ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20240410054928

Original scientific paper
Received: 2024-04-10, Accepted: 2024-11-13

# Investigate Unsolicited Traffic on IoT Devices Using Machine Learning

Badeea Al Sukhni, Bhabendu Kumar Mohanta*, Debnath Bhattacharyya, Tai-hoon Kim*

**Abstract:** The significant growth and use of IoT devices, especially in smart homes, increase the risk of cyber-attacks compromising these devices and their services, resulting in data breaches and privacy invasion. Machine learning technology has been considered a sufficient security solution in IoT due to its effectiveness in detecting unsolicited traffic on an IoT network, ensuring the confidentiality, integrity, and availability of IoT devices, and, most importantly, protecting the users' privacy. In this research, different classification-based machine learning algorithms were leveraged to detect and classify different types of network traffic on the IoT network. At the same time, the effectiveness of these classification machine learning algorithms, Decision Tree (J48), Bayes Net, and Naive Bayes were conducted on our smart home dataset. Before the implementation of this algorithm, a man-in-the-middle attack was introduced on the IoT network while the network traffic was equally captured in the process. Overall, the algorithms successfully evaluated the captured network traffic, detected the introduced MITM attack on IoT devices, and classified the traffic into solicited and unsolicited traffic. According to the findings, the Naive Bayes algorithm outperformed the others with an accuracy of 99.95%.

**Keywords:** Arp Poisoning; Internet of Things (IoT); Machine Learning; MITM attack; Security; Smart home

## 1 INTRODUCTION

Internet of Things (IoT) refers to connecting and remotely managing the users' smart devices to each other and the Internet. Every device is embedded with storage, communication, and computing technologies [1]. Before 1990, the world had the internet, but without things. This indicates that there weren't intelligent cameras, home assistants, or even smart lights until the first IoT device, John Romkey and Simon Hackett, developed the internet toaster [2]. They used the TCP: Transmission Control Protocol/ Internet Protocol (TCP/IP) networking to connect the toaster to the internet. They also utilized the Simple Networking Management Protocol (SNMP) to control the power applied to the toast. In 1999, Kevin Ashton [3] coined the term IoT; he proposed a method to connect smart devices to the internet using primary enabling technologies such as Radio Frequency Identification (RFID) and Wireless Sensor Network (WSN) [4]. Since then, IoT has been a significant research topic in limitless areas, and the usage of IoT devices has also increased significantly. The International Data Corporation (IDC) predicts that by 2025, the data generated from the connected 55.7 billion IoT devices will be 73.1 Zettabytes (ZB), as these devices can monitor and gather information from their surroundings to perform complex tasks that benefit humanity. In other words, the IoT is the reason behind advanced manufacturing, smart homes, smart cities, automatic car tracking, modern surveillance, smart lighting, transportation, agriculture, and health care systems [5]. Many smart home products are available but unused due to various security and privacy challenges [6].

The plethora of usage of IoT devices raises the potential of cyber-attacks where the attackers compromise these devices and their services to achieve their demands. This could result in data leakage, as these devices exchange and store sensitive and private information. For instance, smartphones are used to browse the internet, pay bills, receive medical care, and sign in to various services, all of which require personal information. However, it may appear handy to use smartphones whenever and anywhere, with sensitive information stored on them, such as medical records, bank accounts, or passwords, which can be a security concern. According to a study conducted at the University of Pennsylvania, 68 percent of existing mobile devices are incapable of protecting against cyber-attacks [7]. An example of a cyber-attack that happened in 2011 was the DroidDream Malware, which infected 68 applications on the Android market. These apps were downloaded over 260K times in four days, and they used the Android Debug Bridge vulnerability on root phones and sent unauthorized premium-rate SMS messages late at night [7]. Since the network layer is the backbone for connecting smart devices to the internet, it could pose more security threats than other layers that compromise those devices. Some threats on this layer are Man-In-The-Middle attack (MITM), sniffing attack, Denial of Service (DoS), and network intrusion attacks. The most common security, privacy, and cybersecurity-related issues at this layer that were identified by [8-10] include:

- Heterogeneity: The different technologies and protocols [11] the devices use to communicate in the IoT network make security and network management more difficult to maintain.
- Scalability issues: Due to the massive number of IoT devices connected to the network, these devices can enter and leave anytime. This introduces new challenges to the IoT network, such as congestion and lack of authentication.
- Data disclosure: cyber-attackers may use social engineering techniques that may jeopardize sensitive data and result in stealing them from the network. Also, they can deploy different data recovery methods to extract critical information from the nodes. This leads to sufficient security approaches such as Machine Learning (ML), Intrusion Detection Systems (IDS), honeypots, and firewalls for preventing and detecting cyber-attacks, ensuring confidentiality, integrity, and availability of IoT devices, and finally, protecting the users' privacy invasion.

### 1.1 Research Aims and Objectives

This research aims to evaluate the effectiveness of the Decision Tree (J48), Bayes Net, and Naive Bayes algorithms in detecting MITM attacks on IoT devices and classifying the network traffic into solicited and unsolicited traffic. In this research, the Xiaomi Redmi Note 9S device was used as an example of IoT devices due to the remarkable growth of using smartphones in smart homes, as smartphones contain various kinds of sensors that store a large amount of sensitive information, enabling mobile applications to be used in a variety of IoT areas [12].

