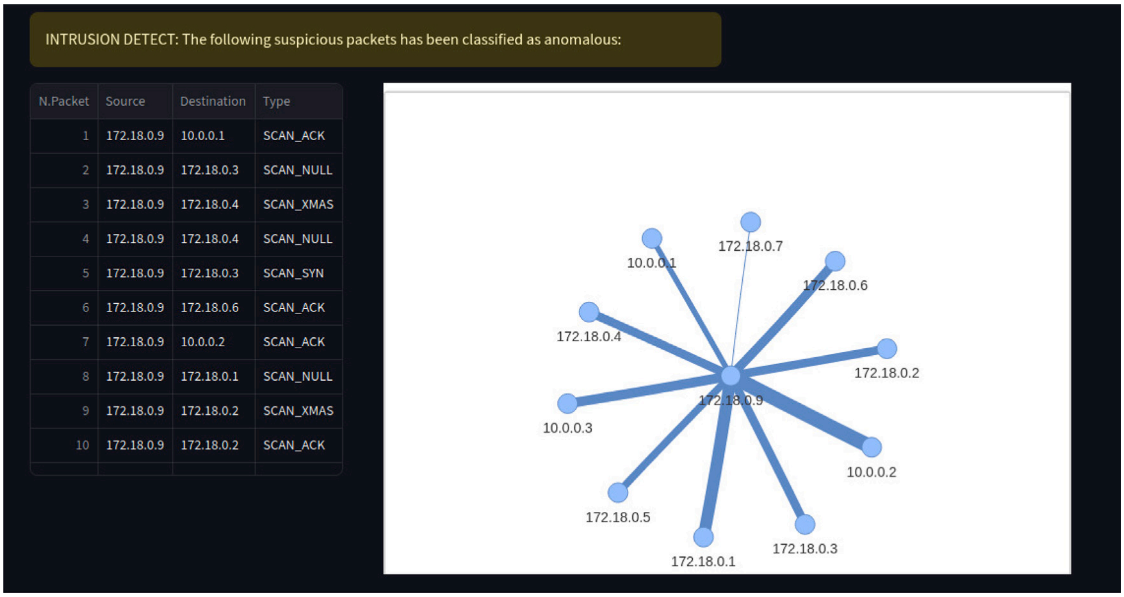


Fig. 9. Network graph schema detected by the forensic edge node.

displaying the type of attack detected, as well as the sender and receiver hosts of the packet. Additionally, a graph depicting the communication between the devices is provided.

