IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 8, 2022, accepted April 1, 2022, date of publication April 8, 2022, date of current version April 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3165809

Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning

MOHAMED AMINE FERRAG[1], OTHMANE FRIHA[2], DJALLEL HAMOUDA[3], LEANDROS MAGLARAS[4], (Senior Member, IEEE), AND HELGE JANICKE[5], (Member, IEEE)

[1]Department of Computer Science, Guelma University, Guelma 24000, Algeria
[2]Networks and Systems Laboratory (LRS), Badji Mokhtar—Annaba University, Annaba 23000, Algeria
[3]Labstic Laboratory, Guelma University, Guelma 24000, Algeria
[4]School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.
[5]Cyber Security Cooperative Research Centre (CSCRC), Edith Cowan University, Perth, WA 6027, Australia

Corresponding author: Leandros Maglaras (leandros.maglaras@dmu.ac.uk)

ABSTRACT In this paper, we propose a new comprehensive realistic cyber security dataset of IoT and IIoT applications, called Edge-IIoTset, which can be used by machine learning-based intrusion detection systems in two different modes, namely, centralized and federated learning. Specifically, the dataset has been generated using a purpose-built IoT/IIoT testbed with a large representative set of devices, sensors, protocols and cloud/edge configurations. The IoT data are generated from various IoT devices (more than 10 types) such as Low-cost digital sensors for sensing temperature and humidity, Ultrasonic sensor, Water level detection sensor, pH Sensor Meter, Soil Moisture sensor, Heart Rate Sensor, Flame Sensor, etc.). Furthermore, we identify and analyze fourteen attacks related to IoT and IIoT connectivity protocols, which are categorized into five threats, including, DoS/DDoS attacks, Information gathering, Man in the middle attacks, Injection attacks, and Malware attacks. In addition, we extract features obtained from different sources, including alerts, system resources, logs, network traffic, and propose new 61 features with high correlations from 1176 found features. After processing and analyzing the proposed realistic cyber security dataset, we provide a primary exploratory data analysis and evaluate the performance of machine learning approaches (i.e., traditional machine learning as well as deep learning) in both centralized and federated learning modes. The Edge-IIoTset dataset can be publicly accessed from http://ieee-dataport.org/8939.

INDEX TERMS Cybersecurity applications, IoT datasets, deep learning, federated learning, edge computing.

I. INTRODUCTION

The Internet of Things (IoT) is a connected network of equipment that has the ability to communicate with each other and provide data to users via the Internet. The explosive growth of IoT in recent years is due in part to its broad applicability, scalability, and support for smart applications. The majority of IoT applications perform tasks in an automated fashion, with little or no interaction with humans.

Industrial IoT (IIoT) is a subclass of IoT, where IoT devices are used in typically closed industrial environments. IIoT has been successful in producing significant resource savings, while increasing productivity [2]. IIoT represents a critical enabler of Industry 4.0, often referred to as the next industrial revolution [3]. Currently, there are more than 8 billion IoT-connected devices, and the number is expected to reach 41 billion by 2027 [4]. In 2021, the global IoT market size was estimated to be above $380 billion and is expected to reach over $1.8 trillion by 2028, growing at a CAGR of 25.4% from 2021 to 2028 [5], with sectors such as automotive, smart home, manufacturing, energy, healthcare, transportation, logistics, and media being at the forefront of IoT evolution.

The enormous increase in IoT calls for appropriate security and privacy policies to prevent potential vulnerabilities and threats introduced by the implementation of this technology.

The associate editor coordinating the review of this manuscript and approving it for publication was Shafiullah Khan.

Furthermore, other key considerations in IIoT, including trustworthiness, expandability, and energy usage, must be addressed, given that legacy security fixes are falling short in many cases [6]. According to Kaspersky researchers, the number of cyberattacks against IoT devices jumped to 1.5 billion up from 639 million in one year period (2020-2021), which represents more than 100% increase, as cybercriminals have cleverly turned their attention to this space, seeking to rob data, mine cryptocurrencies, and create botnets [7]. Another favorite weapon of hackers lately is the ransomware attack, as the average ransom amount paid by organizations jumped by 311% in 2020 and hit about $350 million in crypto-currency, according to a report released by the Ransomware Task Force [8]. For instance, in the first half of 2021, DarkSide (a Russian-based hacker group) claimed responsibility for a ransomware attack on Colonial Pipeline, one of the largest fuel pipelines in the U.S., and forced it to have its SCADA systems down and pay nearly $5 million in a hard-to-trace crypto-currency.

Considerable work has been done by the cybersecurity community in creating sophisticated security tools and techniques for protecting users and data in traditional IT systems. Yet, these measures themselves cannot be immediately deployed for IoT/IIoT-based systems. Many existing techniques are insufficient to address novel threats that can breach IoT networks, making it necessary to delve deeper into advanced forensic approaches to detect and investigate malicious behavior [10]. Purpose-built cybersecurity solutions, that are tailored to IoT and IIoT systems, are needed to manage the limitations such as constrained functionality, limited power, and lightweight network protocols [14], [15]. One such solution are Intrusion Detection Systems (IDS), and their ability to provide detection and surveillance of attacks throughout their lifecycle, enabling a response to advanced persistent threats that can evade existing security measures. [12], [16].

Intrusion detection techniques that are based on machine-learning require training and ongoing callibration using centralized or federated learning approaches [17]–[21]. A key success factor in training IDS is choosing the right dataset. For IoT/IIoT systems security, it is critical to use datasets that closely mirror real-world IoT/IIoT applications. The scarcity of available IoT/IIoT datasets presents a significant barrier to the evaluation of IDS solutions tailored for IoT/IIoT systems. This scarcity of data is mainly caused by privacy concerns. Therefore, a great number of major corporations that are building such datasets are discouraged from sharing it publicly [22].

The goal of our work presented in this paper is to provide a comprehensive dataset that can be used for developing and accurately validating IoT/IIoT security solutions. We propose a new IoT and IIoT dataset collected from a sophisticated seven-layer testbed including more than 10 IoT devices, IIoT-based Modbus flows, 14 IoT and IIoT protocol-related attacks. In addition, a detailed description of the dataset and its features is given in this paper. Furthermore, using the dataset, we have evaluated the performance of intrusion detection through several supervised machine learning methods using two different learning approaches, namely centralized and federated learning. The Edge-IIoTset dataset can be publicly accessed from [1].