The objective is to conduct in-depth research on the Decision Tree (J48), Bayes Net, and Naive Bayes algorithms to understand them better and determine their significance in securing the Xiaomi device and differentiating between normal and abnormal network traffic. Also, to perform an MITM attack, specifically an ARP poisoning attack on the Xiaomi Redmi Note 9S device, because MITM attacks are one of the most common threats that compromise IoT devices and result in stealing sensitive information about users. Then, to capture the network traffic of the home network to which the Redmi Note 9S device is connected. Besides, to perform a critical analysis of the result produced by the Decision Tree (J48), Bayes Net, and Naive Bayes algorithms for detecting the unsolicited network traffic on the Redmi Note 9S device.

The rest of this paper is organized as follows: In Section 2, data collection, test environment, and system design with attack model are explained. Section 3 presents the implementation details. Section 4 discusses the experiment's results. The paper has been concluded, and future scopes are indicated in Section 5.

## 2 METHODOLOGY AND SYSTEM DESIGN

This section discusses the methodology of the proposed approach, which focuses on data collection techniques. The questionnaire and interview were two data collection methods employed in this research. They were used to collect opinions from home users regarding security breaches on IoT devices that result in privacy invasion. This section covered the design of the proposed system, the project test environment, which includes hardware and software requirements, and the network setup.

### 2.1 Data Collection

The first data collection stage involved interviewing home users of IoT devices to get their perspectives on privacy invasion concerns. The next stage involved developing a questionnaire based on the responses obtained from home users. These approaches have been used to determine their opinions regarding security breaches on IoT devices and to understand how they would like their IoT devices to be secured using an effective Naive Bayes Model. The rationale for using the questionnaire and interview approaches is that they are the most effective way to collect data from a large participant in a short period. They are also the most cost-effective method [13] since the data collection medium was online due to the COVID-19 pandemic.

The questionnaire comprises six questions using a close-ended approach in a multiple-choice form. The questionnaire's questions were selected and evaluated by me and IoT home users for the following reasons. This includes:

The first rationale is that we deeply considered instances where unauthorized/third-party attackers could access their home network without their consent. This is aligned with the primary objective of the entire research, which is to ensure that confidentiality is achieved in a network connection that interlinks IoT home users alongside the smart devices they use in their homes. However, after careful consideration with home users, we decided to use these questions as some had no prior knowledge of when a third party was accessing their wireless home network.

Secondly, some IoT home users fell prey to privacy invasion attacks where their secret data were exposed to attackers. For instance, one of the IoT home users we conversed with before developing the questionnaires aired their views on how their smart home camera was breached. This spurred our desire to incorporate specific questions about the privacy invasion experience for potential victims in my questionnaire.

Thirdly, after sharing their experiences, they suggested some defensive mechanisms to prevent them from the catastrophic privacy breaches they had encountered and other future attacks. This inspired us to search for significant questions about protecting their home network and IoT applications.

Overall, because some home users lack the technical know-how to utilize defensive mechanisms to secure their network from privacy concerns, they suggested automatic preventive and defensive measures, which allowed us to include this in the questionnaire.

Moreover, due to the rising number of IoT attacks in recent years and the current attacks within this global pandemic, we included a section relating to the collection of private data of IoT applications by IoT manufacturers. This is because we believe that effective and robust IoT data handling will equally mitigate the growth of IoT attacks, especially attacks dealing with privacy invasion.

Immediately after developing these electronic-based questionnaires, we directly sent them to IoT home users to get their consent on the issues to be addressed, how well they want them solved, and how it will help them stay protected and secured from privacy attacks. However, all responses on the questionnaires and feedback obtained were directly answered by only IoT home users.

### 2.2 Test Environment

This section discusses the design environment's software and hardware requirements and the network setup.

#### 2.2.1 Hardware Requirements

The devices utilized in the project's implementation are listed in the table below. The Xiaomi Redmi Note 9S was a target device for an MITM attack. The Sony laptop was used for various purposes, including capturing network traffic with the Wireshark tool, introducing the ARP poisoning

attack on the smartphone with Kali Linux, and collecting the regular traffic from this device, as well as Alexa (Amazon Echo Dot 3rd gen) and the Apple HomePod mini.

**Table 1** Devices used in the research implementation.

| Device | Notes |
|---|---|
| Redmi Note 9S | Android OS, and 128 GB RAM |
| Sony laptop | Windows 10, 16 GB RAM, and 1TB SSD |
| Amazon Echo Dot 3rd Generation | Alexa integration |
| Apple HomePod mini | Siri integration |

### 2.2.2 Software Requirements

This section explains the software that was utilized during the implementation phase.