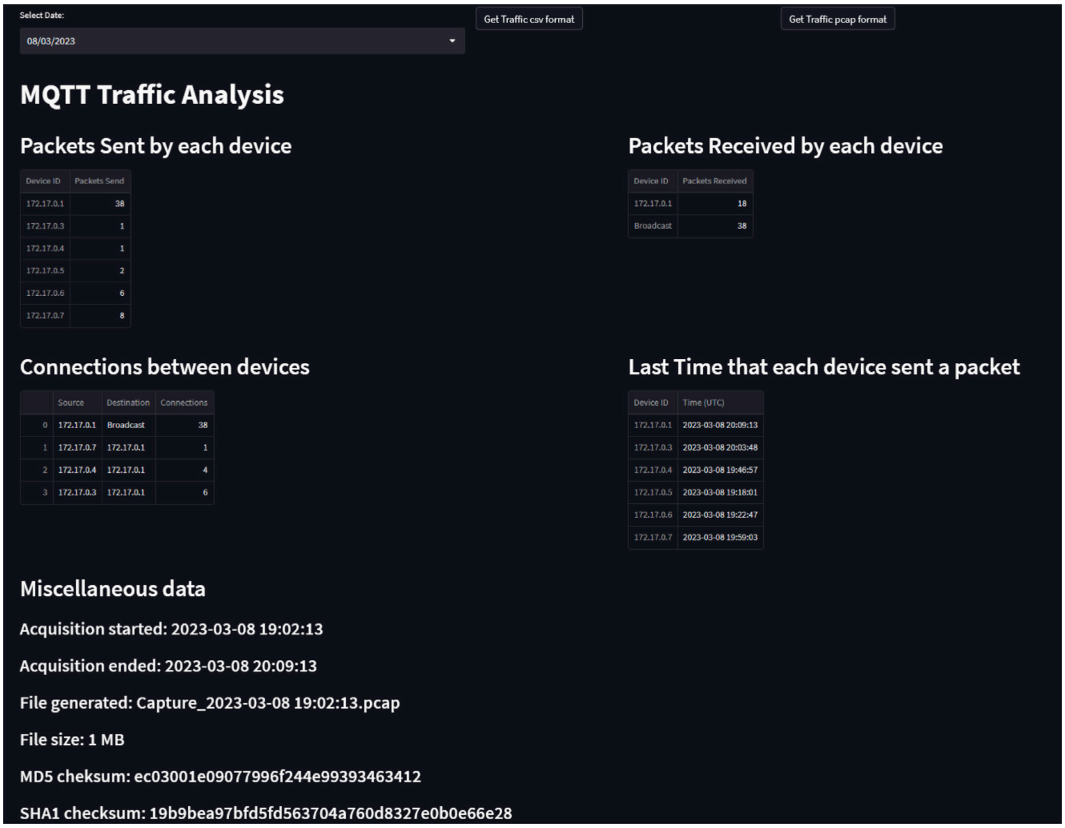
On the other hand, if the investigator would like to check the offline analysis performed with the information that is periodically gathered, they obtain the result presented in Fig. 11. In this case, the analysis shows the information gathered from the MQTT protocol. In particular, it displays the different connections detected for each IoT device, the most frequently established ones, the number of packets sent and received by each device, and when the last communication was established. With respect to the information that can be found about the collected network traffic, the file’s log generated included data such as name, size or SHA1 and MD5 hash. Finally, this page allows the investigator to download the sources of evidence captured if they deem it necessary,



INTRUSION DETECT: The following suspicious packets has been classified as anomalous:

N.Packet	Source	Destination	Type
1	172.18.0.9	10.0.0.1	SCAN_ACK
2	172.18.0.9	172.18.0.3	SCAN_NULL
3	172.18.0.9	172.18.0.4	SCAN_XMAS
4	172.18.0.9	172.18.0.4	SCAN_NULL
5	172.18.0.9	172.18.0.3	SCAN_SYN
6	172.18.0.9	172.18.0.6	SCAN_ACK
7	172.18.0.9	10.0.0.2	SCAN_ACK
8	172.18.0.9	172.18.0.1	SCAN_NULL
9	172.18.0.9	172.18.0.2	SCAN_XMAS
10	172.18.0.9	172.18.0.2	SCAN_ACK

Fig. 10. Alerts generated by the IDS service.



MQTT Traffic Analysis

Packets Sent by each device

Device ID	Packets Send
172.17.0.1	38
172.17.0.3	1
172.17.0.4	1
172.17.0.5	2
172.17.0.6	6
172.17.0.7	8

Packets Received by each device

Device ID	Packets Received
172.17.0.1	18
Broadcast	38

Connections between devices

	Source	Destination	Connections
0	172.17.0.1	Broadcast	38
1	172.17.0.7	172.17.0.1	1
2	172.17.0.4	172.17.0.1	4
3	172.17.0.3	172.17.0.1	6

Last Time that each device sent a packet

Device ID	Time (UTC)
172.17.0.1	2023-03-08 20:09:13
172.17.0.3	2023-03-08 20:03:48
172.17.0.4	2023-03-08 19:46:57
172.17.0.5	2023-03-08 19:18:01
172.17.0.6	2023-03-08 19:22:47
172.17.0.7	2023-03-08 19:59:03

Miscellaneous data

Acquisition started: 2023-03-08 19:02:13

Acquisition ended: 2023-03-08 20:09:13

File generated: Capture_2023-03-08 19:02:13.pcap

File size: 1 MB

MD5 cheksum: ec03001e09077996f244e99393463412

SHA1 checksum: 19b9bea97bfd5fd563704a760d8327e0b0e66e28

Fig. 11. Reporting page in the “Information Display” service.

offering them the option of obtaining a PCAP file with the raw data captured by the forensic edge node, or a CSV file with the filtered data.

