Article

SDN-Enabled FiWi-IoT Smart Environment Network Traffic Classification Using Supervised ML Models

Elaiyasuriyan Ganesan ¹ , I-Shyan Hwang ^{1,*}, Andrew Tanny Liem ² and Mohammad Syuhaimi Ab-Rahman ³

¹ Department of Computer Science and Engineering, Yuan Ze University, Taoyuan 32003, Taiwan; s1069107@mail.yzu.edu.tw

² Department of Computer Science, Universitas Klabat Manado, North Sulawesi 95371, Indonesia; andrew.heriyana@unklab.ac.id

³ Department of Electrical, Electronics, and System Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia; syuhaimi@ukm.edu.my

* Correspondence: ishwang@saturn.yzu.edu.tw

Abstract: Due to the rapid growth of the Internet of Things (IoT), applications such as the Augmented Reality (AR)/Virtual Reality (VR), higher resolution media stream, automatic vehicle driving, the smart environment and intelligent e-health applications, increasing demands for high data rates, high bandwidth, low latency, and the quality of services are increasing every day (QoS). The management of network resources for IoT service provisioning is a major issue in modern communication. A possible solution to this issue is the use of the integrated fiber-wireless (FiWi) access network. In addition, dynamic and efficient network configurations can be achieved through software-defined networking (SDN), an innovative and programmable networking architecture enabling machine learning (ML) to automate networks. This paper, we propose a machine learning supervised network traffic classification scheduling model in SDN enhanced-FiWi-IoT that can intelligently learn and guarantee traffic based on its QoS requirements (QoS-Mapping). We capture the different IoT and non-IoT device network traffic trace files based on the traffic flow and analyze the traffic traces to extract statistical attributes (port source and destination, IP address, etc.). We develop a robust IoT device classification process module framework, using these network-level attributes to classify IoT and non-IoT devices. We tested the proposed classification process module in 21 IoT/Non-IoT devices with different ML algorithms and the results showed that classification can achieve a Random Forest classifier with 99% accuracy as compared to other techniques.

Keywords: SDN-FiWi-IoT; QoS-mapping; network traffic classification; machine learning

Citation: Ganesan, E.; Hwang, I-S.; Liem, A.T.; Ab-Rahman, M.S. SDN-Enabled FiWi-IoT Smart Environment Network Traffic Classification Using Supervised ML Models. *Photonics* **2021**, *8*, 201. <https://doi.org/10.3390/photonics8060201>

Received: 19 April 2021

Accepted: 2 June 2021

Published: 4 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The world has seen the incredible growth in the Internet as a global communication infrastructure in recent decades. The wired and wireless Internet revolutionized the telecommunications paradigm to enable communication with anyone, “anytime.” The emerging Internet of Things (IoT) is creating another paradigm, in which “anything” can be accessed and/or controlled remotely, allowing for a more direct coordination between the physical world and machines-based systems [1]. IoT refers to billions of Internet-connected physical devices worldwide, collecting and sharing data. There will be a rapid increase in the number of different pieces of IoT equipment, as sensors and actuators are widely used in many applications, such as cyber security, automation, metering, health care, utilities and consumer electronics. Gartner predicts that, by 2022, the typical household could contain more than 500 smart devices [2]. Further, the world’s IoT devices are expected to reach 18 billion by 2022 [3]. The rapid development of IoT devices presents a new problem of network allocation on the current Internet, especially the “last mile” access network, which has long been recognized as a major bottleneck in delivering high-speed internet service.

The number of network users and the traffic load due to these users are gradually increasing. Fiber-wireless (FiWi) converged access networks are gaining more attention for their ability to handle this level of traffic in the ubiquitous community. The FiWi integrates the access network connects of the passive optical network (PON) and wireless local area network (WLAN) and takes advantage of both optical and wireless access [4]. Although optical networks can provide a wide bandwidth, it is not possible to deploy these networks everywhere because of their reliance on physical infrastructure and the associated deployment costs. By contrast, wireless networks are more flexible and can provide service over larger areas than optical networks. However, in wireless networks, the coexistence of different types of traffic requires that the communication characteristics and needs of the various devices be further differentiated. The critical problem is coexistence with traditional human-to-human (H2H) traffic, which is a triple-play operation, and emerging machine-to-machine (M2M) traffic, which needs differentiated services such as expedited forwarding (EF), assured forwarding (AF), and best-effort (BE). It is critical for other coexisting traffic coordinates' ability to assist such peers to guarantee that the traffic is not affected. In this scenario, designing resource management programs can be challenging. With its ability to combine the reliability, ubiquity, and mobility of wireless access networks with the high-capacity support of optical access networks (OANs), the FiWi access network has sparked significant research interest [5]. Thus, from an access network architecture perspective, FiWi is an excellent choice for providing communication links to IoT services and for further classifying and mapping the optical network and wireless network quality of service class identifiers (QoS-CIs) based on the IoT traffic types. The FiWi access network has to meet a wide range of service requirements for QoS, security, maintenance and operational improvements, and to empower the network to allocate its resources efficiently with the multitude of IoT applications [6].

Smart environments powered by the internet of things offer a wide range of services to benefit society, including smart homes, smart buildings, smart cities, smart grids, and more. However, the proliferation of IoT creates a critical problem. Network operators in smart environments have difficulty determining which IoT devices are connected to their network and finding out what each device is doing by default, as devices are usually distributed across different sectors [7]. Internet service providers (ISPs) and network operators are often interested in determining the amount of traffic, network performance, and devices connected by their networks to improve security [8]. Moreover, Internet traffic classification is valuable, particularly for the smart environment that occupies many IoT devices. Therefore, network traffic classification is a noteworthy topic in machine learning approaches. Initially, port-based approaches are used to classify network traffic based on ports by employing sophisticated statistical and behavioral approaches that thoroughly analyze network traffic using machine and deep learning approaches [9]. Traffic classification is an important part of large-scale data analysis and plays an important role in ensuring network security and defending against traffic attacks. Whenever a new device connects to the network, it must quickly be managed and protected using the associated security mechanism or QoS policy. Therefore, a major challenge is to differentiating IoT/non-IoT devices in minutes. Unfortunately, there is no clear indication that a device on a network is an IoT. Hence, we use a machine learning approach, where we trained classifiers on the seen dataset of labeled IoT/non-IoT devices, and then analyze the accuracy of our classifiers on an unseen dataset of devices. Consequently, deploying traffic classification in more intelligence networks is becoming more difficult to achieve, because learning to perform control beyond the local domain, from nodes with only a partial view of the entire system, is extremely complex. The recent advancements in SDN will ease the complexity of learning. Figure 1 depicts the roadmap of the traffic classification direction of the new paradigm architecture of the networks. The major advantages of ML-SDN traffic classification compared with ATM-, IP-, and MPLS-based TCs are: more precisely, SDN enables (1) centralized visibility, which includes global network information (e.g., network resource limits or dynamically evolving network status) and worldwide

application data (e.g., QoS requirements); (2) programmability without the need to manage individual infrastructure elements, i.e., it is possible to proactively program OF switches on the data plane; (3) openness, in which data plane components (i.e., OpenFlow switch-es) communicate with the controller via a single interface for data plane programming and network information gathering, regardless of vendor; (4) various flow table pipelines in OF switches will increase the versatility and effectiveness of flow management. Furthermore, network traffic classification can help improve the performance and QoS of IoT devices.