Our research contributions are as follows:

- We present a new platform for creating a new comprehensive realistic cybersecurity dataset of IoT and IIoT applications. The testbed is organized into seven layers: 1. Cloud Computing Layer, 2. Network Functions Virtualization Layer, 3. Blockchain Network Layer, 4. Fog Computing Layer, 5. Software-Defined Networking Layer, 6. Edge Computing Layer, and 7. IoT and IIoT Perception Layer. In each layer, we provide new emerging technologies that satisfy the key requirements of IoT and IIoT applications, such as, ThingsBoard IoT platform, OPNFV platform, Hyperledger Sawtooth, Digital twin, ONOS SDN controller, Mosquitto MQTT brokers, Modbus TCP/IP..
- We produce a highly unique IoT and IIoT cybersecurity dataset that represents the crucial IoT characteristics and heterogeneous network traffic. The IoT data are generated from various IoT devices (more than 10 types) such as Low-cost digital sensors for sensing temperature and humidity, Ultrasonic sensor, Water level detection sensor, pH Sensor Meter, Soil Moisture sensor, Heart Rate Sensor, Flame Sensor, etc.
- We identify and analyze fourteen attacks related to IoT and IIoT connectivity protocols, which are categorized into five threats, including, DoS/DDoS attacks, Information gathering, Man in the middle attacks, Injection attacks, and Malware attacks.
- We extract features obtained from different sources, including alerts, system resources, logs, network traffic, using two networks protocols analyzers, namely, the Zeek tool and TShark tool. Then, we propose new 61 features with high correlations from 1176 features found.
- We propose new processing and analyzing framework for our realistic cyber security dataset of IoT and IIoT applications, which is based on ten steps, including, 1) labeling for binary classification models, 2) labeling for multiclass classification models, 3) merging all CSV files, 4) applying the process of detecting and correcting, 5) dropping unnecessary flow features, 6) converting categorical variable, 7) splitting arrays or matrices into random train and test subsets, 8) encoding categorical features, 9) standardizing features, and 10) implementing the synthetic minority over-sampling technique.
- We provide a primary exploratory data analysis and evaluate the performance of machine learning approaches in both centralized and federated learning modes.
- We provide a complete review and analysis of the available existing datasets with Edge-IIoTset. The findings demonstrate the performance of our proposed platform in creating a new comprehensive realistic cyber security

TABLE 1. Available IoT and IIoT datasets for cyber security.

Dataset*	Year	Description	Features	ML Techniques	Testbed	IoT/IIoT Devices	Threats	Learning Approach		Traffic	
								Centralized	FL	IoT	IIoT
N-BaIoT [9]	2018	Built using 9 IoT devices for legitimate traffic and two botnets (BASHLITE and Mirai) for 10 attack types.	23	LOF, On Class SVM, and IF.	2 Layers	9 Types	10 Attacks	✓	✗	✓	✗
Bot-IoT [10]	2019	Built from IoT legitimate traffic as well as malicious traffic generated by botnets on IoT-specific networks.	46	RNN, SVM, and LSTM.	VM	Simulated	8 Attacks	✓	✗	✓	✗
MQTTset [11]	2020	Built from MQTT protocol traffic and a variety of attack streams associated with IoT devices that leverage it.	33	NN, DT, RF, NB, MP, and GB.	2 Layers	8 Types	5 Attacks	✓	✗	✓	✗
Federated TON_IoT [3]	2020	Consists of three different data types, namely: IoT service telemetry, OSs logs, and network traffic.	31	N/A	3 Layers	Simulated	9 Attacks	✗	✗	✓	✗
X-IIoTID [12]	2021	Consists of device agnostic data used in the context of ML/DL based IDS for both IoT and IIoT systems.	59	DT, NB, SVM, KNN, LR, DNN, and GRU.	3 Layers	N/A	18 Attacks	✓	✗	✓	✓
WUSTL-IIOT-2021 [2]	2021	Created using legitimate and malicious data generated by various IIoT and industrial devices to mimic an actual industrial application.	41	LR, KNN, SVM, NB, RF, DT, and ANN.	4 Layers	5 Types	4 Attacks	✓	✗	✓	✓
Ours	/	Cyber security dataset of IoT and IIoT applications, basedon realistic testbed, for evaluating machine learning-based intrusion detection systems	61	DT, RF, SVM, KNN, and DNN	7 Layers	+10 Types	14 Attacks	✓	✓	✓	✓

*There are other cyber security datasets used by security researchers for evaluating artificial intelligence-based security systems in different IoT and IIoT applications, such as ISCX, DARPA-2009, CICDS2017, UNSW-NB15, KDD99, and NSL-KDD [13]. However, these datasets are not simulated in a real-world IoT/IIoT environment.

dataset of IoT and IIoT applications and the superiority of the Edge-IIoTset dataset in comparison to existing ones.

The structure of the paper is organized as follows. In Section II, we provide a complete review and analysis of the available existing datasets with Edge-IIoTset. Section III presents our proposed IoT and IIoT testbed architecture. Section IV provides the description of Edge-IIoTset dataset. In Section V, we provide the extrapolated features and describe their different types. Section VI presents the experimental results of the proposed Edge-IIoTset dataset. Finally, Section VII concludes this paper.

II. RELATED AVAILABLE IoT AND IIoT DATASETS FOR CYBER SECURITY

Various datasets have been proposed by the community for IoT/IIoT cybersecurity in recent years [14]. This section presents a discussion about some of the most popular datasets that have been recently used for IoT/IIoT-based IDS developments. Tab. 1 provides a brief comparison between these datasets and ours.

A. MQTTset DATASET

Created by Vaccari *et al.* [11] as a way to train ML-based IDSs in the IoT context. The specific objective of the MQTTset is the focus on the MQTT protocol and the threats associated with IoT devices that use it. The lab environment established by the authors for generating the dataset consists of eight sensors and an MQTT broker. The sensor types deployed in two rooms are temperature, humidity, motion, CO-Gas, door lock, fan, smoke, and light sensors. The collection period corresponds to a time window of one week, generating more than 11 million network packets, with a more than 1 GB data size. The MQTTset is

TABLE 2. Hardware and Operating systems used in the creation of Edge-IIoTset dataset.

Name	Description	Specifications
Wireless Router	Routing IoT traffic from IoT devices to Edge servers	ZLT P21(GPRS, EDGE, WiFi, 4G)
Raspberry Pi 4 Model B	Used as an Edge server where the Mosquitto MQTT Server is installed	- 4G of RAM - 1.5 GHz 64-bit quad core ARM Cortex-A72 processor - On-board 802.11ac Wi-Fi
ESP32	Configure, program and connect IoT devices	- This microcontroller is equipped with WiFi and Bluetooth interfaces ideal for connected objects. Male and female side connectors allow the module to be plugged into a quick mounting plate.
Arduino Uno	Configure, program and connect IoT devices	- Operating Voltage of the Arduino is 5V - Flash Memory -32 KB - ATmega328P based Microcontroller - Digital input & output pins (PWM)-6 - Digital input and output pins-14 - Analog i/p pins are 6
Sensors and Actuators	Collecting information and sending them to Edge servers via Wireless Router	Please see Tab.4
Desktop Computers	Hardware used for writing codes and uploading them to the board Arduino	- Processor: Intel Core i5 (8th Gen) - Ram: 8 GB DDR4 RAM - Storage: 256 GB SSD
Smart phone	Some IoT devices are controlled through an app on a smartphone	OPPO A93 (GSM / HSPA / LTE/ Wi-Fi 802.11)
Smart TV	Controlling the IoT devices by connecting IoT devices with the Smart TV	LG Smart TV with wifi network connection
Kali Linux	Used for penetration testing, network security assessments, hacking IoT devices and Edge servers	Kali-Linux-2021.3-vbox-amd64
Windows	Used as a victim operating system	Windows 10 64 bits
Ubuntu	Used as a victim operating system	Ubuntu 21.10
Security Onion	A free and open Linux distribution for log management, threat hunting, and security monitoring	Security Onion 2 - version: 2.3.91
Raspberry Pi OS	A Debian-based operating system for Raspberry Pi	Debian version 11
Android	A mobile operating system for smartphone	Android version 11
ONOS SDN controller	Used for building SDN/NFV solutions	Version 2.6.0