5. Discussion

In this article, we have addressed the state of IoT forensic investigations, focusing on the requirements and challenges that this environment involves. Due to its fundamental differences with respect to conventional scenarios, solutions based on existing forensic techniques are not suitable in this environment. Aspects such as the high number of devices in a network, the vast quantity of data exchanged in it, its lifetime, and the difficulty of effectively carrying out the acquisition methods, have a huge impact on the outcome of an investigation. As a result, new methods are needed to ensure that investigations are carried out in an efficient and complete way.

After studying the related work, it can be concluded that there are few pieces of research that focus on covering all the three main practical phases of the forensic process, namely identification, acquisition, and analysis. In addition, the ones that offer interesting proposals that rely on external devices, architectures, frameworks, or platforms to assist in investigations commonly fail to present mature enough schemes, introducing yet-to-be-developed ones. However, the results are promising and show that using external solutions to assist in IoT investigations may be a useful way to approach the design of IoT-centred tools.

With the aim of solving some of the issues mentioned above, this article presents an edge computing node that is capable of performing real-time traffic analysis to assist in the investigation process. By integrating this node into the IoT the network, the investigator is able to monitor, identify, acquire, and analyse the network packets that are exchanged in it. Furthermore, it does so without causing any interference in the network. As a result, the examiner can easily track the behaviour of IoT devices and have access to the data that they generate. In the same way, by using an IDS based on ML, the edge node is capable of detecting threats and initiating the forensic process, reducing the response time to a minimum. Moreover, the information is presented to the investigator through a self-hosted dashboard that lays out the results of the monitoring and analysis phases, and allows the retrieval of the data collected.

When tested in a proof of concept to evaluate its functionality and feasibility, the results show that it is capable of correctly identifying the devices in an IoT network, as well as of studying and collecting the network traffic in it, even when multiple IoT protocols are used. In addition, its ability to detect threats is evaluated, confirming that its effectiveness as an incident response tool. Therefore, it can be concluded that the proposal is suitable for use in IoT forensic investigations, both in a proactive and reactive approach.

In summary, in a schematic way, we can highlight the following findings:

- The process of identifying, acquiring and analysing data sources in IoT device testing varies from conventional scenarios. However, IoT protocols emerge as a standardized source of evidence that is used in multiple contexts, so IoT-centred tools that follow a generic approach can be proposed if using network traffic as a source.
- The proposed IoT node is capable of retrieving network packets in real time to assist in the identification, acquisition and analysis phases of the examination process, both in a reactive and a proactive way.
- The proactive approach allows the node to detect and mitigate various threats in real time, as well as automatically start the forensic process, which gives the scenario a high degree of forensic soundness and readiness.
- The dashboard allows a quick evaluation of the network in real time and presents an analysis of the results to the investigator. It also enables the retrieval of the pieces of evidence collected by the node.

When it comes to ensuring that the proof-of-concept is sufficient to validate proposal, it is necessary to take into account the characteristics of both the forensic and the research field. There are two main factors. First of all, reproducibility and replicability, which are crucial in any scientific research. Secondly, that forensic investigations are shaped upon context, meaning that there are a certain set of actions that lead to the opening of the investigation. As a result, one examination might be different from another, even if the same type of devices are involved, as the way in which the actions materialize differ from investigation to investigation. These two concepts are entirely opposite, so prioritizing one over the other will lead to the experiment not fulfilling one of these requirements.

To solve this issue, the authors decided to try and design a scenario that resembles a real-life one, but aiming to generalize it as much as possible and use open-source tools that allow recreating the scenario to the greatest extent possible.

In terms of significance, we believe that the proof of concept carried out is sufficient, as the data used comes from external proposals. With this, we aim to rule out any possibility of having tailored-scenario-data that might suit our proposal but may not depict a real situation. In addition, it also allows other researches to be able to replicate the case study, which means that the results obtained can be easily checked. Finally, the data that is used is formed by following the protocols' standards, so its structure and content is the same as in any real scenario.