Figure 1. Depicts the roadmap of traffic classification concepts from past to future.

The major progress in SDN and ML methods has created a new network management era. By utilizing a global view, SDN's centralized feature is used to manage the network. Thereby, the performance of the FiWi-IoT network is improved by optimizing bandwidth usage, load balancing, and minimizing latency. SDN's abilities make the application of machine learning techniques easier [10]. Recent advances in computing technologies (such as Graphics Processing Unit (GPU) and Tensor Processing Unit (TPU)) provide us with a good opportunity to contribute machine learning techniques. The SDN Controller has quite a global view of the network and is capable of collecting numerous network traffic data, enabling machine learning algorithms to be applied [11]. This new concept combines network intelligence and network programming to create an autonomous high-performance network that will expand 5G capabilities. In recent decades, artificial intelligence (AI) and ML concepts have been developed for a variety of applications using a variety of approaches [12,13]. Further, the AI/ML approaches are based on recent statistics. Integrating such tools into the networking industry could enable service providers to set up self-configuring, self-healing, and self-enhancing networking. This type of network can be named a knowledge-defined network. Therefore, multichannel IoT devices must be automatically classified to provide reliability, security, and improved QoS for upstream applications. However, IoT devices traffic classification plays an important role in ensuring network security and protecting against traffic attacks as an important part of massive data analysis. Furthermore, ensuring network security and protecting against traffic attacks is the main part of massive data interpretation [14].

In this paper, we aim to propose a new, SDN-enhanced Fi-Wi-IoT access network framework with the coexistence of regular service and IoT service. We aim to observe IoT device actions on the network by combining supervised SD-FiWi telemetry and machine learning methods on the network, using a combination of SD-FiWi-IoT telemetry and machine learning methods. We also address the device identification issue by building a robust framework which classifies every IoT device separately with good accuracy, utilizing statistical attributes obtained from network data characteristics. Our focus is on accurately detecting devices and tracking their dynamic behaviors based on traffic flow patterns.

This paper's major contributions are as follows:

- (1) The novelty of this paper focuses on the proposed SDN-Based FiWi architecture, IoT traffic QoS management and IoT traffic classification mechanism using ML supervised

models. This article addresses the possibilities and possible challenges of developing and implementing IoT traffic classification mechanisms in a fiber wireless smart environment to support internet service providers' (ISP) network performance. The intelligent SDN controller, in conjunction with the fiber-wireless network and machine learning, enables the combination of the benefits of programmable flow-based telemetry and modular data-driven models for the management of IoT devices based on their network operation and defence against cyberattacks;

- (2) An integrated, SDN-based, Fiber wireless network access scheme is proposed and the primary operational components are described. Further, the EPON and WLAN QoS mapping is proposed;
- (3) Using the global view of SDN and the need for traffic flows, an optimized scheme is built for the multipath transmission of IoT applications;
- (4) We implemented the proposed systems and demonstrated the performance of our classifier using 21 IoT and non-IoT devices, representing different types of device.
- (5) We propose an enhanced framework to identify IoT device specifications, in which we devise a method for extracting invariant dependencies along with all devices and deriving features from them;
- (6) Finally, we evaluate our methods on the real-time IoT dataset. Our proposed model might achieve satisfactory accuracy with a small training set in classifying new IoT and Non-IoT devices. Finally, we discuss the achieved results and compares the performance with other classifiers.

The paper is organized as follows: Section 2 outlines relevant prior work. Section 3 discusses our proposed system and its operation. Section 4 outlines network traffic classification techniques. Section 5 outlines the proposed machine learning process module. Section 6 describes performance evaluation. Finally, Section 7 presents the conclusion and future work.

2. Related Work

In this section, we give a brief overview of related work that is end-to-end FiWi access network communication for resource provision of IoT and tactile internet (TI), intelligently manage network traffic flow in SDN, device classification in IoT and network traffic analysis. To maximize resource utilization and minimize energy consumption in IoT devices, numerous resource allocation schemes such as joint bandwidth allocation have been proposed in [5]. In 2019, Y. Liu et al. proposed the bandwidth provision problem in the FiWi network, supporting the IoT service based on the machine learning prediction method [4]. An integrated heterogeneous networking system for cloud computing and the virtualization of FiWi access networks is proposed in [15]. In addition, machine learning techniques are applied to the fiber wireless network to improve system performance [16]. The SDN controller provides a centralized view of all network traffic. SDN architecture deployed in an enterprise network that collects traffic data via the Open-Flow protocol and classifies it using machine learning techniques [17,18]. The approach proposed in [19] uses fingerprinting for authentication and identification purposes by training an ML method based on network traffic features to identify similar device types. An automated classification scheme was also created for system Identifier (SysID) characteristics [20]. This work focuses on examining this IoT device identification scenario using statistical attributes including activity cycles, port numbers, signaling patterns, and cipher suites to classify IoT traffic and devices in smart cities. The final test results showed that a 95% classification accuracy rate was reached [21]. A major challenge is proposed for the robustness of the unknown method of traffic classification [22]. The author [23] uses CNN and RNN models in a profound learning combination to identify flow forms, such as HTTP, SMTP, Telnet, QUIC, Office365, and YouTube, with six features, specifically source/destination port number, payload length, TCP window size, interarrival time and directions from the first 20 flow packets. Another method employs autonomous IoT system classification using a combination of textual and flow features used for classification [24,25]. From the

perspective of network security, ML methods are used to classify IoT devices with the aim of determining if a system is on a whitelist of devices that are authorized to link to the network. In [26], a deep learning approach of LSTM-CNN cascade-based time series classification method is proposed to identify the seen and unseen devices.

3. Proposed Software-Defined-FiWi-IoT System Architecture and Operation

This section explains the proposed SD-FiWi-IoT architecture, such as enhanced optical network unit access point (ONU-AP) and optical line termination (OLT) architectures, and the system operation based on the SDN controller. Our proposed SD-FiWi-IoT system architecture is shown in Figure 2 and consists of an SD-OLT, in which the standard OLT is improved by adding an integrated OpenFlow-based SDN controller to centralize the FiWi system. As illustrated in the figure, an SDN controller can communicate with clients (service applications) via a northbound API (NB-API). The SDN management framework will aggregate all input from client applications and forward it to the SDN controller via the NB-API. A southbound API allows the controller to communicate with the OLT (SB-API). The underlying physical network is the back-end PON and the front-end ONU integrated Wi-Fi (wireless fidelity). Routers serve as connection points for end users of both traditional and IoT services such as M2M and H2M in wireless mesh networks (WMN).

Figure 2. SDN-FiWi IoT network communication architecture.