comprised of both legitimate and malicious traffic. The version of MQTT is 3.1.1 with the authentication disabled. The dataset is composed of 33 features, including three related to TCP and 30 related to MQTT. The malicious traffic was generated by launching attacks against the MQTT broker. The attack vector used in the dataset include flooding DoS using the MQTT-malaria tool, MQTT publish flood using the IoT-Flock tool, Slow DoS against Internet of Things Environments (SlowITe), malformed data attack using the MQTTSA tool, and brute force authentication by using MQTTSA also. For the validation of the proposed dataset in terms of intrusion detection, the authors considered the following algorithms: neural network, Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Multilayer Perceptron (MP), and Gradient Boost (GB). For the multi-classification approach RF shower the best performance results with 99% accuracy and 99% F1-score, while MP showed the worst with 94% accuracy and 96% F1-score. However, the dataset only contains MQTT traffic, which means that there is no IIoT traffic such as Modbus protocol, which in turn makes this dataset not suitable for IIoT security applications.

B. N-BaIoT DATASET

Created by Meidan *et al.* [9] as a way to evaluate a proposed network-based anomaly detection scheme that retrieves behavioral snapshots out of the network, and leverages deep autoencoders for detecting abnormal network traffic originating from exploited IoT devices. The constructed lab environment is composed of nine IoT devices with the following types: doorbells, thermostats, baby monitors, security cameras, and a webcam. In addition to an access point, a sniffer host, and a C&C server. The total number of instances reported in the dataset is 7062606 from the nine IoT devices. The dataset contains a set of 23 features from five-time windows, consists of statistics of streams: weight, mean, std, radius, magnitude, cov, and pcc (approximated covariance between two streams). The normal traffic was captured right after the new installation, to ensure that no infected streams were injected. The malicious traffic consists of 10 attack types carried by 2 botnets namely BASHLITE and Mirai. The authors implemented an optimized deep autoencoder for the validation of the proposed method and dataset and conducted a comparison with three models, namely: Local Outlier Factor (LOF), One-Class SVM, and Isolation Forest (IF). For most devices, deep autoencoders have shown superiority in terms of TPR, FPR, and detection time, with a TPR of 100%, a mean FPR of 0.007 ± 0.01, and a time of 174 ± 212 ms. However, the dataset includes only malicious attacks from two botnets, with no IIoT traffic involved, making it impossible to detect other types of IoT attacks, such as MiTM, and not relevant to IIoT security applications.

TABLE 3. Open source tools used in the creation of Edge-IIoTset dataset.

Tool name	Description	Link
Node Red	A visual tool for wiring the Internet of Things	https://nodered.org/
Modbus of Node Red	The all in one Modbus TCP and Serial contribution package for Node-RED	https://flows.nodered.org/node/node-red-contrib-modbus
Zeek	An open-source software network analysis framework	https://zeek.org/
Wireshark	An open-source packet analyzer	https://www.wireshark.org/
netsniff-ng	A free Linux networking toolkit for network development and analysis, debugging, auditing or network reconnaissance	http://netsniff-ng.org/
Arduino IDE	The open-source Arduino Software (IDE) is used to write code and upload it to the board arduino	https://www.arduino.cc/en/software/
Spyder	An open-source cross-platform integrated development environment for scientific programming in the Python language	https://www.spyder-ide.org/
Eclipse Ditto	A tool that abstracts the device into a digital twin in an IoT environment	https://www.eclipse.org/ditto/
Eclipse Hono	A tool for connecting large numbers of IoT devices using remote service interfaces	https://www.eclipse.org/hono/
Virtual Network Computing	A tool used to remotely control computers using a graphical desktop-sharing system	https://www.realvnc.com/
pfSense	An open source firewall and router installed on a physical computer or a virtual machine	https://www.pfsense.org/
Hyperledger Sawtooth	Flexible and modular enterprise class product for building, implementing and operating blockchains technology.	https://sawtooth.hyperledger.org/
ThingsBoard	An IoT platform for the development, monitoring and scaling of IoT applications. It supports multiple deployment types, including cloud and fog.	https://thingsboard.io/
Mosquitto MQTT Broker	Highly adaptable and multi-platform MQTT protocol message broker	https://mosquitto.org/
Apache web server	Scalable, reliable and robust web server that offers HTTP based services in accordance with the current HTTP standards	https://httpd.apache.org/
OPNFV	A platform that makes it easy to develop and scale NFV elements throughout various open source ecosystems.	https://www.opnfv.org/
vsftpd	An FTP server for Unix-like systems. It is the default FTP server in many well-known Linux destros, including ubuntu.	https://security.appspot.com/vsftpd.html
OpenSSH	a set of network security utilities on the basis of the Secure Shell (SSH) protocol, to protect the communication channel over insecure networks	https://www.openssh.com/

C. BOT-IoT DATASET

Created by Koroniotis *et al.* [10] at the Research Cyber Range lab of UNSW Canberra using both real and simulated IoT network traffic, with the goal of detecting and identifying botnets on IoT-specific networks. The lab environment is constituted by three elements: a) Network services and platforms, including legitimate and malicious virtual machines (VMs), b) Simulated IoT-based smart-home, using the Node-red tool, and including traffic of simulated IoT devices, including thermostat, garage door, refrigerator, weather monitoring system, and lights, and c) Forensics analytics, using the Argus tool. The dataset consists of over 72 million records, with a size of 69.3 GB for the captured PCAP files, and 16.7 GB CSV for the extracted flow traffic. The protocols used in the dataset include TCP, UDP, ARP, ICMP, IGMP, and RARP. The reported features are of two types: real protocol parameters and generated flow features. The malicious traffic was generated using cyber-attacks originating from Kali Linux VMs, and including probing (port scanning and OS fingerprinting), DoS/DDoS, and information theft (data theft and keylogging). The dataset was tested under three ML and DL models, namely Recurrent Neural Network (RNN), Support Vector Machine (SVM), and Long-Short Term Memory (LSTM). SVM showed the best accuracy performance with 99%. However, there is only IoT data in the dataset, so there's no IIoT traffic, making it unsuitable for IIoT security.

D. FEDERATED TON_IoT DATASET

Created by Moustafa *et al.* [3] at the IoT lab of UNSW Canberra, by including federated data sources collected from three dataset types: a) IoT services telemetry, b) Operating systems, and c) Network traffic. The testbed is layered into three levels: a) Edge Layer: houses IoT and networking appliances, b) Fog Layer: houses VMs and gateways, and c) Cloud Layer: consists of services, like data analytic. The dataset is composed of both normal and attacks traffic. The Windows 7 dataset contains 10000/5980 normal/attack records, while the Windows 10 dataset contains 10000/11104 normal/attack records. The dataset includes nine attack categories, namely DoS/DDoS, scanning, ransomware, backdoor, injection, XSS, password, and Man-In-The-Middle (MITM) attacks. The authors reported the correlation analysis of the selected features. The correlation matrix was adjusted to pick the most correlated features with a threshold value greater than or equal to 0.85. However, the dataset does not contain IIoT traffic, nor does it provide intrusion assessment using different machine learning techniques with the proposed dataset to validate it.