In terms of the scenario, it has been designed using the most common protocols that can be found on IIoT and IoT networks. In fact, we decided to use serveral to demonstrate that the results have not been obtained by overfitting to a certain protocol.

Under these circumstances, the authors believe that, even though it is difficult to simulate a real-life forensic scenario, the experiment carried out has been designed trying to be as close as possible, and, as a result, the proof of concept demonstrated the usefulness of the proposed tool.

6. Conclusions and future work

After completing the experiment, and gaining a significant amount of knowledge in the design of automatic solutions for IoT forensic investigations, the research questions formulated at the beginning of this research can be answered.

- **(RQ1)** Given the inability of IoT devices to be forensic-friendly, is it possible to develop a solution that can assist in making IoT scenarios forensic-ready to a certain extent?
- The difficulty, from a forensic standpoint, of IoT devices can be compensated for through the use of devices external to the IoT network that are capable of performing forensic tasks and are fully accessible for an investigator. In this way, there is an intermediary between the IoT devices and the examiner that provides the latter with a new way of interacting with the data that are generated inside the IoT network. However, this limits the handling of sources of pieces of evidence to mainly network traffic, as trying to access the non-volatile memory would require directly interacting with the IoT devices, and that can be done by the investigator manually with common forensic tools.
- **(RQ2)** Since it is crucial to reduce the response time in IoT forensic investigations to a minimum due to the short lifetime of the data, can an approach be proposed that is able to detect threats and take measures to collect and preserve data if an examination is needed?
- Considering the nature of the IoT, the data that must be evaluated in order to detect threats inside an IoT network is network traffic. Firstly, because it is the largest data source due to the interoperability of the environment, and, secondly, as it is the data source that most accurately and instantly represents the behaviour of a device. In this article, through the use of an IDS based on ML, it has been demonstrated that it is possible to use network packets for threat detection, and use this information to automatize the initial phases of a forensic investigation.
- **(RQ3)** Taking into account the difficulty of identifying the devices in an IoT forensic investigation, is it possible to track the behaviour of the IoT network and provide this data to the forensic investigator in order to facilitate the identification process?
- A tracking mechanism can be developed by monitoring the network packets that are exchanged in an IoT network. Upon discerning the key fields of the most popular protocols used by IoT devices, the task of effectively identifying and keeping track of them becomes feasible. However, in order to do so, it is necessary to be aware of the IoT-protocol use statistics and update the solutions accordingly, as generic ones cannot be designed to be compatible with all of them. Another interesting aspect is that, obviously, to have a clear picture of all the events that occurred during an incident, it is necessary for the monitoring device to be active when it arises, which is an unsolvable issue, and requires the IoT manager to have implemented it beforehand, which is not likely to happen. However, this tracking mechanism is a very useful resource for investigators to have in a reactive forensic examination as well.
- **(RQ4)** Considering that network traffic can be a useful source of evidence in IoT forensic examinations, and that it generates a great amount of data, can this data be studied, filtered, and collected following an IoT-centred approach in order for them to be used as a source of evidence?
- It can be done, but requires a thorough analysis of the protocols used in the environment in order to extract valuable information from them, as their structure varies depending on the one that is under study. This is a consequence of the IoT having such a high degree of heterogeneity and being used in many contexts. Taking all this into account, this experiment has proven that there are clear benefits to be gaining from studying, collecting and filtering IoT network traffic, as it assists the investigator in crucial tasks of the forensic process, such as the following:

  - Determining how many devices, and of which type, are present in an IoT network, and, when used in a proactive approach, what changes in terms of devices connecting and disconnecting from the network have occurred. This assists the investigator in the process of determining the range of the investigation and identifying the sources of evidence.
  - Making sure that there is, at least, one plausible useful source of evidence for the investigator to analyse. Considering how difficult the process of acquiring sources of evidence is in the IoT, with the non-volatile memory, and, especially, the volatile one, being quite challenging, having access to the network traffic can be crucial in an examination. In addition, network traffic is extremely volatile, so collecting it at the outset of the materialization of the incident can have a huge impact on the outcome of an investigation.
  - Automating the filtering and analysis process of the network traffic means reducing the vast amount of data that an investigator may find when carrying out an examination. Combining this aspect with having access to the raw traffic capture means offering a high degree of flexibility to the examiner.