3.1. System Architecture

Figure 2 shows the IoT over a converged FiWi network communication architecture. As explained earlier, the bilateral transmission of H2M/M2M packets crosses internet segments, including optical access networks and embedded software-defined optical access points. (SD-ONU-APs). This architecture consists of three respective layers, such as a service layer (application), transport (control plane) layer and infrastructure layer (data plane); it also consists of two interfaces, i.e., SB-API and NB-API. The service layer is responsible for providing different services for the clients. The transport layer is responsible for transmitting and receiving the data packet information from the applications orchestrating the client to the SDN controller. The infrastructure layer (data plane) is where the actual

physical transfer of control and data packets takes place. The operation administration and management (OAM) system consist of an optical back-end with a single or multiple optical network in its (SD-OLT) located in the central office, which serves single or multiple optical network units (SD-ONU/SD-ONU-AP) on the customer premises. An MUX/DEMUX is used to combine/split the signal in order to drive one or more wavelengths. The feeder fiber line is divided by the power splitter in order to meet the distributed ONU-APs [27]. An advanced SD-ONU-AP that maintains the generic SD-ONU-AP ability and includes an integrated OpenFlow agent and tunable transmitter. An Ethernet passive optical network (EPON)-based back-end is employed, with its typical tree and branch topology. A branch of ONUs may also be located at residential and business subscribers' premises, delivering FTTx services (e.g., fiber to home/buildings) to one or more wired subscribers. Another segment of the SD-ONU-AP is equipped with Wi-Fi mesh networks that consist of mesh and access points to cover the user within the coverage area by radio and fiber technologies (R&F). The end-user IoT devices exchange their sensitivity and surveillance data in the same way as regular front-end user devices with FiWi infrastructures. Typically, front-end IoT device communication is directly connected to wireless SD-ONU-AP.

However, depending on the abilities of a device and its compatibility with radio access technology (RATs), it can connect to the chosen appropriate access mode. Here, central office (CO) resource allocation, to provide end-user (IoT devices) traffic via access networks is critical to improving delay efficiency. To collect the data of IoT/non-IoT device (e.g., cameras, Amazon echo, smoke sensors, mobile, and laptop) traffic, trace files are captured using the Wireshark protocol analyzer tool. Future FiWi-IoT access networks are expected to meet the ever-increasing dynamic bandwidth allocation (DBA) requirements of the next generation of PON and WLAN technologies; this would ensure low latency, a high data rate and broad network coverage for the next generation of wireless networks [28]. Finally, the DBA plays a major role in successfully governing the performance of these integrated networks and mapping the QoS optical wireless networks, as shown in Table 1 and Figure 3. The proposed quality of service class identifier (QoS-CI) mapping is based on traffic types, such as voice, video, IoT and data. QoS parameter mapping changes QoS parameters among access networks, defined for each service requirement [29,30]. However, EPON and 5G/4G WiFi/WLAN offers various types of services (ToS) with different latency and bandwidth requirements, but they have some common properties and, thus, mapping is possible. In addition, traffic classification and class service mapping are performed at the SD-ONU based on their traffic type, characteristics, and QoS specifications in order to support these services via the SD-FiWi-IoT architecture. Due to their specifications, IoT services are assigned the highest priority in this CoS queue, as constant bit rate (CBR) is a type of service that is mainly used for voice traffic and Table 1 shows the CoS priority.

Table 1. Proposed QoS mapping.

Priority	SD-ONU-AP	EPON	WLAN	Designation
1	EF	NC, VO	AC_VO	Voice
2	AF	VI, CL	AC_VI	Video
3	IoT	IoT	AC_IoT	IoT
4	BE	EE, BK	AC_BE, BK	Best Effort, Background

The distributed-coordination function (DCF) model and enhanced distributed-channel access (EDCA) mode of front-end WLAN are depicted in Figure 4a. The basic IEEE 802.11 Medium Access Control (MAC) protocol is DCF, based on the Collision Avoidance (CSMA/CA) and Binary Exponential Backoff (BEB) mechanisms. The pre-specified QoS mapping plan is performed where all data frames (DFs) are represented with a single priority queue (all-in-one mapping) using STA's DCF model, and QoS is provided in DCF mode. The WiFi-AP has the same QoS configuration and STAs, compete with an equal DCF bandwidth. Moreover, through mapping a CoS for each frame in this scheme, no packet

classification is needed for all STA frames, avoiding the associated overhead as in SD-FiWi-IoT. The EDCA is the basic and obligatory 802.11e mechanism; the EDCA classifies traffic flows between four access categories (ACs), each associated with a different transmission queue and acting independently. Therefore, in the EDCA mode, each packet is marked with a user-priority (UP) value that has eight distinct values, i.e., 0 to 7. After marking the packet, it will then be sent to the SD-ONU via the AP. The access category, packets are classified with a right EPON queue (i.e., AC_EF, AC_AF) form in every SD-ONU-AP, as shown in Figure 4b. Packet Classifier classifies the uplink traffic in both the AP and SD-ONU according to their QoS specifications and buffers it into the appropriate priority queues. To the OLT and ONUs, assign DiffServ Code Point (DSCP) values. Additionally, the packet schedulers at the SD-ONU-AP and SD-OLT must use the same packet forwarding approach for EPON and 4G, 5G WLAN/WiFi upstream/downstream traffic with each configured QCI/DSCP value associated with an IP flow. We refer these traffic flows to four sources of traffic types (VOICE, VIDEO, IoT (Alarm), HTTP). The H2H, MTD (IoT) services and applications are listed in Table 2.

Figure 3. Proposed SD-FiWi QoS mapping architecture.

Figure 4. (a) Packet classification WiFi-STA (b) Packet classification of SD-ONU-AP.

Table 2. H2H/IoT Service and Applications Specifications.

Priority	Service/Application	ToS/DSCP	Traffic Type H2H/IoT	Protocols
1	EF VoIP	CS6	H2H	UDP, SIP, VoIP
2	AF Streaming	CS4	H2H	TCP, FTP
3	IoT Live Monitoring	CS4(I)	IoT	UDP, RTSP
4	BE P2P File transfer	CS0	H2H	HTTP, FTP

Furthermore, according to [31], there are three key DWBA framework building blocks for optical wireless networks: QoS mapping block (QoSMB), QoS provisioning block (QoSPB), and (QoSB) scheduling block. First, QoSMB is responsible for solving the QoS diversity issue of allowing various hybrid-network technologies. Second, QoSPB is prompted to decide if the data packet (connection request) on single or multiple criteria is accepted or dropped (rejected). Third, the SB controls how data packets are sent or how data flow from optical to wireless, and vice versa.

3.2. Operation of Software Defined Network

SDN controller is a program that manages one or more SDN-OLTs and SDN-ONU-APs to perform the complex, non-real-time operations that make up FiWi's control plane. Thus, the control plane's SDN controller will track and evaluate the traffic conditions in the access network and reconfigure the forward-plan devices to modify the operating mode accordingly [15]. The controller keeps a log of all SD-OLTs and registered SD-ONU-APs. This database provides statistical details on the OLT wavelength and all SD-ONU-APs average buffer status. It also eliminates the mechanism of re-registration during wavelength switching. The wavelength/link-rate configuration changes are performed by the SDN controller by sending OFPT_MOD_PROP_OPTICAL to SD devices. The Media Access Control (MAC) client on the L-OLT sends GATE messages to assign the SD-ONU-APs transmission time slots. Additionally, the OpenFlow agent activates SD-OLT and SD-ONU control mechanisms, which link the SD-OLT/ONU-APs to the controller and communicate through OpenFlow signaling messages. The protocol communication channels to the OpenFlow switches provide one or more flow tables containing flow entries. Each flow entry contains matching fields and behaviors, and the controller populates the tables. Additionally, the match fields include data packet headers, which include the Internet protocol (IP) source and destination addresses, the port number, and other relevant details. Each behavior determines how the packet's instructions are implemented in accordance with the entry law. The overall operation of network traffic classification in a FiWi environment is focused on actively managing all facets of the network, including SD-OLT initialization, new wavelength activation, wavelength shutdown, SD-ONU-AP wavelength tuning, link-rate tuning, and transmission timeslot monitoring [32].