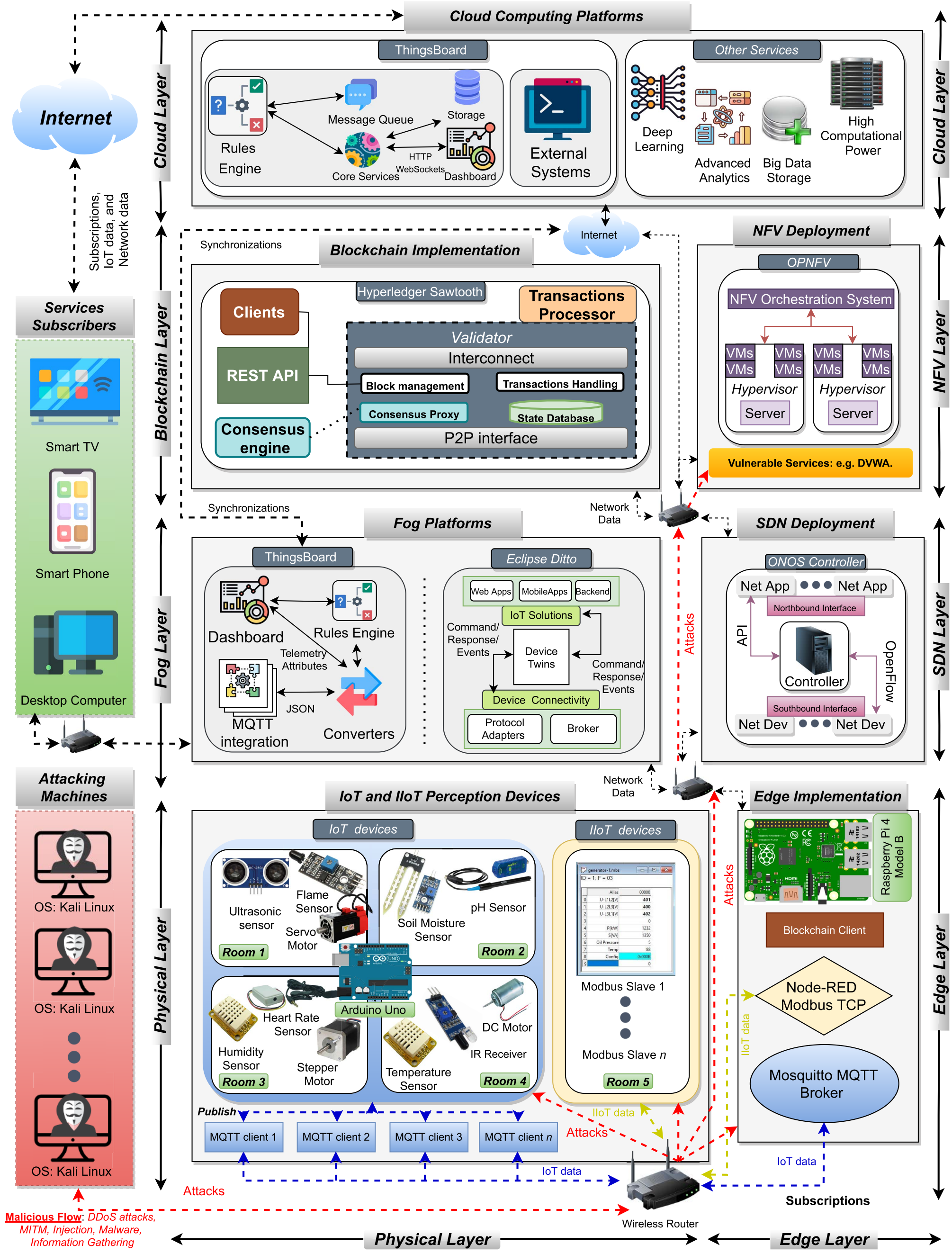


FIGURE 1. Our testbed architecture.

E. X-IIoTID DATASET

Generated by Al-Hawawreh *et al.* [12] at the University of New South Wales (UNSW) in Canberra, as connectivity and device-agnostic dataset for evaluating and training ML/DL-based IDSs for IoT and IIoT systems. The lab's architecture is based on the Industrial Internet Reference Architecture (IIRA) model. The architecture used consists of three levels, namely: the edge level, the platform level, and the enterprise level, where various industrial and IoT devices and protocols, cloud services, and attack machines are deployed. The datasets contain 421,417 normal records, 399,417 malicious records, and 59 features, collected from network traffic, device resources, and device/alert logs. The protocol used includes Modbus, MQTT, TCP, CoAP, and SMTP. The malicious records were generated using three different frameworks, namely CKC, MALC, and ATT&CK. The testbed attack process consists of the following attack stages: a) Reconnaissance, b) weaponization, c) exploitation, d) lateral Movement, e) Command and Control, f) exfiltration, g) tampering, h) crypto-Ransomware, and i) Ransom DoS.

The dataset was evaluated using the following models: DT, NB, SVM, K-nearest Neighbor (KNN), Logistic Regression (LR), Deep Neural Network (DNN), and Gated Recurrent Unit (GRU). DT obtains the highest performance of all algorithms, with 99.54% accuracy for binary classification, and 99.49% for multi-classification. However, the dataset only utilizes centralized learning approaches for providing intrusion detection evaluations with the proposed dataset to validate it. Using federated learning is essential in different situations within IoT/IIoT environments to address privacy, network, and storage issues [17].

F. WUSTL-IIOT-2021 DATASET

Created by Zolanvari *et al.* [2], as a cybersecurity-targeted network-driven dataset of IIoT applications, by modeling and emulating actual industrial systems in the real world. the architecture implemented includes various IIoT sensors and actuators, HMI, PLC, logger, and alarming device, for simulating real-life industrial applications. The data set consists of 2.7 GB of data, collected in approximately 53 hours, with a total of 1,194,464 observations, including 1,107,448 for benign samples and 87,016 for malicious samples. The dataset contains 41 features selected based on the variation of their values during the attack phases. The attacks used in the testbed include command injection, DoS, reconnaissance, and backdoor. The model used to evaluate the generated dataset are LR, KNN, SVM, NB, RF, DT, and ANN. The RF model scored the best accuracy with 99.99%, and NB showed the least accuracy with 97.48% for binary classification. However, the dataset only contains data from an IIoT architected environment, with no traffic, data, or attacks from IoT-based devices. Therefore, this dataset is not suitable for evaluating IoT-based IDSs.