This study represents an initial investigation into the potential applications of an edge computing node to enhance forensic investigations within the domain of the IoT. Consequently, there exists an extensive range of fascinating associated projects, prospective avenues of enquiry and research challenges [53] that warrant further analysis:

- **Integration of Explainable AI (XAI):** The utilization of XAI in conjunction with artificial intelligence (AI) in the context of IoT forensics is intended to provide transparency and understanding in the decision-making process, thereby aiding accountability and trust. It allows investigators to understand and trust the decisions made by the AI, which is crucial in forensic investigations.

- **Self-Adaptive Systems deployment:** In order to be effective, Self-Adaptive Systems must be capable of adapting their own behaviour and structure based on their perception of their surrounding environment and the system itself. In the context of the IoT, developing such systems means creating IoT devices and networks that are able to autonomously manage their own protection and security, and to detect and mitigate cyber threats in order to ensure efficient operation and resilience against such threats.
- **Inclusion of Autonomic Computing Models:** Autonomic computing is a term used to describe the self-managing characteristics of distributed computing resources. These resources adapt to unpredictable changes while hiding their intrinsic complexity from operators and users. To advance these models in the context of the IoT, it is necessary to develop systems that are capable of supporting self-healing, self-configuring, self-optimizing, and self-protecting features. These systems will be expected to recover from failures, adapt to changing environments, improve performance over time, and protect against potential threats. This has the potential to significantly enhance the resilience and reliability of IoT infrastructures.
- **Blockchain Technology in IoT Forensics:** The utilization of blockchain technology can be employed in the enhancement of data integrity and traceability within IoT environments. By storing forensic data on a decentralized and immutable blockchain, investigators can ensure that the data has not undergone any form of tampering, while also being able to reliably trace the origin of any piece of information. This could significantly improve the reliability and efficiency of IoT forensic investigations. Furthermore, smart contracts on the blockchain have the potential to automate certain aspects of the forensic process, thus reducing the time and resources required for the investigation process. This represents a promising future direction for the field.

CRediT authorship contribution statement

Sergio Ruiz-Villafranca: Writing – review & editing, Writing – original draft, Software, Investigation, Formal analysis, Data curation, Conceptualization. **Juan Manuel Castelo Gómez:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **José Roldán-Gómez:** Writing – review & editing, Resources, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used is available online from different sources.

Acknowledgements

This work was supported by the Universidad de Castilla-La Mancha, Spain under the predoctoral contract 2022-PRED-20677 and the project 2023-GRIN-34056, both financed by the European Social Fund Plus (FSE+), and by the JCCM under the project SBPLY/21/180501/000195. Also, this work is part of the R&D project PID2021-123627OB-C52, funded by the MCIN and the European Regional Development Fund: "a way of making Europe", and it is also funded by MCIN/AEI/10.13039/501100011033 (FEDER, EU) under grant PID2022-142332OA-I00.

References

[1] Lionel Sujay Vailshery. Statista, IoT connected devices worldwide 2019–2030 - Statista, 2023, https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

[2] Kaspersky, Kaspersky security bulletin 2023. Statistics, 2023, https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2023/11/28102415/KSB_statistics_2023_en.pdf.

[3] M. Mansour, A. Gamal, A.I. Ahmed, L.A. Said, A. Elbaz, N. Herencsar, A. Soltan, Internet of Things: A comprehensive overview on protocols, architectures, technologies, simulation tools, and future directions, Energies 16 (8) (2023) http://dx.doi.org/10.3390/en16083465, URL: https://www.mdpi.com/1996-1073/16/8/3465.

[4] E. Oriwoh, D. Jazani, G. Epiphaniou, P. Sant, Internet of Things forensics: Challenges and approaches, in: 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2013, pp. 608–615.

[5] A. MacDermott, T. Baker, Q. Shi, IoT forensics: Challenges for the IoA era, in: 2018 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS, 2018, pp. 1–5, http://dx.doi.org/10.1109/NTMS.2018.8328748.