3.3. Network Traffic Characteristics of IoT Smart Environment

Machine-to-machine (M2M) communication is a fundamental part of the IoT paradigm. IoT-smart environments have unique properties which make them distinctive. The number of devices connecting to the Internet, as well as network traffic, is increasing at an exponential rate. The characteristics of IoT traffic behavior comprise a combination of machine-type communication (MTC) and human-type communication (HTC). MTC was the primary communication when IoT was introduced. At the time, devices and application characteristics were mostly restricted to limited sessions with a few data bytes [18]. However, owing to the new, revolutionary applications, the characteristics of traffic generated by the various MTCs as a result of IoT devices, such as surveillance videos and automobiles, have been completely updated. Furthermore, IoT devices now have the following traffic characteristics based on new applications: short bursts of data sent periodically, short active time, long sleep time, low data rates, and small packet size. The device battery power,

network tolerable delay, large network size and network type are also important traffic characteristics [33]. Hence, meeting the requirements for network traffic characteristics of smart environment applications can be challenging. Another critical feature of traffic is that the network is usually medium to large in scale, linking hundreds and thousands of devices over a wide area. Traffic rates are also irregular and relatively less; however, many applications are based on the detection of rare events, although there is high demand for QoS. Furthermore, most applications require only medium to high security.

4. Overview of Network Traffic Classification Techniques

Network traffic classification is a critical issue in network resource management that emerges from network pattern analysis, as well as network planning and design. Numerous methods have been proposed and implemented over the last two decades. This section discusses some methods for classifying network traffic.

4.1. Port-Based Classification

Port numbers may identify network traffic. These port numbers are assigned by the Internet Assigned Numbers Authority (IANA). The numerous applications of these techniques make use of the port number allocated by IANA to a local host on the network. The World Wide Web and email also use regular port numbers. As a result, it is simple to classify the traffic associated with these applications. However, some applications, such as B2B, gaming, and multi-media, do not use set port numbers. They make use of the port numbers associated with other commonly used applications (e.g., HTTP/FTP connections), which sometimes results in a suboptimal performance. When applications make use of dynamically assigned port numbers, these strategies are ineffective [34].

4.2. Payload-Based Classification

This classification technique is also called packet-based classification or deep packet inspection (DPI). Packet content can be calculated by defining the characteristic signatures of traffic network applications. The majority of payload-based classification algorithms evaluate the packet's contents and compare them to the signatures contained in the database. These approaches are alternated with port-based approaches and have reliable results compared with port-based techniques. They are particularly well-suited for peer-to-peer (P2P) traffic. However, they have some disadvantages and weaknesses [35,36]. They require very expensive hardware for the payload method search. Additionally, they do not work in encrypted network application traffic. Finally, payload-based approaches require a continuous updating of the signature format of new applications.

4.3. Statistical Classification

Statistical classification is a logical technique using statistical properties of network traffic flow to classify the application. Packet duration, packet inter-arrival time, packet length, and traffic flow idle time are some examples of traffic characteristics. As the measured characteristics are unique to each type of application, different implementations may be distinguished from one another [37,38]. Therefore, classifiers must use data processing techniques to perform real classification based on statistical properties, particularly ML methods, because they must handle different traffic patterns from large datasets. Due to their independence from the packet-based technique, ML models are considered lightweight and low-cost. These techniques outperform payload-based techniques; however, as they do not handle packet content. Hence, encrypted traffic can easily be analyzed.

5. Proposed Machine Learning Methodology

This section describes the implementation of the ML classification module. The architecture is depicted in Figure 5, which contains the following functional blocks: packet capture and collected raw data, pre-processing, transformed data, ML-training, ML-testing, ML-classification model, and classification results.

Figure 5. Proposed machine learning based traffic classification process.

5.1. Packet Capture and Collected Block

This module utilizes an SDN-FiWi-IoT interface to catch IoT network packets. The gateway is connected to the Internet through the smart environment, whereas the IoT devices (i.e., smart devices) are connected through the SD-FiWi. Our smart environment has a total of 21 unique IoT and non-IoT devices representing different categories of devices. The function block collects data from the network interface pertaining to IoT network traffic and saves it to a file for further processing. The tcpdump tool of the Wireshark protocol analyzer performs this task [39]. The Wireshark packet analyzer collects information about incoming and outgoing traffic flows and generates associated records. The record contains the entirety of the data contained in that package, from the MAC layer to the application layer. Wireshark provides a graphical user interface for monitoring network traffic, selecting the desired network interface, and capturing packets in real time. The software presents raw data in the form of a hexadecimal dump, as well as distorted information about various protocols used in communications, including source and target IP addresses and ports. The tcpdump tool collects data from a network interface and saves it to an external hard drive as a PCAP file; it also provides several features. All packets were captured from the SD-ONU-AP LAN side.

5.2. Pre-Processing and Transformed Data Block

After the IoT network traffic data are captured, the data are then subjected to pre-processing. The pre-processing block is in charge of receiving the captured file in packet capture (PCAP) format and collecting the necessary information. The block is composed of two functions: identification of traffic and variable extraction. Each packet is labeled by the traffic identification feature according to the system from which it originated: IoT or non-IoT device traffic. This is important because the classifier is supervised by machine learning. The data extraction task generates a collection of statistical variables from information contained in packet headers and payloads. Subsequently, the extraction of features is performed by determining strategies to handle missing fields and altering data as required. Then, useful features that can be used to represent the data are extracted, depending on the goal or task (i.e., the data are transformed using the labeling of IoT/Non-IoT Traffic). The effective number of variables under consideration can be reduced or invariant representations for the data can be found using dimensional reduction or transformation methods. The transformed features include port source and destination numbers, IP source and destinations, domain name services (DNS) and NTP, packet size, etc. Further, the extracted features are used to train and test the ML classifier [40].

5.3. Training Block

The aim of this module, based on all features extracted by the pre-processing module, is to enforce feature selection. The selection of features allows for the creation of a scalable model, depending on the features, and offers a more accurate dataset classification. The selected train module method for this research is based on the white-listed, where a binary classifier is generated for each device type. The predict module can be used to directly classify models from the train to predict device type and feature name: packet size, packet id, port number, and DNS. The classifier can be trained using the training dataset. Then, the effectiveness of the proposed supervised classifier was evaluated using the classifier to classify an independent test dataset. During the training phase, the information is used to identify new examples, which are not present in the experimental phase (classification process); 70% of the observations in the original dataset were placed in the training set.

5.4. Testing Block

The algorithm was then used to classify test data. The test block uses the model in the training block to identify new instances. Datasets used in the training and testing blocks must be independent and labeled in advance. When training the dataset, “flow statistics processing” is implemented by calculating the statistical properties of these flows (packet id, time, size, ethernet source and destination, IP source and destination, port numbers, etc.) as a prelude to the generation of features.