- **Scanning Home Network Devices.** Advanced IP Scanner was used to find the devices connected to the home network and their IP and MAC addresses. Advanced IP Scanner is a free network scanning tool that retrieves the IP and MAC addresses of the devices connected to a LAN network to perform network analysis. Its user-friendly interface detects all network devices and provides remote control, including the ability to turn them on and off.
- **Capturing the Traffic.** This research used Wireshark, a joint open-source network analyzer software known as a network monitor or packet sniffer, to capture network traffic. The primary rationale for using this program is that it is a powerful tool for network traffic monitoring and troubleshooting, as it can scan any type of network, including Ethernets, Wi-Fi, Bluetooth, and even monitor mode [14]. It also captures raw bits and decodes them into a human-readable format for analysis.
- **Introducing ARP Poisoning Attack.** To implement an ARP poisoning attack on the home network, Kali Linux must be installed on Windows 10 using Oracle Virtualbox to act like a Hacking machine, as the virtual machine provides a realistic environment. After installing Kali Linux in the Virtualbox, the ARP table is needed to check the IP addresses and the associated MAC addresses of the default gateway and Kali Linux are working as expected. This can be done by entering arp –an in the command prompt on a Windows 10 machine. To implement an ARP poisoning attack on the home network, the Arpspoof tool must be installed on Kali Linux and the dsniff package, including the Arpspoof tool. The Arpspoof tool is very effective for sniffing network traffic in a home network, and it's simple to use because the coding is straightforward [15]. It also alters the flow of packets such that they pass via the attacker's machine. Besides, various attacks can be performed on the same platform because Arpspoof comes with the entire Dsniff package [15].
- **Deploying Decision Tree (J48), Bayes Net and Naive Bayes algorithms.** To evaluate the proposed solution, a Python script is needed to extract the features and convert the .pcap file to .csv. At this point, installing Python 3 on Kali Linux and the Tshark tool is required. Tshark is part of the Wireshark tool that makes the .pcap file more straightforward to understand, extracts the packets from the pcap file, and converts it to a vector space model. Besides, the Weka tool must also be installed on Windows 10. Weka is a machine learning toolkit used to implement the Decision Tree (J48), Bayes Net, and Naive Bayes algorithms to easily apply to the research dataset, as the algorithms are available as an open-source through the Weka interface. This tool provides a graphical user interface for various machine-learning algorithms for data analysis and predictive modeling [16]. Also, it allows preprocessing of a dataset, feeding it directly into machine learning algorithms, and examining the final classifier and its performance without writing any code [16].

### 2.2.3 Network Setup

As mentioned, Kali Linux runs on Windows 10 devices using Oracle Virtualbox. Virtual machines have different networking modes, such as host-only adapters, internal networks, bridged adapters, and NAT networks. A bridged adapter network mode was chosen since the Xiaomi Redmi Note 9S device and Kali Linux must be connected to the same internal network environment to create the ARP poisoning attack. That's because bridge mode allows Kali Linux and the Xiaomi Redmi Note 9S device to access each other in the network. However, NAT mode allows one-way access. Kali Linux can access them through the network, but the Xiaomi Redmi Note 9S device cannot access it. Also, they cannot access each other in the Host-only adapter and internal mode because they are not in the same network.

### 2.2.4 System Design

As discussed earlier, Decision Tree (J48), Bayes Net, and Naive Bayes algorithms were employed in this research to evaluate its effectiveness in investigating unsolicited traffic, classifying it as normal and abnormal traffic, and protecting IoT devices from privacy invasion. Also, the IoT devices used in this work were the Xiaomi Redmi Note 9S, Sony laptop, Alexa, and HomePod mini device to test how to capture the traffic. The abnormal traffic was collected from Xiaomi devices, and the normal traffic was gathered from Sony laptops, Alexa, and HomePod mini devices. The design of the proposed system combines both the flowchart and pseudocode.

#### 2.2.4.1 Flowchart

A flowchart is a diagram that is used to represent an algorithm. It shows the steps of writing the algorithm in appropriate shapes and connecting them with arrows to direct the flow of the processes. The flowchart shown in Figure 1 starts with the data collection plan for capturing the network traffic, and it ends with evaluating the performance of the Decision Tree (J48), Bayes Net, and Naive Bayes algorithms that were applied to the vector space model. After the network traffic has been collected and captured, the next two steps are utilized to extract the features and build a vector space model, which is then used to represent the network traffic in a format suitable for the machine learning algorithm. The following step deals with applying the feature selection method and reducing the extracted features, then using the three classifiers on the vector space model to evaluate the efficacy of all the extracted features or applying the three classifiers directly with the feature selection process. The final step deals with assessing the performance

of the algorithms in detecting the attack and classifying the traffic.

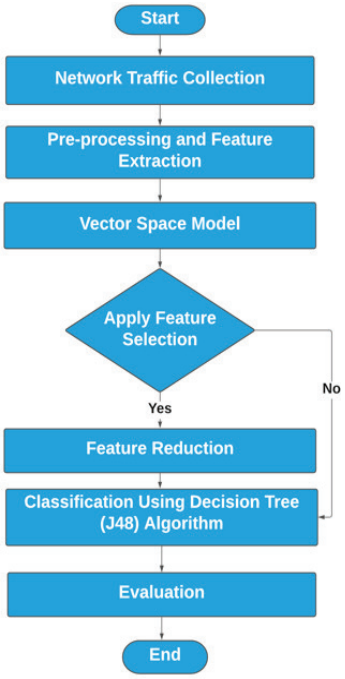


**Figure 1** Flowchart of the proposed system for investigating unsolicited traffic

- **Network Traffic Collection.** The Xiaomi Redmi Note 9S, Alexa, Sony laptop, and HomePod were connected to the home network to collect network traffic. Wireshark's packet sniffer tool was used to capture the raw data for an extended period while capturing the traffic. An ARP poisoning attack was performed on the Xiaomi device, as shown in Fig. 2.


**Figure 2** Network traffic collection

- **Pre-processing and Feature Extraction.** After collecting the home network traffic, the next step is to preprocess the data and extract the features. This step removes unnecessary features from the raw data to create patterns involving legitimate and suspicious activities [17].
- **Vector Space Model.** The following process applies the data to the vector space model. A vector space model is an n-dimensional vector of unique features that represents the data geometrically so that it fits into machine learning algorithms.
- **Feature Selection.** The feature selection process improves the effectiveness of machine learning algorithms and data classification performance [18]. Furthermore, since the dataset contains many non-essential features, the feature selection procedure can help reduce the number of extracted features by eliminating irrelevant features [18]. The feature selection process can also improve learning accuracy and speed up the modeling process [18].
- **Classification Using Decision Tree Algorithm.** The next step is to apply the open-source Decision Tree classifier using the Weka interface to the prepared dataset to detect the ARP poisoning attack on the home network and distinguish traffic between normal and abnormal behavior.
- **Evaluation.** The evaluation step calculates the proposed machine learning algorithm's accuracy in detecting attacks and classifying the traffic.