[6] I. Yaqoob, I.A.T. Hashem, A. Ahmed, S.A. Kazmi, C.S. Hong, Internet of Things forensics: Recent advances, taxonomy, requirements, and open challenges, Future Gener. Comput. Syst. 92 (2019) 265–275, http://dx.doi.org/10.1016/j.future.2018.09.058, URL: http://www.sciencedirect.com/science/article/pii/S0167739X18315644.

[7] F. Servida, E. Casey, IoT forensic challenges and opportunities for digital traces, Digit. Investig. 28 (2019) S22–S29, http://dx.doi.org/10.1016/j.diin.2019.01.012, URL: https://www.sciencedirect.com/science/article/pii/S1742287619300222.

[8] S. Perumal, N.M. Norwawi, V. Raman, Internet of things(IoT) digital forensic investigation model: Top-down forensic approach methodology, in: 2015 Fifth International Conference on Digital Information Processing and Communications, ICDIPC, 2015, pp. 19–23, http://dx.doi.org/10.1109/ICDIPC.2015.7323000.

[9] A. Nieto, R. Rios, J. Lopez, A methodology for privacy-aware IoT-forensics, in: 2017 IEEE Trustcom/BigDataSE/ICESS, 2017, pp. 626–633, http://dx.doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.293.

[10] M. Hossain, R. Hasan, S. Zawoad, Trust-IoV: A trustworthy forensic investigation framework for the Internet of Vehicles (IoV), in: 2017 IEEE International Congress on Internet of Things, ICIOT, 2017, pp. 25–32, http://dx.doi.org/10.1109/IEEE.ICIOT.2017.13.

[11] E. Al-Masri, Y. Bai, J. Li, A fog-based digital forensics investigation framework for IoT systems, in: 2018 IEEE International Conference on Smart Cloud, SmartCloud, 2018, pp. 196–201.

[12] DFRWS Attendees, A Road Map for Digital Forensic Research, Technical Report, DFRWS, 2010.

[13] V.R. Kebande, I. Ray, A generic digital forensic investigation framework for Internet of Things (IoT), in: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud, FiCloud, 2016, pp. 356–362, http://dx.doi.org/10.1109/FiCloud.2016.57.

[14] X. Feng, E.S. Dawam, S. Amin, A new digital forensics model of smart city automated vehicles, in: 2017 IEEE International Conference on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data, SmartData, 2017, pp. 274–279, http://dx.doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.47.

[15] M. Harbawi, A. Varol, An improved digital evidence acquisition model for the internet of things forensic I: A theoretical framework, in: 2017 5th International Symposium on Digital Forensic and Security, ISDFS, 2017, pp. 1–6.

[16] V.R. Kebande, N.M. Karie, A. Michael, S. Malapane, I. Kigwana, H.S. Venter, R.D. Wario, Towards an integrated digital forensic investigation framework for an IoT-based ecosystem, in: 2018 IEEE International Conference on Smart Internet of Things, SmartIoT, 2018, pp. 93–98.

[17] M.B. Al-Sadi, L. Chen, R.J. Haddad, Internet of Things digital forensic investigation using open source gears, in: SoutheastCon 2018, 2018, pp. 1–5, http://dx.doi.org/10.1109/SECON.2018.8479042.

[18] L. Sadineni, E. Pilli, R.B. Battula, A holistic forensic model for the Internet of Things, in: G. Peterson, S. Shenoi (Eds.), Advances in Digital Forensics XV, Springer International Publishing, Cham, 2019, pp. 3–18.

[19] N.K. Bharadwaj, U. Singh, Acquisition and analysis of forensic artifacts from raspberry pi an Internet of Things prototype platform, in: P.K. Sa, S. Bakshi, I.K. Hatzilygeroudis, M.N. Sahoo (Eds.), Recent Findings in Intelligent Computing Techniques, Springer Singapore, Singapore, 2019, pp. 311–322.

[20] D.H. Kasukurti, S. Patil, Wearable device forensic: Probable case studies and proposed methodology, in: S.M. Thampi, S. Madria, G. Wang, D.B. Rawat, J.M. Alcaraz Calero (Eds.), Security in Computing and Communications, Springer Singapore, Singapore, 2019, pp. 290–300.