5.5. Implementation of ML Classification Model (Pattern) Block

This block contains the implementation step; this process involves determining which models and parameters could be suitable and matching a specific data-mining method to apply ML classifiers to different instances. We used six well-known data classification machine learning classifiers. ML algorithms are used: Random Forest (RF), (SVM), K-Nearest Neighbor (KNN), Neural Network-Multi-Layer Perception (MLP), Naive Bayes, logistic regression and Support Vector Machine (SVM). These algorithms and technical details are described as follows. First, a Random Forest is a meta-estimator that fit a set of decision tree classifiers on different dataset sub-samples and uses an average to boost predictive accuracy and control over-fitting issues. Further, RF is an ensemble ML algorithm. Second, a supervised, non-parametric classification method is k-Nearest-Neighbors (KNN). In the technique, k training samples are found with relatively similar (closest) attributes to test samples. These samples are called Nearest-Neighbors. Third, in a neural network, MLP uses a supervised learning technique called backpropagation. Its multiple layers and non-linear activation distinguish MLP from a linear perception. Fourth, the NB is a method used to classify data, based on the Bayes theorem. This paper is used on the gaussian naive Bayes. Fifth, LR is a statistical analysis technique that employs regression analysis to ascertain a quantitative relationship between two or more variables in mathematical statistics. Finally, the SVM is an ML technique that separates the attribute space with a hyperplane, maximizing the margin between different class or class value instances. The technical details of the studied classification techniques are tuned and the parameters are given in Table 3. Moreover, a comparative analysis of the algorithms was performed [41]. Various models were employed for comparison.

5.6. Classification Result Block

After the implementation of machine learning classifiers, the simulation tool presents the results in terms of classification accuracy.

Table 3. ML Model Technical configuration.

Algorithm	Technical Details
Random Forest (RF)	Number of estimator tree: 10, Number of trees considered at each split: 5, Replicable training: Fix the seed for tree generation, Balance distribution: weigh classes and Do not split subsets smaller than 5
K-nearest neighbors (KNN)	Number of neighbors (K) 5. Metric Euclidean and weight Uniform
Neural Network (MLP)	Neurons per hidden layers: 100, Activation: ReLu, solver: Adam Maximal number of iterations: 200
Naive Bayes (NB)	GaussianNB
Logistic Regression (LR)	Regularization type: Ridge (L2)
Support Vector Machine	Radial Basics Function (RBF)

6. Performance Evaluation

This section presents the effects of applying multiple machine learning models to classify the device. We utilized publicly available datasets to evaluate our proposed ML classification module. First, we provide a dataset overview. Next, we present performance indicators, system classification, experimental configuration, and results. This section ends by discussing future work.

6.1. Dataset

We used the dataset collected by Sivanathan et al. [21]. The dataset was collected in the IoT smart environment using different types of IoT/non-IoT device traffic traces, captured over 20 days and released online. We used a total of 21 devices from the UNSW dataset containing different types of IoT/non-IoT devices. There are a total of 28 IoT/non-IoT genre devices or different group devices in UNSW [21]: non-IoT devices, such as MacBook, laptop, Samsung Galaxy and TB-link, and IoT devices such as Amazon echo, Tribby speaker, HP printer, Smart Things, Netatmo Weather Station, Netatmo Welcome, and Withings. The IoT/non-IoT devices belong to various categories, such as smart health, smart homes and cities. Although the categories differed from the classification, in this study, in total, we used 21 UNSW IoT and non-IoT devices. The MAC address and connectivity information of all these devices is listed in Table 4.

6.2. Performance Metrics

In this study, we used a confusion matrix to evaluate the classification performance. Our goal is to classify IoT and non-IoT device network traffic to identify specific devices and improve classification accuracy. To measure the accuracy of the classifier, several performance metrics are introduced. These metrics are calculated using the classification results of the ML classifier [42]. The following performance metrics are frequently used for binary classification:

TP: True positive;
 FP: False positive;
 FN: False negative;
 TN: True negative.

We used the aforementioned four metrics, along with the confusion matrix, to evaluate the performance of each model.

Thus, the calculation is as follows:

$$\text{Current accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$(\text{Sens/Spec}) \text{ Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F1 \text{ Score} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

The F1-score is the harmonic mean of precision and recall and indicates the classification accuracy of a machine-learning-supervised classification model. We calculated all device labels' accuracy, recall, accuracy, and F1-score. Then, we averaged them to obtain overall accuracy and F1-score measurements for the performance comparison of the six ML models. Note that F1 represents the balance between accuracy and recall values, and is calculated by the harmonic mean of these two values in Equation (4). For each model class, all measurements take a value of 0-1 to obtain the FP and FN of all labels. In our multi-class models, we derive FP and FN by summation across all "incorrect" labels.

Table 4. List of IoT/non-IoT devices used in the experiment.

No	Device	Mac Address	Connectivity
1.	Amazon echo	44:65:0d:56:cc:d3	WiFi
2.	Belkin Motion Sensor	ec:1a: 59:79: f4:89	WiFi
3.	Belkin Switch	ec:1a: 59:83:28:11	WiFi
4.	Dropcam	30:8c: fb:2f: e4: b2	WiFi
5.	HP Printer	70:5a:0f: e4:9b:c0	WiFi
6.	Insteon Camera	00:62:6e: 51:27:2e	Wired
7.	Laptop	74:2f: 68:81:69:42	WiFi
8.	LiFx Smart Bulb	d0:73: d5:01:83:08	WiFi
9.	MacBook	ac: bc:32: d4:6f:2f	WiFi
10.	Netamo Welcome	70:ee: 50:18:34:43	WiFi
11.	Netatmo Weather Station	70:ee: 50:03: b8:ac	WiFi
12.	PIX-Star Photo Frame	e0:76: d0:33:bb:85	WiFi
13.	Samsung Galaxy Tab	08:21: ef:3b:fc: e3	WiFi
14.	Samsung Smart Cam	00:16:6c: ab:6b:88	WiFi
15.	Smart Things	d0:52: a8:00:67:5e	Wired
16.	TB-Link	14:cc: 20:51:33: ea	Wired
17.	Triby Speaker	18: b7:9e: 02:20:44	WiFi
18.	TP-Link Smart plug	50:c7:bf: 00:56:39	WiFi
19.	Withings Baby Monitor	00:24: e4:11:18: a8	Wired
20.	Withings Scale	00:24: e4:1b:6f:96	WiFi
21.	Withings Sleep Sensor	00:24: e4:20:28:c6	WiFi

6.3. Experimental Setup

The proposed method was implemented using the orange-ML simulation tool [43] with the system configuration shown in Table 5. In the simulation environment, we used 21 IoT/non-IoT devices and 41911 instances from the traffic traces, and the total instance counts of each class are shown in Table 6. The 21 devices were chosen based on the low traffic generated by the devices (e.g., Smoke Alarm, Netamo Weather Station, Netamo Welcome). The following features are used to simulate the environment: packet id, size, ethernet source and destination, port source, destination, IP source and destination, and DNS. The instances were randomly divided into two groups: 70% of training and 30% for testing.

Table 5. System configuration.

Operating System	Windows 10 x64-based PC
Processor	Intel(R) Core (TM) i7-9700 CPU @ 3.00GHz, 3000 MHz, 8 Core(s), 8 Logical Processor(s)
RAM	40.0 GB
Hard Disk	2 TB

Table 6. Devices total instance count.