#### 2.2.4.2 Pseudocode

A pseudocode was used to evaluate an algorithm to detect unsolicited traffic on the IoT network. Pseudocode is a text-based approach that helps design and develop algorithms. It uses informal English that does not use any particular programming language to describe the steps of an algorithm in a way that anyone with basic programming knowledge can understand. Then, a suitable programming language can be used to code the algorithm. The following pseudocodes in algorithms one and two demonstrate the ARP poisoning attack and the proposed approach to detect unsolicited traffic.

The above pseudocode in algorithm 1 demonstrates the ARP poisoning attack on the Xiaomi device. The first step after connecting the target and the attacker machine on the same network is to collect the traffic for further analysis. Then, IP forwarding is enabled to route the packets through the attacker's machine, preventing the target system from losing internet connectivity. The next step is to configure ARP poisoning. This can be done by launching the first mentioned command, which is used to ARP spoof the target device and inform it that the attacker's MAC address is the router's IP address. However, the following command in the configuration is used to ARP spoof the router and inform it that the attacker's MAC address is the router's IP address. The pseudocode's final step is to browse the Xiaomi smartphone to sniff the user's personal information.

As presented in the above pseudocode in algorithm 2, the input is the captured network traffic after introducing the MITM attack on the IoT network. The expected output is classifying the traffic into normal and abnormal traffic. The first step in the procedure is to create a Python script to extract the features from the captured network traffic. Then, filter the data from the .pcap file to store the ARP, DNS, and HTTP packets on the Redmi Note 9S IP address and consider them Abnormal_Traffic. Consider the TCP on the Sony laptop, Alexa, and HomePod IP address as Normal_Traffic.

After pre-processing the data, the next step is creating the vector space model and writing the string "label" and 33 features to the header of the .csv file. After preparing the headers of the .csv file, the next step is to filter all the .pcap file packets and store them into two .pcap files with the label name, as this step will reduce the raw data into relevant

packets. The next step is writing the content of these files into the .csv file to apply it to the Decision Tree (J48), Bayes Net, and Naive Bayes algorithms to detect the unsolicited traffic and classify the traffic.

## 3 IMPLEMENTATION

This section discusses the implementation of the proposed solution on the Xiaomi Redmi Note 9S device. The Xiaomi Redmi Note 9S smartphone, Sony laptop, Alexa, and Apple HomePod device were used to demonstrate a real-world IoT network. The Xiaomi smartphone was utilized as a target device for an MITM attack, specifically ARP poisoning, to capture the abnormal traffic. The average traffic was collected using a Sony laptop, Alexa, and the HomePod. The test environment, including the hardware, software, and network configurations, was also handled. Following the collection of network traffic, a Python script was created to extract features and build a vector space model. The process ended by preparing a dataset with 33 attributes and storing 13980 instances; 12240 instances are normal traffic, and 1740 instances are abnormal traffic. The implementation was concluded using the Weka interface to apply the three classifiers to the dataset.

### 3.1 Network Traffic Collection

Collecting sufficient amounts of network traffic in both standard and abnormal states is needed to distinguish the IoT network between normal and abnormal. We followed an approach proposed by [19] to collect the usual traffic. Using the Wireshark tool, we captured the regular traffic from seven IoT devices installed in a private home network for five days. In this research, we connected a Sony laptop, Alexa, and HomePod to the home router and captured the traffic for an extended period using Wireshark. Then, we connected the Xiaomi Redmi Note 9S device to collect the abnormal traffic to the network. We also used Virtualbox to launch Kali Linux and selected the bridge network mode. We also used the Advanced IP Scanner tool to find the IP and MAC addresses of the devices connected to the home network. Tab. 2 lists the IP and MAC addresses of each device.

**Table 2** The IP and MAC addresses of the devices connected to the home network

| Name | IP | MAC address |
|---|---|---|
| Home network router | 192.168.8.1 | 94:E9:EE:8D:81:91 |
| RedmiNote9S | 192.168.8.100 | 18:87:40:EF:19:B1 |
| Kali Linux | 192.168.8.157 | 08:00:27:5C:65:26 |
| Sony laptop | 192.168.8.144 | 00:23:14:52:66:58 |
| HomePod | 192.168.8.152 | 4C:20:B8:DB:B2:8C |
| Alexa | 192.168.8.111 | 08:84:9D:D5:C7:76 |

### 3.2 Pre-processing and Feature Extraction

Before pre-processing the captured PCAP file, we created a Python script to filter the collected traffic to have only the TCP, DNS, HTTP, and ARP packets. The TCP protocol has been chosen to represent the Normal traffic because this traffic has been collected from the Sony laptop's IP address, Alexa's IP address, and HomePod's IP address. On the other hand, the ARP, DNS, and HTTP packets were chosen to represent the Abnormal traffic, as security breaches and data leakage can occur if one or more of these protocols are successfully exploited [17]. Because the ARP protocol maps logical (IP) addresses to physical (MAC) addresses and launching an ARP poisoning attack allows the capture of the DNS packets that were requested from the user as the DNS helps internet users access their requested websites using domain names because the role of the DNS is to translate domain names into IPs [17]. Also, The HTTP protocol's purpose is to exchange hypertext, which is a textual format with hyperlinks [17]. From here, two classes were used in the research: Normal_Traffic and Abnormal_Traffic, based on the IP address and the protocol.