[21] R. Jacob, A. Nisbet, A forensic investigation framework for Internet of Things monitoring, Forensic Sci. Int.: Digit. Investig. 42–43 (2022) 301482, http://dx.doi.org/10.1016/j.fsidi.2022.301482, URL: https://www.sciencedirect.com/science/article/pii/S2666281722001639.

[22] A. Karagiozidis, M. Gergeleit, An OT forensic model based on established IT forensics using IIRA, in: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation, ETFA, 2022, pp. 1–8, http://dx.doi.org/10.1109/ETFA52439.2022.9921588.

[23] P. Biondi, Scapy framework, 2023, https://scapy.readthedocs.io/en/latest/. (Accessed 21 March 2023).

[24] R. Russell, Iptables firewall application, 2023, https://www.netfilter.org/. (Accessed 21 March 2023).

[25] M. Fox, NetfilterQueue Python module, 2023, https://github.com/oremanj/python-netfilterqueue. (Accessed 21 March 2023).

[26] B. Mishra, A. Kertesz, The use of MQTT in M2M and IoT systems: A survey, IEEE Access 8 (2020) 201071–201086.

[27] D. Silva, L.I. Carvalho, J. Soares, R.C. Sofia, A performance analysis of Internet of Things networking protocols: Evaluating MQTT, CoAP, OPC UA, Appl. Sci. 11 (11) (2021) 4879.

[28] H. Hui, K. McLaughlin, S. Sezer, Vulnerability analysis of S7 PLCs: Manipulating the security mechanism, Int. J. Crit. Infrastruct. Prot. 35 (2021) 100470, http://dx.doi.org/10.1016/j.ijcip.2021.100470, URL: https://www.sciencedirect.com/science/article/pii/S1874548221000573.

[29] N. Goldenberg, A. Wool, Accurate modeling of modbus/TCP for intrusion detection in SCADA systems, Int. J. Crit. Infrastruct. Prot. 6 (2) (2013) 63–75, http://dx.doi.org/10.1016/j.ijcip.2013.05.001, URL: https://www.sciencedirect.com/science/article/pii/S1874548213000243.

[30] C.W. Badenhop, S.R. Graham, B.W. Ramsey, B.E. Mullins, L.O. Mailloux, The Z-Wave routing protocol and its security implications, Comput. Secur. 68 (2017) 112–129, http://dx.doi.org/10.1016/j.cose.2017.04.004, URL: https://www.sciencedirect.com/science/article/pii/S0167404817300792.

[31] M.B. Yassein, W. Mardini, T. Almasri, Evaluation of security regarding Z-wave wireless protocol, in: Proceedings of the Fourth International Conference on Engineering ; MIS 2018, ICEMIS '18, Association for Computing Machinery, New York, NY, USA, 2018, http://dx.doi.org/10.1145/3234698.3234730.

[32] L. Deniel, Wireshark manufacture database, 2023, https://www.wireshark.org/tools/oui-lookup.html. (Accessed 21 March 2023).

[33] S. Iftikhar, S.S. Gill, C. Song, M. Xu, M.S. Aslanpour, A.N. Toosi, J. Du, H. Wu, S. Ghosh, D. Chowdhury, M. Golec, M. Kumar, A.M. Abdelmoniem, F. Cuadrado, B. Varghese, O. Rana, S. Dustdar, S. Uhlig, AI-based fog and edge computing: A systematic review, taxonomy and future directions, Internet of Things 21 (2023) 100674, http://dx.doi.org/10.1016/j.iot.2022.100674, URL: https://www.sciencedirect.com/science/article/pii/S254266052200155X.

[34] J. Roldán, J. Boubeta-Puig, J. Luis Martínez, G. Ortiz, Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks, Expert Syst. Appl. 149 (2020) 113251, http://dx.doi.org/10.1016/j.eswa.2020.113251, URL: https://www.sciencedirect.com/science/article/pii/S0957417420300762.