Device	Instance Count
Amazon echo	1750
Belkin Motion Sensor	1900
Belkin Switch	1940
Dropcam	2100
HP Printer	2400
Instean Camera	1900
Laptop	2200
LiFx Smart Bulb	1955
MacBook	1800
Netamo Welcome	1400
Netatmo Weather Station	2444
PIX-Star Photo Frame	1990
Samsung Galaxy Tab	1850
Samsung Smart Cam	1930
Smart Things	2000
TB-Link	2100
Triby Speaker	1952
TP-Link Smart plug	1950
Withings Baby Monitor	2100
Withings Scale	2400
Withings Sleep Sensor	1850

6.4. Device Classification and Analysis of Receiver Operating Characteristics (ROC) Curve

The types of IoT device that can appear on the network are classified into specific IoT and non-IoT device types. An initial activity includes the creation of classification models to differentiate traffic between devices. Our method is structured as follows: each IoT/non-IoT system traffic capture is expressed as $x = (x_1, x_2, \dots, x_n)$. A function vector represents these traffic flows. Hence, each function vector x_i should be assigned a label. As the number of devices increases over time, we construct a single classifier for each device class, and n different classifiers for each device. As a result, each classifier is a binary classifier that decides whether the unknown device's input feature vector fits the device class or not. This method is known as one-vs.-all. For each classification model, the class fits all other classes. This is the most-used multi-class classification technique. Additionally, we used the receiver operating characteristic (ROC) curve and area under the curve (AUC) as statistical measures. The receiver operating characteristic (ROC) curve is a commonly used machine learning and data mining technique. This graph illustrates the relationship between the TP rate (sensitivity) and the FP rate (1-specificity). The classification of correct and incorrect results at various thresholds shows the model's overall efficiency, and ROC

provides visual and numerical descriptions of a classifier's behavior. Additionally, the region under the curve (AUC) has gained significantly more attention in the ML community and is a widely used performance metric in supervised machine learning. The AUC score is used as a criterion because, under the binary classification problem, the data are generally balanced. This means that most feature vectors usually do not represent a device; some feature vectors represent devices. Therefore, the accuracy indicator may not be sufficient to reflect the distribution of the base classes. The receiver operating characteristics (ROC) represents the recall and precision based on the true false positive rate (TPR) and the false positive rate (FPR) [44]. The closer the area under the curve is to 1, the better the classification. Here, we use the average AUC score to measure the model performance. A classifier efficiency measurement with a high AUC score is considered favorable. The AUC and ROC curves in Figure 6a,b illustrate the predictive efficiency of the Amazon Echo and Belkin motion sensors, respectively. Figure 6, shows the Random Forest and KNN classifier and received a higher AUC score of 1.00 and 0.995 compared to other ML algorithms. For all attributes, the Random Forest Classifier yields a greater region under ROC curves.

(a)

(b)

Figure 6. Receiver operating characteristic (ROC) curve for demonstrating the significance of the proposed method with various ML classifiers: (a) Amazon echo, (b) Belkin motion sensors.

6.5. Overall Performance Result

In this section, we describe our experimental process and data-based results. After implementation of the machine learning algorithm, the simulation tool provides detailed results about the applied machine learning algorithms, such as (1) area under the curve (AUC), (2) current accuracy (CA), (3) precision, (4) recall, and (5) F1 score. Table 7 shows the predictive accuracy of the proposed model classification and is shown in Figure 7.

Table 7. Accuracy result and machine learning algorithm.

Algorithm	AUC	CA	F1 Score	Precision	Recall
Random Forest	1.000	0.996	0.999	0.996	0.996
KNN Tree	0.995	0.968	0.968	0.968	0.968
Neural Network	0.998	0.952	0.951	0.953	0.952
Naïve Bayes	0.991	0.875	0.871	0.884	0.875
Logistic Regression	0.815	0.547	0.425	0.347	0.547
SVM	0.907	0.485	0.474	0.524	0.485

Figure 7. Graph of device classification results with attributes.

It is clear that the feature set selected in this work is robust, effective and achieves excellent performance results for the IoT datasets. The experimental study used six well-known ML supervised classifiers. All the applied ML classifiers were found to have good classification performance. The random forest classifier showed the best overall performance in our experiments. Figure 7 shows the comparison of the classification current accuracy, F1 score, precision and recall of the six ML-algorithms. The Random Forest algorithm outperforms all other algorithms due to its high tolerance of overfitting in comparison to other decision tree classifiers. However, with an accuracy of 0.48 and 0.54, respectively, the SVM and logistic regression do the worst of all. The accuracy of the other algorithms ranges between 0.996 and 0.87, and the Random Forest is the optimal algorithm. Figure 8 presents the confusion matrix of the IoT and non-IoT classifier and devices based on the above performance metrics (True/False, positive/negative) occurrences, where the rows represent the actual device and the columns represent the expected classification correctly. We note that the proposed model works well at predicting most classes. The diagonal entries are close to 100 percent classified, with just four exceptions: the

Amazon Echo	97.80	0.00	0.00	0.00	0.00	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Belkin Motion Monitor	0.00	99.50	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Belkin Switch	0.00	0.03	99.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Dropcam	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HP Printer	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Instean Camera	0.11	0.00	0.00	0.00	0.00	98.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LIFX Smart Bulb	0.00	0.00	0.00	0.00	0.00	0.00	99.20	0.00	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Labtop	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MacBook	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	98.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11
Netamo Weather Station	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Netamo Welcome	0.13	0.00	0.00	0.00	0.00	0.19	0.00	0.00	0.00	0.01	92.30	0.00	0.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PIX-STAR Photo-frame	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Samsung Galaxy Tab	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	98.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Smart Things	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Smasung Smart Cam	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	99.90	0.00	0.00	0.00	0.00	0.00
TB-Link	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
TP-Link Smart plug	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00
Triby Speaker	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00
Withings Sleep Sensor	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.04	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	98.50	0.00
Withings Smart Baby Monitor	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
Withings Smart scale	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Figure 8. IoT device classification confusion matrix using all attributes.

Amazon Echo, Belkin Motion Sensor, Instean Camera Netamo Welcome, and the Withings Sleep Sensor.

6.6. Discussion on Our Work with Related Work

In this subsection, in this section, we present our insights into the realistic implementation on model analysis and IoT data acquisition compared to previous studies. Our proposed methodology is a generic IoT system classification tool. Our experimental findings in Section 6 show that our method could be able to automatically classify new IoT devices by analyzing their network traffic sources, which are generally easy to acquire. We use a multi-class classification using supervised ML models to identify the individual IoT and non-IoT devices. We classify the test data from the proposed several ML models. The experimental results show that the proposed RF model achieves higher predictive accuracy than existing reference models. After that, we analyze the AUC and ROC curve to find the best performance of the device identification. The confusion matrix of a 21 device classification test (shown in Figure 8) shows that our approach fails to accurately distinguish Netatmo welcome, MacBook and Amazon Echo. This can be attributed to the very limited information available in our small-scale dataset. Moreover, this experiment's devices and smart environment are one of the use cases, and more development is needed to extend the proposed user identifier to different devices in actual smart environments. Finally, we compare our methods to similar studies in terms of the method, features, number of devices, identification speed and accuracy. The Table 8 represents a summary of the state-of-the-art from various perspectives.

Table 8. Comparison of the proposed work with state-of-the-art techniques.