To keep it simple and specific to the problem, we focused on extracting 33 features from the ARP, DNS, HTTP, IP, and TCP headers to represent the raw data in the vector space model.

Because the .pcap file is vast and contains a lot of packets and data, as the list of the required IPs and protocols was discussed, we followed the [20] approach to filtering the .pcap file into two individual pcap files (with the same names as the selected labels), each one containing the data of a particular class. This process helps convert the .pcap to a CSV file with the extracted features quickly and easily.

**Algorithm 1: Introducing ARP poisoning attack**

**Input:** Connecting Xiaomi Redmi Note 9S and Kali Linux to the same network
**Output:** Introducing ARP poisoning attack on Xiaomi Redmi Note 9S
**Begin:**
1. Capture the traffic.
2. Configure the Kali Linux machine for IP forwarding.
   echo 1 > /proc/sys/net/ipv4/ip_forward
3. Configure ARP poisoning
   arp spoof –i eth0 –t target machine IP address –r router IP address
   arp spoof –i eth0 –t router IP address -r target machine IP address
4. Browse Xiaomi Redmi Note 9S

**End**

**Algorithm 2: Detecting the unsolicited traffic**

**Input:** The captured network traffic
**Output:** Classifying the traffic into normal and abnormal traffic
**Begin:**
1. Create a Python script to extract the features.
2. Filter the data from the .pcap file.
3. if (arp.dst.proto_ipv4 ==Redmi Note 9S IP address) || (dns && (ip.dst==Redmi Note 9S IP address)) || (http && (ip.dst== Redmi Note 9S IP address)
4. [Abnormal_Traffic]= these protocols data.
5. if (TCP && (ip.src==Sony laptop IP address) || (ip.src==Alexa IP address) || (ip.src==Apple HomePod IP address))
6. [Normal_Traffic] = this protocol data.
7. Create a vector space model file to store the features and labels.

8. Write the string "label" and 33 features to the header of the .csv file.
9. Filter all the .pcap file packets and store them into two .pcap
files with the label name.
10. Write the content of these files in the .csv file.
11. Use the Decision Tree J48 algorithm
12. Calculate the accuracy of the Decision Tree J48 model

**End**

### 3.3 Vector Space Model

As mentioned earlier, a vector space model is necessary to represent the data geometrically so that it can easily fit into machine learning algorithms. Eq. (1) below shows how to define the vector space model; this equation is written by [17].

$$I_{1:N} = \begin{bmatrix} f_{11} & f_{12} & \cdots \\ f_{21} & f_{22} & f_{ij} \end{bmatrix} \quad Y_{1:N} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}. \tag{1}$$

From Eq. (1), $I$ is network traffic observations, $Y$ is the classes ($c$), which are Normal_Traffic and Abnormal_Traffic, N is the number of observations, and f is the features, which are the fields of TCP, IP, and DNS, HTTP, and ARP headers to write the Python script to extract the features from the Pcap file and convert it to a CSV file, which is aligned with the proposed system.

It was necessary to import the required libraries into the script to extract helpful information from files and directory names, such as os and glob; os and glob libraries are used to access files and directories. Then, a dictionary called ip_filter was defined to filter the data from the .pcap file. Abnormal_Traffic and Normal_Traffic are the keys of the filtering string. Also, they represent the classes of the dataset. (arp. dst.proto_ipv4 == 192.168.8.100) || (dns (ip.dst==192.168.8.100)) || (http (ip.dst==192.168.8.100)) means to filter the ARP, DNS, and HTTP packets from 192.168.1.100, which is the IP address of Redmi Note 9s device and consider them as the abnormal traffic. On the other hand, (TCP (ip.src==192.168.8.144) || (ip.src==192.168.8.111) || (ip.src==192.168.8.152)) means to filter the TCP packets from 192.168.1. 144 (the IP address of the Sony laptop), 192.168.1. 111 (which is Alexa's IP address) and 192.168.1. 152 (the IP address of the HomePod) represents the usual traffic.

The next step is to create a CSV file to write the string "label" and 33 features to the header of the .csv file. Next, filter all the .pcap file packets and store them into two .pcap files with the label name. To do that, glob. The glob function from the glob library was used to process the pcap file in the NetworkTraffic folder. Also, the T-shark was used to write the packets matching the particular filter, as these filtered pcap files will be stored in the Filtered_NetworkTraffic folder. The next step is to write the content of new pcap files into the CSV file. Again glob. Glob function was used to process the pcap files in the Filtered_NetworkTraffic folder. Also, the T-shark command is used to extract features from the new pcap files. After this process, the os. The open function was used to execute the T-shark command and save the result to the features variable. This results in preparing the dataset to be applied to the three classifiers and storing 13980 instances (12240 instances as regular traffic and 1740 instances as abnormal traffic) with 33 attributes in addition to the "Label" header in the CSV file.

## 4 RESULTS ANALYSIS

This section analyzes the questionnaire results and evaluates the performance of the Decision Tree algorithm in detecting unsolicited traffic and classifying it using the Weka interface.

### 4.1 Questionnaire Results

The questionnaire consists of six questions, and we received responses from 100 home users who completed it. It begins with how they would know if a third party obtained unauthorized access to their home network. 64% said they would get help from a network technician, 32% said they wouldn't know, and 4% said they would find out from a friend. While 81.8% had experienced a data breach or had their network attacked. 85.9% of them realized their IoT network had been hacked with the help of a technical expert, 12.9% of them from a friend, and 1.2% from their parents. Besides, 84% of them want their IoT network to be protected from cyber-attacks via a system that automatically detects attacks and classifies them as normal or abnormal, 15% want a system that detects the attack and shuts down the home network, and 1% want a system that detects the attack. Finally, 59% said IoT device manufacturers should conform to stricter privacy regulations. As a result of this questionnaire, the requirements of the home users were better understood and aligned with the proposed solution.