G. OTHER DATASETS

EMNIST dataset [23] is considered as a standard benchmark for AI-based computer vision systems. The dataset consists of handwritten character digits derived from the special NIST 19 database. Federated EMNIST dataset (FEMNIST) is the federated version of EMNIST, which partitions the dataset into individual clients with each client being assigned a corresponding number/character set of records in EMNIST. However, these datasets do not contain IoT or IIoT network traffic, so IoT/IIoT-based based IDS cannot be trained on them.

III. PROPOSED IoT AND IIoT TESTBED ARCHITECTURE

Given the limited number of IoT/IIoT datasets available for the cyber security sector, in which researchers typically rely on proprietary or open-source datasets that are not field-specific. In our work, we designed a realistic testbed that closely mirrors a real-world IoT/IIoT environment, and conducted realistic cyberattacks against it, to acquire real-world datasets with both legitimate and malicious traffic. The testbed consists of seven interconnected layers, namely: cloud computing layer, NFV layer, Blockchain layer, fog layer, SDN layer, edge layer, and IoT/IIoT perception layer, as shown in Fig. 1. Tab. 2 provides a list of the equipment and associated operating systems used for creating our dataset. We used open-source software to build our testbed as presented in Tab. 3 so that it can be easily re-used and validated by the research community. This section provides a detailed description of each layer.

A. CLOUD COMPUTING LAYER

This layer is not physically deployed in the lab, however, it acts as a provider of various services and resources such as IoT platforms, data storage, and computing power over the Internet. Cloud-based data storage, processing, visualization, and device management are mandatory operations for almost all IoT/IIoT-based applications. We have used the ThingsBoard IoT platform [24], since it supports a variety of IoT protocols, including MQTT, CoAP, and HTTP, for device connectivity. The platform also supports the capability of creating rich custom dashboards for real-time data visualization and remote device control, which is relevant to most IoT use cases. Every access to this layer will be via the Internet, as opposed to the other layers, where access is done locally through wireless routers.

B. NETWORK FUNCTIONS VIRTUALIZATION LAYER

NFV abstracts network functions to reduce overall costs and speed service deployment by separating network functions from their dedicated equipment by deploying them on virtual servers. This brings significant advantages, including savings in power usage, lower equipment and maintenance expenses, smoother upgrades, and better assets lifestyles. OPNFV is an industry-supported open-source NFV Infrastructure (NFVI) platform [25], that allows builds to be rolled out and tested on a range of different hardware settings. OPNFV combines various components, such as OpenStack, Kubernetes, and OpenDaylight, to create an end-to-end platform for computing, storage, and networking virtualization. Vulnerable services and applications are deployed in the layer, including Damn Vulnerable Web Application (DVWA). Attacks against these services and applications are discussed in detail in the following sections.

C. BLOCKCHAIN NETWORK LAYER

The applications of blockchain extend significantly beyond the realm of crypto-currencies, through its potential to create more transparency and equity while saving companies time and money. In an effort to build a sophisticated real-world testbed, we've included an enterprise-level blockchain platform called Hyperledger Sawtooth [26], which enables both distributed applications and ledger networks. In addition, Sawtooth also offers a high degree of modularity, allowing companies to make the most appropriate strategic decisions and let applications choose the appropriate consensus, access, and transaction protocols that suit the customer's particular needs. The framework supports making design decisions within the transaction processor, permitting several types of

TABLE 4. IoT sensors and actuators adopted in the creation of Edge-IIoTset dataset.

IoT Device name	Description	IIoT/IoT application	Type	Features and Specifications	Pin Configuration
Sound detection sensor	The sound sensor is one type of module used to notice the sound.	- Security system for office or home - Home automation - Ambient sound recognition	LM393 Sound Detection Sensor Module	- The voltage gain 26 dB (V=6V, f=1kHz) - The sensitivity of the microphone (1kHz) is 52 to 48 dB	- Vcc pin powers the module (+5V) - Power supply ground - Digital output pin - Analog output pin
Low-cost digital sensor for sensing temperature	Consists of a thermistor for sensing temperature	- IoT agricultural - Heating, ventilation and air conditioning systems - Internet of Vehicles	DHT11 Sensor	- The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy	- VCC - GND - Digital output pin
Low-cost digital sensor for sensing humidity	Consists of a capacitive humidity sensing element	- IoT agricultural - Heating, ventilation and air conditioning systems - Internet of Vehicles	DHT11 Sensor	- The humidity range of DHT 11 is from 20 to 80% with 5% accuracy	- VCC - GND - Digital output pin
Ultrasonic sensor	Find out the distance of the object from the sensor	- IoT agricultural - Internet of Vehicles - Internet of Drones - IoT-based Smart Monitoring	HC-SR04 Ultrasonic Sensor	- The input pulse width of trigger is10uS - Measuring angle is 30 degrees - The distance range is 2cm to 800 cm	- VCC pin - Trig pin - Echo pin - GND pin
Water level detection sensor	The sensor produces an output voltage according to the resistance, which is designed for water detection	- IoT agricultural - Water distribution systems -Smart Irrigation systems	Water level detection sensor module	- An electronic brick connector - A 1 MΩ resistor, and several lines of bare conducting wires	- VCC - GND - Analog output pin
pH Sensor Meter	It can be used in a variety of PH measurements	- IoT agricultural - Water distribution systems	pH-sensor PH-4502C	- Measuring Range :0-14PH - Measuring Temperature :0-60 ? - Accuracy : ± 0.1pH	- VCC - GND - Analog output pin
Soil Moisture sensor	By measuring the charge and discharge time, the sensor can determine how moisty the earth is	- IoT agricultural - Smart crops monitoring - Smart irrigation	Capacitive moisture sensor V. 1.2	- Input Voltage: 3.3–5V - Output Voltage: 0–4.2V - Input Current: 35mA - Output Signal: both analog and digital	- VCC - GND - Analog output pin
Heart Rate Sensor	- A small ambient light photo sensor with a microchip's MCP6001 Op-Amp and a bunch of resistors and capacitors	- Internet of Healthcare Things - IoT-based livestock health monitoring	Heart Rate Sensor	- The module operates from a 3.3 to 5V DC Voltage supply with an operating current of < 4mA	- VCC - GND - Analog output pin
Flame Sensor	Detect fire and provide a HIGH signal upon the detection	- IoT agricultural - Internet of Healthcare Things - Internet of Vehicles - Internet of Drones	Flame Sensor based on the YG1006 sensor	Detect infrared light with a wavelength ranging from 700nm to 1000nm and its detection angle is about 60°	- VCC - GND - Digital output pin
Servo Motor	Geared motor capable of turning 180 degrees, and activated by sending electrical impulses	- IoT/IIoT automation and control - Agricultural IoT - Internet of Drones. - Internet of Vehicles	Servo SG90	- Working Voltage: 3.5-6V - The Operating Speed is 0.12 sec/ 60° under 4.8V, and 0.10 sec/60° under 6.0V	- GND - Power pin - Signal pin
Stepper Motor	Electric direct current motor that splits a complete rotation into a number of equal steps.	- IoT/IIoT automation and control - IoT Medical Imaging. - Internet of Drones	Stepper motor 28BYJ-48 and the ULN2003 driver module	- The Motor's Rated voltage is 5V DC - The Motor Frequency is 100Hz -DC resistance: 50Ω ± 7% - 25°C	Coil 1 Coil 2 Coil 3 Coil 4 -Common
DC Motor	Electric motors that transform direct current electrical energy into mechanical energy.	- Internet of Drones - IoT/IIoT automation and control - Internet of Vehicles - Agricultural IoT	DC motor and the L293D chip	The input voltage is 6.5-9v (DC) -The output voltage is 3.3V/5v - The maximum output current is 700 mA	DC Motor: (+) and (-) pins L293D chip: - Enable 1, In1, out 1, ov, ov, out 2, in 2, and +Vmotor pins
IR Receiver Sensor	Used for remote control detection	- IoT agricultural - Internet of Vehicles - Internet of Drones - IoT-based Smart Monitoring	IR Receiver Diode - TSOP38238	- The peak LED color is 940 nm - The peak frequency detection is at 38 KHz	Signal, Voltage and Ground