[35] D.N.P. Suthishni, K.S.S. Kumar, A review on machine learning based security approaches in intrusion detection system, in: 2022 9th International Conference on Computing for Sustainable Global Development, INDIACom, 2022, pp. 341–348, http://dx.doi.org/10.23919/INDIACom54597.2022.9763261.

[36] I.H. Sarker, Machine learning: Algorithms, real-world applications and research directions, SN Comput. Sci. 2 (3) (2021) 1–21.

[37] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.

[38] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), in: Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017, pp. 1–9, URL: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[39] R. Shwartz-Ziv, A. Armon, Tabular data: Deep learning is not all you need, Inf. Fusion 81 (2022) 84–90, http://dx.doi.org/10.1016/j.inffus.2021.11.011, URL: https://www.sciencedirect.com/science/article/pii/S1566253521002360.

[40] D. Cahoolessur, B. Rajkumarsingh, Fall detection system using XGBoost and IoT, R&D J. 36 (2020) 8–18.

[41] N.S. Dhillon, A. Sutandi, M. Vishwanath, M.M. Lim, H. Cao, D. Si, A Raspberry Pi-based traumatic brain injury detection system for single-channel electroencephalogram, Sensors 21 (8) (2021) 2779.

[42] R. Kamath, M. Balachandra, S. Prabhu, Raspberry Pi as visual sensor nodes in precision agriculture: A study, Ieee Access 7 (2019) 45110–45122.

[43] M.A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, H. Janicke, Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning, IEEE Access 10 (2022) 40281–40306, http://dx.doi.org/10.1109/ACCESS.2022.3165809.

[44] S. Ruiz-Villafranca, J. Roldán-Gómez, J. Carrillo-Mondéjar, J.M.C. Gómez, J.M. Villalón, A MEC-IIoT intelligent threat detector based on machine learning boosted tree algorithms, Comput. Netw. (2023) 109868.

[45] L. Zahedi, F.G. Mohammadi, S. Rezapour, M.W. Ohland, M.H. Amini, Search algorithms for automated hyper-parameter tuning, 2021, http://dx.doi.org/10.48550/ARXIV.2104.14677, URL: https://arxiv.org/abs/2104.14677.

[46] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.

[47] Streamlit team development, Streamlit framework, 2023, https://docs.streamlit.io/. (Accessed 21 March 2023).

[48] S. Asadollahi, B. Goswami, M. Sameer, Ryu controller's scalability experiment on software defined networks, in: 2018 IEEE International Conference on Current Trends in Advanced Computing, ICCTAC, IEEE, 2018, pp. 1–5.

[49] O. Grygorash, Y. Zhou, Z. Jorgensen, Minimum spanning tree based clustering algorithms, in: 2006 18th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'06, IEEE, 2006, pp. 73–81.

[50] S. Ruiz-Villafranca, J. Carrillo-Mondéjar, J.M. Castelo Gómez, J. Roldán-Gómez, MECInOT: A multi-access edge computing and industrial internet of things emulator for the modelling and study of cybersecurity threats, J. Supercomput. (2023) http://dx.doi.org/10.1007/s11227-023-05098-2.

[51] D. Inc., Docker container, 2022, https://www.docker.com/. (Accessed 19 September 2022).

[52] C. Fiandrino, A. Pizarro, P. Mateo, C. Andrés Ramiro, N. Ludant, J. Widmer, openLEON: An end-to-end emulation platform from the edge data center to the mobile user, Comput. Commun. 148 (2019) 17–26, http://dx.doi.org/10.1016/j.comcom.2019.08.024.

[53] S.S. Gill, M. Xu, C. Ottaviani, P. Patros, R. Bahsoon, A. Shaghaghi, M. Golec, V. Stankovski, H. Wu, A. Abraham, M. Singh, H. Mehta, S.K. Ghosh, T. Baker, A.K. Parlikad, H. Lutfiyya, S.S. Kanhere, R. Sakellariou, S. Dustdar, O. Rana, I. Brandic, S. Uhlig, AI for next generation computing: Emerging trends and future directions, Internet Things 19 (2022) 100514, http://dx.doi.org/10.1016/j.iot.2022.100514, URL: https://www.sciencedirect.com/science/article/pii/S254266052200018X.