### 4.2 Classification Accuracy

The Weka interface was used for the classification process. Weka provides several services, such as data pre-processing, classification, clustering, association, visualization, and feature selection [18]. As stated earlier, WEKA also provides various open-source algorithms developed using Java programming to conduct data mining processes [18]. These algorithms can be used directly on the dataset or developed and changed using Java code.

In this research, we utilized the open-source Decision Tree (J48), Bayes Net, and Naive Bayes algorithms to evaluate their effectiveness in detecting and classifying unsolicited traffic that could compromise network confidentiality. The classification process has two stages: classification before and after feature selection.

#### 4.2.1 Classification before Feature Selection

Before applying feature selection to the dataset, all 33 features were used to evaluate the accuracy of the three classifiers. Four metrics, accuracy, precision, recall, and F1 (F-measure), were used to evaluate the performance. A confusion matrix was used to introduce them, as it shows how many elements from a class are correctly and incorrectly classified. Tab. 3 demonstrates the confusion matrix.

**Table 3** Confusion matrix

| | Classified Positive | Classified Negative |
|---|---|---|
| Actual Positive | TP | FN |
| Actual Negative | FP | TN |

Where [21]:

- TP: the true positive is the actual average records correctly classified as benign behaviors.
- TN: the true negative indicates that abnormal records are correctly classified as attacks.
- FP: the false positive indicates that attack behaviors are classified incorrectly as benign.
- FN: the false negative indicates that benign records are incorrectly classified as attacks.

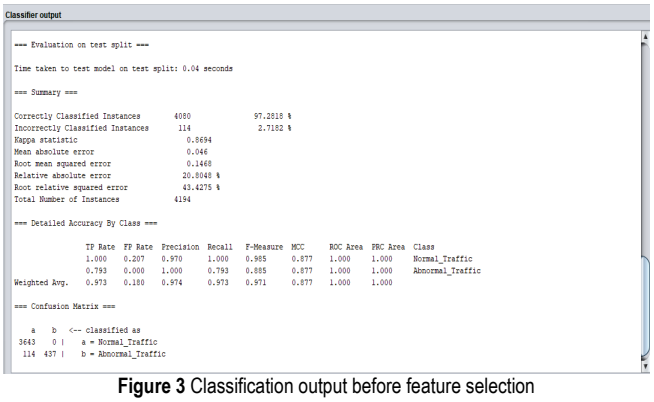


**Figure 3** Classification output before feature selection

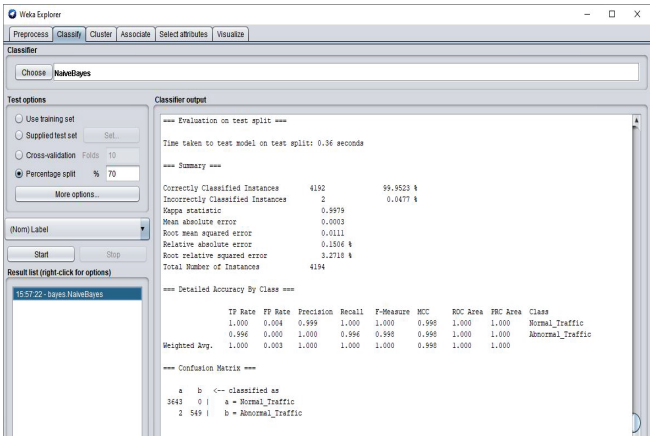


**Figure 4** Naive Bayes result

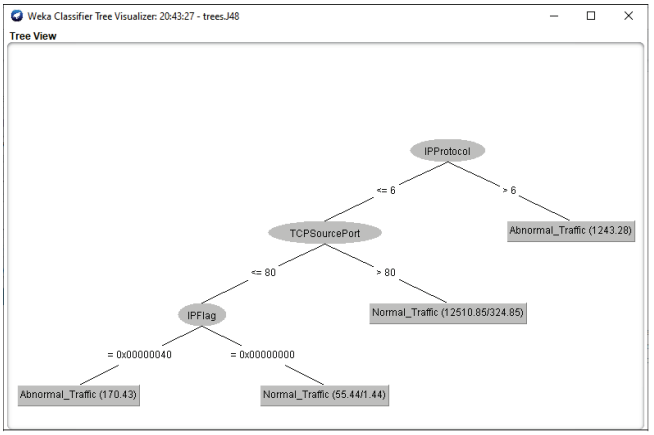


**Figure 5** Visualization of the result of J48

Fig. 3 shows the classification results before feature selection. From the confusion matrix, it can be concluded that the total correctly detected traffic was 4080, with 3643 normal traffic and 437 abnormal traffic. However, 114 regular traffic was incorrectly detected as abnormal. Fig. 4 shows the result of the Naive Bayes algorithm using Weka.

Fig. 5 demonstrates the visualization of J48 results. It shows that the tree was 7 feet tall and had four leaves.

### 4.2.2 Classification after Feature Selection

In this research, the attribute evaluator, which is InfoGainAttributeEval, was employed with the ranker search method in the Weka interface. The InfoGainAttributeEval evaluates how each extracted feature decreases the input dataset's overall entropy [20]. A relevant attribute with the most significant information and hence reduces entropy the most could be prioritized above another using this strategy [20]. This technique was used to rank each attribute from the most significant to the least important.