applications (IoT and IIoT applications) to operate within a single blockchain network instance. Individual applications can set up custom transaction processors tailored to their specific business requirements.

D. FOG COMPUTING LAYER

This layer acts as a mediator between the edge and the cloud layers for various purposes, including determining the relevance of data from the edge for relieving pressure on the network and the cloud by selecting the most important data. ThingsBoard is used as an IoT fog platform since it supports fog deployment, and it will also be responsible for synchronizing the data with the cloud instance. We have also deployed a digital twin for our testbed using Eclipse Ditto [27], in order to create a virtual model designed to accurately reflect the implementation of our cyber-physical testbed in the real world. Since our testbed is equipped with various sensors and actuators, it produces data related to many aspects of real-world physical object performance, such as temperature, PH, light, etc. Once the data is generated, it is transmitted to a virtual model of the physical object, which is then fed into a processing system that updates the digital copy.

E. SOFTWARE-DEFINED NETWORKING LAYER

This layer employs SDN technology, which is a sophisticated network management concept that enables dynamically efficient, programmatic configuration across the network to improve network performance and control. SDN is designed to overcome conventional networks by localizing the network logic into a single component and separating the transmission of packets from the routing operation. We used the ONOS SDN controller [28] for this layer. ONOS is a flexible, scalable, distributed SDN controller that makes it easy to administer, deploy, and set up new network components, such as network applications. It also supports real-time network control and configuration with user-friendly programmatic interfaces.

F. EDGE COMPUTING LAYER

Instead of having everything exported to the cloud for processing and analysis, this layer is positioned much closer to the data sources, by bringing the calculation features to the edge of the network, and handling IoT/IIoT data far away from the cloud nodes, near the edge of the network. By doing so, it allows data to be properly prioritized locally, thereby minimizing traffic flow on its way back to the cloud, making the Fog layer less complex with fewer possible points of failure, reducing bandwidth and cloud resources, and optimizing network latency. To accomplish this, specifically for IoT data, we installed various Mosquitto MQTT brokers [29]- an open-source message broker that implements the MQTT protocol - on several Raspberry Pi boards. In the case of IIoT data, we used Node-RED Modbus TCP [30], a Modbus master/slave creation tool intended to assist Modbus slave device builders to test and simulate the Modbus protocol.

G. IoT AND IIoT PERCEPTION LAYER

The perception layer or physical layer is equipped with a range of sensors that detect and gather environmental information, including the detection of specific physical parameters and/or the recognition of other types of information in the environment. It also includes actuators that act on the environment when certain conditions are met. Modbus slaves also belong to this layer and they receive requests from the master and send back replies. Tab. 4 provides a detailed description of each and every IoT sensor and actuator used in the testbed. The table includes the type of devices used, a brief description of the operation the device is to perform, the different application modes in which the device can be deployed, the product reference number, the features and specifications of the device such as the voltage, and the pin configuration used with the Arduino Uno board.

H. EXTERNAL ENTITIES

While the layers discussed above represent a sophisticated IoT/IIoT-based system, in this part the components represent the entities that interact with the system either with good or bad intentions. Specifically, we consider two entities: the service subscribers and the attacking machines.

1) SERVICES SUBSCRIBERS

These are the devices that subscribe to telemetry (IoT) and IIoT data from the various services deployed in the system. We consider smart TV, smartphone, and desktop computer usage. When such a device has made a subscription to a specific type of data, say for example the *ultrasonic sensor* located in *room 1*, whenever a change occurs, the IoT platform receives the change and notifies the subscriber in question with the change in real-time.

2) ATTACKING MACHINES

These entities are the malicious traffic generators for our dataset, as they use various attack software, tools, and scripts that are installed on these entities. A complete list of the attacks and their techniques is presented in the next section.

IV. DESCRIPTION OF EDGE-IIoTset DATASET

In this section, we thoroughly explore the various steps we took to generate our dataset [1]. We initially provide a discussion of the proposed generation framework, followed by a description of malicious traffic management using multiple attacks, approaches, and tools.

A. METHODOLOGY OF CREATING THE EDGE-IIoTset DATASET

As presented in Fig. 2, the methodology of creating the Edge-IIoTset Dataset is organized in the following sevens steps:

1) SETUP AND CONFIGURATION OF NETWORK EQUIPMENT

We started with the installation of the software and hardware equipment, which are presented in tables 2, 3, and 4.

TABLE 5. The list of attack scenarios included in Edge-IIoTset dataset.