Work	Purpose	Methods	Features	Devices	Speed/Accuracy
[19]	Automatically identify the devices using (TCP/IP)	Decesion Tree (DT), K48, OneR, and PART	Unique featurrs, Transport and application layer	23 IoT devices	Fast/95%
[20]	Identifying and authenticating behavioral fingerprints	KNN, Decesion Tree (DT), XG Boost Random Forset	Features based on TCP sessions	14 home IoT devices	Fast/99%
[21]	To classify IoT devices using traffic characteristics	Random Forest	Statistical attributes: activity cycles, port number and cipher suites	28 IoT devices	Fast/99%
[24]	Autonomous identification of IoT device type	RF, DT, LR, SVM, Navie Bayes (NB)	Textual featurrs and flows features of the network traffic	28 Hetrogeneous devices	Fast few seconds/ 97%
[25]	Detection of Unauthorized IoT Devices classification	Detection of Unauthorized IoT Devices	TCP/IP	17 IoT devices	Fast/ 99.49%
[26]	IoT device classification	LSTM-CNN	TCP session features	15 IoT devices with four grpups	74.8%
[44]	To identify IoT device types from the white list	RF, Gradient Boosting Machine (GBM), XG Boost	TCP/IP features	9 IoT devices, PC and smart phones	Fast/99%
Our work	IoT/Non-IoT devices identification	Multiclass-classification RF, KNN, NB, SVM, MLP, LR	Statistical Features	21 IoT/Non-IoT deviccs (latop, HP printer, Smart Phone)	Fast Few seconds/ 99%

Our method is subject to the following requirements and limitations:

1. The tested IoT and Non-IoT devices are various enough with 21 devices;
2. The coverage is complete and 99% accuracy is good enough;
3. The study only examined devices that communicated through TCP/IP;
4. We collect harmless IoT and non-IoT traffic flow, i.e., we do not abuse or unusually use the IoT system. As a result, our assumptions only apply to the capture of the usual activity patterns of a variety of IoT system types.

7. Conclusions

This paper implements an end-to-end network traffic classification system based on a fiber wireless access network by mapping an Ethernet passive optical network (EPON) and wireless local area network (WLAN) traffic, based on the quality of service class identification (QoS-CI) for traffic types such as voice, video, IoT and data. The identification of the devices that comprise a network, referred to as network mapping, serves as the foundation for a variety of network management applications, ranging from resource allocation and network slicing to security management. The proposed ML process modules were tested with the UNSW dataset. We collected a smart environment dataset with 21 unique IoT devices, analyzed the trace file and extracted the traffic behavior features. Then, we used multiclass classification techniques that were uniquely identified with the individual devices. We employed a multi-class, machine-learning-based classification system to ensure that IoT devices are uniquely identified. Six different supervised machine-learning models were used to automatically classify specific IoT/non-IoT devices. We found that the proposed random forest classifier achieved 99% accuracy compared to other classifiers (KNN, logistic regression, SVM, neural network, Naive Bayes), and the identified speed is also quick at classifying specific device types using behavior features on the UNSW dataset. Despite their being room for progress, our work successfully demonstrates an ability to automatically identify IoT/non-IoT devices based on their network traffic flows. In future work, we intend to study the classification of anomaly detection datasets using different machine learning approaches. The goal of anomaly detection is important for the extraction of essential business insights and maintenance of key functions. Anomaly

detection is a critical tool for detecting fraud, network interference, and other unusual but significant incidents.

Author Contributions: A.T.L. and M.S.A.-R. studied the proposed the system architecture and SD-FiWi-IoT operations; E.G. is responsible for the data collection, data analysis, and simulation; I.-S.H. conducted the research and edited the paper; all authors had approved the final version. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in Ministry of Science and Technology (MOST): 109-2221-E-155-029-MY2, Taiwan.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the anonymous referees who gave precious suggestions to improve this work.

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

Abbreviations

Notations	Description
SDN	Software Defined Network
OF	OpenFlow
FiWi	Fiber-Wireless
IOT	Internet of Things
DBA	Dynamic Bandwidth Allocation
DWBA	Dynamic Wavelength and Bandwidth Allocation
QoS	Quality of Service
EDCA	Enhanced Distribution Channel Access
DCFPC	Distributed-Coordination Function Packet Classifier
DSCP	Differentiated Services Code Point
PON	Passive Optical Network
TWDM	Time and Wavelength Division Multiplexing
TDM	Time Division Multiplexing
NBAPI	NorthBound Application Programming Interface
SBAPI	SouthBound Application Programming Interface
SD-OLT	Software-Defined-Optical Line Terminal
SD-ONU-AP	Software Defined-Optical Network Unit-Access Point
EF	Expedited Forwarding (Voice)
AF	Assured Forwarding (Video)
BE	Best Effort
CoS	Class of Service
ToS	Type of Service
CBR	Constant Bit Rate
H2M	Human-To-Machine Communication

References

1. Rimal, B.P.; Van, D.P.; Maier, M. Mobile edge computing empowered fiber-wireless access networks in the 5G era. *IEEE Commun. Mag.* **2017**, *55*, 192–200. [CrossRef]
2. Yousefnezhad, N.; Malhi, A.; Framling, K. Automated IoT device identification based on full packet information using real-time Network traffic. *Sensors* **2021**, *21*, 2660. [CrossRef] [PubMed]
3. Huxley. Available online: <https://www.huxley.com/en-sg/blog/2018/06/the-use-of-iot-devices-is-expected-to-reach-18-billion-by-2022-what-does-this-mean-for-cyber-security/> (accessed on 4 June 2021).
4. Liu, Y.; Yang, Y.; Han, P.; Shao, Z.; Li, C. Virtual network embedding in fiber-wireless access networks for resource-efficient IoT service provisioning. *IEEE Access* **2019**, *7*, 65506–65517. [CrossRef]
5. Van, D.P.; Rimal, B.P.; Chen, J.; Monti, P.; Wosinska, L.; Maier, M. Power-saving methods for Internet of Things over covered fiber-wireless access networks. *IEEE Commun. Mag.* **2016**, *54*, 166–175. [CrossRef]
6. Hwang, I.S.; Yeah, T.J.; Hwang, B.J.; Lee, J.Y. Synchronous interleaved dynamic bandwidth assignment for quality of service over GPON-LTE converged network. *J. Internet Technol.* **2015**, *16*, 1259–1270.