In addition to the attribute evaluator and its search method, the attribute selection mode was also selected. In this research, a 10-fold cross-validation method was chosen for the dataset, as the 10-fold cross-validation process can effectively determine the value of the attribute subset. The 10-fold means the process was repeated 10 times, as the nine folds were used as training data, and one fold was used as testing data.

The results of applying feature selection showed that 17 of the features were irrelevant, and 16 features relating to IP and TCP headers were relevant to the problem. The IP checksum had the highest rank. Fig. 6 presents the attribute selection output.

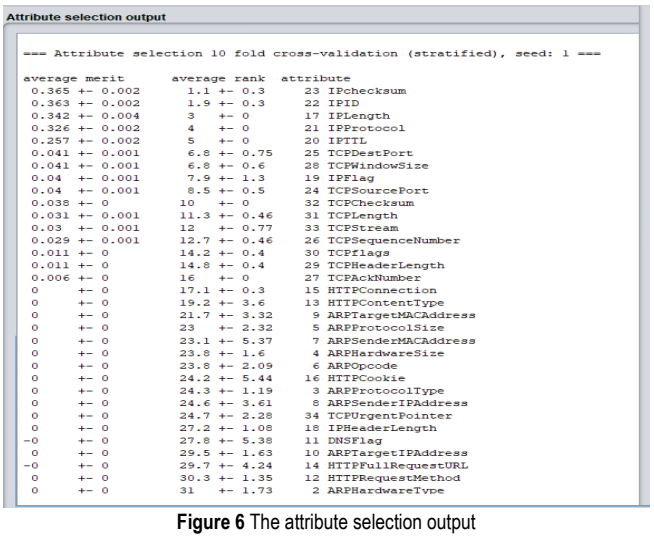


**Figure 6** The attribute selection output

**Table 4** The evaluation results of three algorithms for detecting ARP poisoning attack

| Classifier | Accuracy (%) | Precision (%) | Recall (%) | F-measure (%) | FPR (%) |
|---|---|---|---|---|---|
| J48 | 97.2818 | 97.4 | 97.3 | 97.1 | 1.8 |
| Bayes Net | 99.5708 | 99.6 | 99.6 | 99.6 | 1 |
| Naive Bayes | 99.05 | 100 | 100 | 100 | 0.03 |

After applying the 16 relevant features, the results obtained before the feature selection were not changed after the feature selection. J48 Decision Tree demonstrated an accuracy detection rate of 97.2818%, a precision of 97.4%, a recall of 97.3%, an F-measure (F1) of 97.1%, and a low false positive rate of 1.8% for all 33 features. Bayes Net demonstrated an accuracy detection rate of 99.5708%, a precision of 99.6 %, a recall of 99.6%, an F-measure (F1) of 99.6%, and a low false positive rate of 1.0% for all 33 features. Naive Bayes demonstrated an accuracy detection rate of 99.95 %, a precision of 100 %, a recall of 100 %, an F-measure (F1) of 100 %, as well as a low false positive rate of 0.03% for all 33 features shown in Tab. 4. These results indicate that all three algorithms effectively identify anomalous behaviors on the IoT network and differentiate regular traffic from attack traffic with high accuracy, precision, recall, and F-measure. The naive Bayes algorithm achieved the highest accuracy, which is 99.95%. Also, algorithms achieved gains in confidentiality, integrity, and availability when applied to the Redmi Note 9S device.

## 5 CONCLUSION

This research discussed machine learning and what solutions it can provide to industry, specifically IoT devices, to maintain information confidentiality, integrity, and availability while protecting users' privacy. Chosen machine learning algorithms J48, Bayes Net, and Naive Bayes were proposed solutions to investigate the unsolicited traffic and classify the traffic as normal and abnormal to safeguard IoT devices. The implementation was evaluated to see how effectively the three algorithms detect unsolicited traffic and classify the traffic. The Naive Bayes achieved a high accuracy detection rate of 99.95%, a precision of 100%, a recall of 100%, an F-measure (F1) of 100%, and a low false positive rate of 0.03%. Also, the Naive Bayes showed that it performs variable feature selection implicitly. To conclude, the Decision Tree (J48), Bayes Net, and Naive Bayes were practical algorithms for securing IoT devices and preventing privacy invasion. Naive Bayes is the best among the three used algorithms in IoT home networks.

## 6 REFERENCES


[1] Navani, D., Jain, S., & Nehra, M. S. (2017, December). The Internet of Things (IoT): A study of architectural elements. In *IEEE 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS2017)*, 473-478. https://doi.org/10.1109/SITIS.2017.83

[2] Romkey, J. (2016). Toast of the IoT: the 1990 interop internet toaster. *IEEE Consumer Electronics Magazine*, *6*(1), 116-119. https://doi.org/10.1109/MCE.2016.2614740

[3] Ashton, K. (2009). That 'Internet of Things' thing. *RFID Journal*, *22*(7), 97-114.