Attack category	Attack type	IoT vulnerabilities	Tools	Attackers (@IP)
DoS/DDoS attacks	TCP SYN Flood DDoS attack	Make the victim's IoT edge server unavailable to legitimate requests	Sending manipulated SYN packets using the tool hping3-based python script	207.192.25.133 94.196.109.185 133.149.252.77 220.146.94.148
	UDP flood DDoS attack	Overwhelm the processing and response capabilities of IoT devices	Sending manipulated UDP packets using the tool hping3-based python script	190.123.219.128 16.226.184.201 153.125.214.15 91.184.12.91
	HTTP flood DDoS attack	Exploits seemingly-legitimate HTTP GET or POST requests to attack IoT application	Use 200000 connections with GET requests using the slowhttptest tool	192.168.0.170 216.58.198.74
	ICMP flood DDoS attack	The IoT edge servers become inaccessible to normal traffic By flooding them with request packets (i.e., with ICMP echo-requests (pings))	Sending manipulated ICMP packets using the tool hping3-based python script	213.117.18.213 183.223.100.122 166.153.227.121 49.81.59.152 227.117.33.125
Information gathering	Port Scanning	Discover open doors or weak points in the edge-based IoT network	Discover active hosts using the Nmap and Netcat tools	192.168.0.170
	OS Fingerprinting	Analyzing IoT data packets to spot the weakness of IoT devices as well as Edge servers	An active operating system fingerprinting tool, named xprobe2	192.168.0.170
	Vulnerability scanning attack	Identifying IoT network security vulnerabilities	A web server scanner tool, named Nikto, for performing comprehensive tests against web servers	192.168.0.170 142.250.200.205 172.217.19.35 142.250.201.10
Man in the middle attacks	DNS Spoofing attack	The interception of communications between IoT devices and a DNS server	Sniffing and spoofing using Ettercap tool	192.168.0.101 192.168.0.152 172.217.19.35 192.168.0.170
	ARP Spoofing attack	Linking an attacker's MAC address with the IP address of an IoT device or Edge server	Sniffing and spoofing using Ettercap tool	192.168.0.101 192.168.0.152 172.217.19.35 192.168.0.170
Injection attacks	Cross-site Scripting (XSS) attack	Send a malicious script to an unsuspecting user, which can access sensitive information, session tokens, cookies, etc.	Detect, exploit and report XSS vulnerabilities using xsser tool in a PHP/MySQL web applications (DVMA application)	192.168.0.170 172.217.19.42 104.16.87.20
	SQL Injection	(Read/Insert/Update/Delete) sensitive data from the IoT database by the injection of a SQL query	Detecting and exploiting SQL injection flaws using sqlmap tool	192.168.0.170
	Uploading attack	Uploading files that contain malwares' command and control data	Creating php backdoor using Metasploit framework and uploading through a PHP/MySQL web application (e.g., Damn Vulnerable Web App (DVWA))	192.168.0.170
Malware attacks	Backdoor attack	Install backdoors to take control of vulnerable IoT network components	Creating python script backdoor using Metasploit framework and then transferring it using curl tool	192.168.0.170
	Password cracking attack	Identify an unknown or forgotten password to an IoT device in order to obtain unauthorized access to IoT resources	The CeWL tool is used as a ruby app for creating a list of words (password crackers) and email addresses (usernames)	192.168.0.170
	Ransomware attack	Publish or blocks access to IoT data or an IoT device system by encrypting it, until the victim pays a ransom fee to the attacker	After applying the Backdoor attack, the OpenSSL cryptography toolkit is used for creating RSA public/private keys and encrypting and decrypting victim files	192.168.0.170

More specifically, we configured these tools for each corresponding layer, including, cloud computing layer, NFV layer, Blockchain layer, fog layer, SDN layer, edge layer, and IoT/IIoT perception layer.

2) THREAT AND ATTACK MODELING

This step consists of modeling the attacks and threats against the IoT and IIoT applications. More accurately, we identified and analyzed fourteen attacks as presented 5, which are categorized into five threats, including, DoS/DDoS attacks, Information gathering, Man in the middle attacks, Injection attacks, and Malware attacks. The DoS/DDoS attacks make the victim's IoT edge server unavailable to legitimate requests by sending manipulated packets, which include four attacks, namely, TCP SYN Flood DDoS attack, UDP flood DDoS attack, HTTP flood DDoS attack, and ICMP flood