7. Mehamood, Y.; Ahmad, F.; Yaqoob, I.; Adnane, A.; Imran, M.; Guizani, S. Internet-of-Things-based smart cities: Recent advances and challenges. *IEEE Commun. Mag.* **2017**, *55*, 16–24. [\[CrossRef\]](#)
8. Shafiq, M.; Yu, X.; Laghari, A.; Yao, L.; Karan, N.K.; Abdessamia, F. Network traffic classification techniques and comparative analysis using machine learning algorithms. In Proceedings of the IEEE International Conference on Computer and Communications, Chengdu, China, 14–17 October 2016; pp. 2451–2455.
9. Sivanathan, A. IoT behavioral monitoring via network traffic analysis. *arXiv* **2020**, arXiv:2001.10632.
10. Deepika, V.; Samrudhi, N. Software-defined networks. *IEEE Potentials* **2018**, *37*, 21–24.
11. Farris, I.; Taleb, T.; Khettab, Y.; Song, J. A survey on emerging SDN and NFV security mechanisms for IoT systems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 812–837. [\[CrossRef\]](#)
12. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Wang, C.; Liu, Y. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 393–430. [\[CrossRef\]](#)
13. Kuranage, M.P.J.; Piamrat, K.; Hamma, S. Network Traffic Classification Using Machine Learning for Software Defined Networks. In *Machine Learning for Networking*; Boumerdassi, S., Renault, E., Muhlethaler, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12081, pp. 28–39.
14. Yao, H.; Gao, P.; Wang, J.; Zhang, P.; Jiang, C.; Han, Z. Capsule network assisted IoT traffic classification mechanism for smart cities. *IEEE Internet Things J.* **2019**, *6*, 7515–7525. [\[CrossRef\]](#)
15. Liu, J.; Shou, G.; Liu, Y.; Hu, Y.; Guo, Z. Performance evaluation of integrated multi-access edge computing and fiber-wireless access networks. *IEEE Access* **2018**, *6*, 30269–30279. [\[CrossRef\]](#)
16. He, J.; Lee, J.; Kandeepan, S.; Wang, K. Machine Learning Techniques in Radio-over-Fiber Systems and Networks. *Photonics* **2020**, *7*, 105.
17. Amaral, P.; Dinis, J.; Pinto, P.; Bernardo, L.; Tavares, J.; Mamede, H.S. Machine learning in software defined networks: Data collection and traffic classification. In Proceedings of the IEEE 24th International Conference on Network Protocols (ICNP), Singapore, 1–5 November 2016.
18. Sarica, A.K.; Angin, P. Explainable security in SDN-based IoT networks. *Sensors* **2020**, *20*, 7326. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Bezawada, B.; Bachani, M.; Peterson, J.; Shirazi, H.; Ray, I.; Ray, I. Behavioral fingerprinting of IoT devices. In *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security, ASHES@CCS 2018, Toronto, ON, Canada, 19 October 2018*; Chang, C., Rührmair, U., Holcomb, D., Guajardo, J., Eds.; ACM: New York, NY, USA, 2018; pp. 41–50.
20. Aksoy, A.; Gunes, M.H. Automated IoT device identification using network traffic. In Proceedings of the 2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, 20–24 May 2019; pp. 1–7.
21. Sivanathan, A.; Gharakheili, H.H.; Loi, F.; Radford, A.; Wijenayake, C.; Vishwanath, A.; Sivaraman, V. Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Trans. Mob. Comput.* **2018**, *18*, 1745–1759. [\[CrossRef\]](#)
22. Zhang, J.; Chen, X.; Xiang, Y.; Zhou, W.; Wu, J. Robust network traffic classification. *IEEE Trans. Comput. Soc. Syst.* **2015**, *2*, 1257–1270. [\[CrossRef\]](#)
23. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Network traffic classifier with convolutional and recurrent neural network for internet of things. *IEEE Access* **2017**, *5*, 18042–18050. [\[CrossRef\]](#)
24. Ammar, N.; Noirie, L.; Tixeul, S. Autonomous identification of IoT device types based on a supervised classification. In Proceedings of the ICC 2020–2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
25. Meidan, Y.; Bohadana, M.; Shabtai, A.; Ochoa, M.; Tippenhauer, N.O.; Guarnizo, J.D.; Elovici, Y. Detection of unauthorized IoT devices using machine learning techniques. *arXiv* **2017**, arXiv:1709.04647.
26. Bai, L.; Yao, L.; Kanhere, S.S.; Wang, X.; Yang, Z. Automatic device classification from network traffic streams of internet of things. In Proceedings of the IEEE 43rd Conference on Local Computer Networks (LCN), Chicago, IL, USA, 1–4 October 2018; pp. 1–9.
27. Van, D.P.; Rimal, B.P.; Andreev, S.; Tirronen, T.; Maier, M. Machine-to-machine communication over FiWi enhanced LTE networks: A power-saving framework and end-to-end performance. *J. Lightwave Technol.* **2016**, *34*, 1062–1071. [\[CrossRef\]](#)
28. Bhatt, U.R.; Sharma, A.; Mishra, V.; Upadhyay, R. Dynamic bandwidth allocation in fiber-wireless (FiWi) access networks. *Eur. J. Adv. Eng. Technol.* **2017**, *4*, 668–677.
29. Hwang, I.S.; Lee, J.Y.; Lu, C.H.; Rahman, M.S.A.; Liem, A.T. Hybrid uplink traffic scheduling algorithms in FMC networks: A comparative study of performance. *J. Internet Technol.* **2017**, *18*, 521–532.
30. Ganesan, E.; Hwang, I.-S.; Liem, A.T.; Ab-Rahman, M.S. 5G-enabled tactile internet resource provision via software-defined optical access networks (SDOANs). *Photonics* **2021**, *8*, 140. [\[CrossRef\]](#)
31. Mohammadani, K.H.; Butt, R.A.; Memon, K.A.; Hassan, F.; Majeed, A.; Kumar, R. Highest cost first-Based QoS mapping scheme for fiber wireless architecture. *Photonics* **2020**, *7*, 114. [\[CrossRef\]](#)
32. Pakpahan, A.F.; Hwang, I.S.; Nikoukar, A. OLT energy savings via software defined dynamic resource provisioning in TWDM-PONs. *IEEE J. Opt. Commun. Netw.* **2017**, *9*, 1019–1029. [\[CrossRef\]](#)
33. Mocnej, J.; Pekar, A.; Seah, W.K.G.; Zolotova, I. Network Traffic Characteristics of the IoT Application Use Cases. Available online: https://ecs.victoria.ac.nz/foswiki/pub/Main/TechnicalReportSeries/IoT_network_technologies_embfonts.pdf (accessed on 28 March 2021).
34. Dashevskiy, M.; Luo, Z. Network traffic classification and demand prediction. In *Conformal Prediction for Reliable Machine Learning*; Balasubramanian, V.N., Ho, S.-S., Vovk, V., Eds.; Morgan Kaufmann: Burlington, MA, USA, 2014; pp. 231–259.

35. Goli, Y.G.; Ambika, R. Network traffic classification techniques—A review. In Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 21–22 December 2018; pp. 219–222.
36. Finsterbusch, M.; Richter, C.; Rocha, E.; Muller, J.; Hanssger, K. A survey of payload-based traffic classification approaches. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1135–1156. [[CrossRef](#)]
37. Fan, Z.; Liu, R. Investigation of machine learning based network traffic classification. In Proceedings of the International Symposium on Wireless Communication Systems (ISWCS), Bologna, Italy, 28–31 August 2017; pp. 1–6.
38. Tahaei, H.; Afifi, F.; Asemi, A.; Zaki, F.; Anuar, N.B. The rise of traffic classification in IoT networks: A survey. *J. Netw. Comput. Appl.* **2020**, *154*, 1–20. [[CrossRef](#)]
39. To Capture Online Traffic, Wireshark Tool, Application. Available online: <https://www.wireshark.org/> (accessed on 2 March 2021).
40. Nguyen, T.T.T.; Armitage, G. A survey of techniques for internet traffic classification using machine learning. *IEEE Commun. Surv. Tutor.* **2008**, *10*, 56–76. [[CrossRef](#)]
41. Narayanan, U.; Unnikrishanan, A.; Paul, V.; Joseph, S. A survey of various supervised classification algorithms. In Proceedings of the International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 1–2 August 2017; pp. 2118–2124.
42. Chen, C.; Zhang, J.; Xie, Y.; Xiang, Y.; Zhou, W.; Hassan, M.M.; Abudlhameed, E.; Majed, A. A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE. Trans. Comput. Soc. Syst.* **2015**, *2*, 65–76. [[CrossRef](#)]
43. Machine Learning and Data Visualization Tool. Available online: <https://orange.biolab.si/> (accessed on 2 March 2021).
44. Meidan, Y.; Bohadana, M.; Shabtai, A.; Guarnizo, J.D.; Ochoa, M.; Tippenhauer, N.O.; Elovici, Y.; Profilio, T. A machine learning approach for IoT device identification based on network traffic analysis. In Proceedings of the Symposium on Applied Computing, Marrakech, Morocco, 3–7 April 2017; pp. 506–509.