[4] Tewari, A., & Gupta, B. B. (2020). Security, privacy, and trust of different layers in the Internet-of-Things (IoT) framework. *Future generation computer systems*, *108*, 909-920. https://doi.org/10.1016/j.future.2018.04.027

[5] Mohanta, B. K., Jena, D., Satapathy, U., & Patnaik, S. (2020). Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet of Things*, *11*, 100227. https://doi.org/10.1016/j.iot.2020.100227

[6] Mahmoudi, R., Roozi, S., Saghiri, A. M., & Mahmoudi, A. (2020). Extracting strategies for improving internet-of-things-based home industries in Iran: A strengths, weaknesses, opportunities, and threats analysis. *IEEE Transactions on Engineering Management*, *68*(2), 586-598. https://doi.org/10.1109/TEM.2020.2991859

[7] Varol, N., Aydogan, A. F., & Varol, A. (2017, April). Cyber attacks targeting Android cellphones. In *IEEE 5th International Symposium on Digital Forensic and Security (ISDFS2017)*, 1-5. https://doi.org/10.1109/ISDFS.2017.7916511

[8] Mohanta, B. K., Jena, D., Ramasubbareddy, S., Daneshmand, M., & Gandomi, A. H. (2020). Addressing security and privacy issues of IoT using blockchain technology. *IEEE Internet of Things Journal*, *8*(2), 881-888. https://doi.org/10.1109/JIOT.2020.3008906

[9] Carayannis, E. G., Grigoroudis, E., Rehman, S. S., & Samarakoon, N. (2019). Ambidextrous cybersecurity: The seven pillars (7Ps) of cyber resilience. *IEEE transactions on engineering management*, *68*(1), 223-234. https://doi.org/10.1109/TEM.2019.2909909

[10] Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in IoT security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, *22*(3), 1686-1721. https://doi.org/10.1109/COMST.2020.2986444

[11] Tournier, J., Lesueur, F., Le Mouël, F., Guyon, L., & Ben-Hassine, H. (2021). A survey of IoT protocols and their security issues through the lens of a generic IoT stack. *Internet of Things*, *16*, 100264. https://doi.org/10.1016/j.iot.2020.100264

[12] Talari, S., Shafie-Khah, M., Siano, P., Loia, V., Tommasetti, A., & Catalão, J. P. (2017). A review of smart cities based on the Internet of Things concept. *Energies*, *10*(4), 421. https://doi.org/10.3390/en10040421

[13] Garba, A. A., Siraj, M. M., Othman, S. H., & Musa, M. A. (2020). A study on cybersecurity awareness among students in Yobe State University, Nigeria: A quantitative approach. *Int. J. Emerg. Technol*, *11*(5), 41-49.

[14] Goyal, P., & Goyal, A. (2017, September). Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark. In *IEEE 9th International Conference on Computational Intelligence and Communication Networks (CICN2017)*, 77-81. https://doi.org/10.1109/CICN.2017.8319360

[15] Nam, S. Y., Djuraev, S., & Park, M. (2013). Collaborative approach to mitigating ARP poisoning-based Man-in-the-Middle attacks. *Computer Networks*, *57*(18), 3866-3884. https://doi.org/10.1016/j.comnet.2013.09.011

[16] Al Sukhni, B., Mohanta, B. K., Dehury, M. K., & Tripathy, A. K. (2023, July). A Novel Approach for Detecting and Preventing Security attacks using Machine Learning in IoT. In *IEEE 14th International Conference on Computing Communication and Networking Technologies (ICCCNT2023)*. 1-6. https://doi.org/10.1109/ICCCNT56998.2023.10307883

[17] Moustafa, N., Turnbull, B., & Choo, K. K. R. (2018). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet of Things Journal*, *6*(3), 4815-4830. https://doi.org/10.1109/JIOT.2018.2871719

[18] Al Sukhni, B., Dave, J. M., Manna, S. K., & Zhang, L. (2022, December). Investigating the security issues of multi-layer IoT attacks using machine learning techniques. In *IEEE Human-Centered Cognitive Systems (HCCS2022)*, 1-9. https://doi.org/10.1109/HCCS55241.2022.10090400

[19] Salman, O., Elhajj, I. H., Chehab, A., & Kayssi, A. (2022). A machine learning based framework for IoT device identification and abnormal traffic detection. *Transactions on Emerging Telecommunications Technologies*, *33*(3), e3743. https://doi.org/10.1002/ett.3743

[20] Cabrera, A., & Calix, R. A. (2016, October). On the anatomy of the dynamic behavior of polymorphic viruses. In *IEEE International Conference on Collaboration Technologies and Systems (CTS2016)*. 424-429. https://doi.org/10.1109/CTS.2016.0081

[21] Al Sukhni, B., Manna, S. K., Dave, J. M., & Zhang, L. (2022, December). Machine learning-based solutions for securing IoT systems against multilayer attacks. In *International Conference on Communication, Networks and Computing* (pp. 140-153). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-43140-1_13

[22] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017). Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. *IEEE Access*, *5*, 18042-18050. https://doi.org/10.1109/ACCESS.2017.2747560

**Author's contacts:**

**Badeea Al Sukhni**
Department of Cyber Security, University of Westminster,
309 Regent Street, W1B 2HW London, United Kingdom
E-mail: badeaa.alsukhni94@gmail.com

**Bhabendu Kumar Mohanta**
(Corresponding author)
College of Information Technology, United Arab Emirates University,
Al Ain, P.O. Box 15551, United Arab Emirates
E-mail: bhabendu@uaeu.ac.ae

**Debnath Bhattacharyya**
Department of Information Technology,
Aditya Institute of Technology and Management,
Tekkali, Andhra Pradesh, 532201, India
E-mail: debnathb@gmail.com

**Tai-hoon Kim**
(Corresponding author)
School of Electrical and Computer Engineering,
Yeosu Campus, Chonnam National University,
50, Daehak-ro, Yeosu-si, Jeollanam-do, 59626, Republic of Korea
E-mail: taihoonn@chonnam.ac.kr