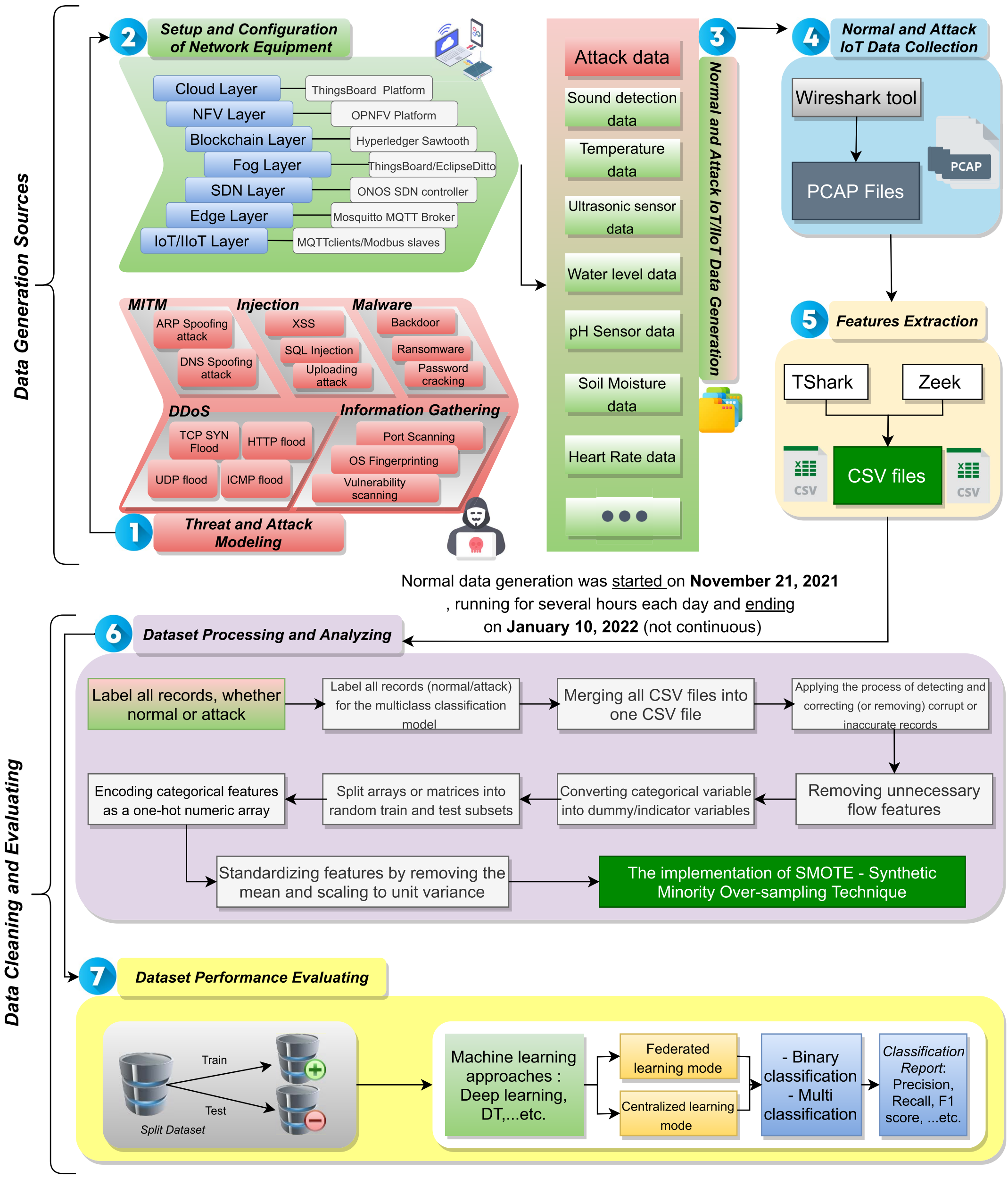

Data Generation Sources
2 Setup and Configuration of Network Equipment
Cloud Layer
ThingsBoard Platform
NFV Layer
OPNFV Platform
Blockchain Layer
Hyperledger Sawtooth
Fog Layer
ThingsBoard/EclipseDitto
SDN Layer
ONOS SDN controller
Edge Layer
Mosquitto MQTT Broker
IoT/IIoT Layer
MQTTclients/Modbus slaves
MITM
ARP Spoofing attack
DNS Spoofing attack
Injection
XSS
SQL Injection
Uploading attack
Malware
Backdoor
Ransomware
Password cracking
DDoS
TCP SYN Flood
HTTP flood
UDP flood
ICMP flood
Information Gathering
Port Scanning
OS Fingerprinting
Vulnerability scanning
1 Threat and Attack Modeling
Attack data
Sound detection data
Temperature data
Ultrasonic sensor data
Water level data
pH Sensor data
Soil Moisture data
Heart Rate data
●●●
3 Normal and Attack IoT/IIoT Data Generation
4 Normal and Attack IoT Data Collection
Wireshark tool
PCAP Files
PCAP
5 Features Extraction
TShark
Zeek
CSV files
CSV
CSV
Normal data generation was started on November 21, 2021
, running for several hours each day and ending
on January 10, 2022 (not continuous)
Data Cleaning and Evaluating
6 Dataset Processing and Analyzing
Label all records, whether normal or attack
Label all records (normal/attack) for the multiclass classification model
Merging all CSV files into one CSV file
Applying the process of detecting and correcting (or removing) corrupt or inaccurate records
Removing unnecessary flow features
Converting categorical variable into dummy/indicator variables
Split arrays or matrices into random train and test subsets
Encoding categorical features as a one-hot numeric array
Standardizing features by removing the mean and scaling to unit variance
The implementation of SMOTE - Synthetic Minority Over-sampling Technique
7 Dataset Performance Evaluating
Train
Test
Split Dataset
Machine learning approaches : Deep learning, DT,...etc.
Federated learning mode
Centralized learning mode
- Binary classification
- Multi classification
Classification Report: Precision, Recall, F1 score, ...etc.

FIGURE 2. The proposed dataset generation framework.

DDoS attack. The Information gathering consists of analyzing IoT data packets to spot the weakness of IoT devices as well as Edge servers, which include three attacks, namely, Port Scanning, OS Fingerprinting, and Vulnerability scanning attack. The man in the middle attacks consists of the interception of communications between IoT devices and edge servers, which include two attacks, namely, ARP Spoofing attack and DNS Spoofing attack. The injection attacks consist of sending a malicious script to an unsuspecting user, which can access sensitive information, session tokens, cookies, etc. Finally, the malware attacks consist of installing backdoors to take control of vulnerable IoT network components, which include three attacks, namely, Backdoor attack, Password cracking attack, and Ransomware attack. Tab. 5 provides the list of attack scenarios included in Edge-IIoTset dataset.

3) NORMAL AND ATTACK IoT DATA GENERATION

In this phase, we generated IoT data from different components (i.e., IoT devices, Edge servers, SDN controller, *Mosquitto* MQTT brokers, etc.), as well as we launched the attacks against these components. The time period for normal data generation was started on *November 21, 2021*, running for several hours each day and ending on *January 10, 2022* (not continuous). Moreover, the generated attack data experiments were performed at different hours and days from *November 21, 2021*, to *January 10, 2022*, where each attack experiment was conducted multiple times to generate more records.

4) NORMAL AND ATTACK IoT DATA COLLECTION

This phase consists of capturing packet data from the IoT network using the Wireshark tool and storing it in a PCAP file format. Tab. 6 presents statistics of normal instances included in Edge-IIoTset dataset with PCAP files size. Traffic capture is done on a short-term period, a maximum of 3 hours of continuous collection. We configure the tool to collect all PCAP files from all the edge server interfaces (i.e., Raspberry Pi 4 Model B).

5) FEATURE EXTRACTION

This phase is focused on extracting the features from PCAP using two networks protocols analyzers, namely, the *Zeek* tool and *TShark* tool, and then storing it in a CSV file format for further processing. We identified and selected 61 features with a high correlation from 1176 found features.

6) DATASET PROCESSING AND ANALYZING

This phase is focused on processing and analyzing the Edge-IIoTset dataset. Specifically, we applied the following steps:

- *Step 1*: We added a new label, named Attack_label, in order to label all records, whether normal or attack. The Attack_label contains 0 or 1, which is used for the binary classification model (i.e., 0 indicates normal and 1 indicates attacks).
- *Step 2*: We added a new label, named Attack_type, which presents the attack categories, for the multiclass classification model.
- *Step 3*: We merged all CSV files into one CSV file.
- *Step 4*: We applied the process of detecting and correcting (or removing) corrupt or inaccurate records from the Edge-IIoTset dataset. Specifically, we removed duplicates and missing values such as NAN (Not A Number) or 'INF' (Infinite Value).
- *Step 5*: We removed unnecessary flow features such as IP addresses, ports, timestamp and payload information.
- *Step 6*: We applied the *pandas.get_dummies* package for converting categorical variable into dummy/indicator variables.
- *Step 7*: We used *train_test_split* from the *sklearn.model_selection* package for split arrays or matrices into random train and test subsets.
- *Step 8*: We used *OneHotEncoder* from the *sklearn.preprocessing* package for encoding categorical features as a one-hot numeric array.
- *Step 9*: We applied *StandardScaler* from the *sklearn.preprocessing* package for standardizing features by removing the mean and scaling to unit variance.
- *Step 10*: We applied the *SMOTE* class from the *imblearn.over_sampling* for the implementation of SMOTE - Synthetic Minority Over-sampling Technique.

7) DATASET PERFORMANCE EVALUATING

This phase is particularly focused on evaluating the performance of machine learning approaches in both centralized and federated learning modes. More particularly, we used the following machine learning approaches: RandomForest, Support Vector Machine (SVM), Decision Tree (DT), XGBoosT, as well as the most popular Deep Neural Network (DNN).

B. ATTACKS IN EDGE-IIoTset DATASET

The quality and diversity of legitimate entries in a dataset are critical for building the normal behavioral profile of a system. Additionally, malicious entries are essential for security solutions to recognize not only the precise attack patterns but also to identify new ones.

1) DoS/DDoS ATTACKS

In these attack categories, the attackers tend to deny the services from legitimate users, either solely or in a distributed fashion. We consider four of the most commonly used techniques, namely: TCP SYN Flood, UDP flood, HTTP flood, and ICMP flood.

- *TCP SYN Flood DDoS attack*: This is a version of a distributed denial of service (DDoS) attack that takes the exploitation of a normal three-way TCP handshake to use energy on the affected server and disable it completely. With SYN flood DDoS, the attacker essentially forwards requests for TCP connections more quickly in order to process them than the targeted machine can handle, which causes saturation of the IoT network. Once an IoT device and an Edge server have established a regular TCP "three-way handshake," the IoT device initiates the process of requesting the connection by sending an SYN (synchronization) message to the Edge server. The Edge server then acknowledges by returning an SYN-ACK (synchronization-acknowledgment) message to the IoT device. The IoT device answers with an ACK (acknowledgment) message and the connection is established. The offensive systems with the following IP addresses: 207.192.25.133, 94.196.109.185, 133.149.252.77, and 220.146.94.148 were used to send manipulated SYN packets using the tool *hping3*-based python script.
- *UDP flood DDoS attack*: This is a type of denial of service attack in which a high volume of User Datagram Protocol (UDP) packets are transmitted to a targeted Edge server in order to overwhelm the processing and response capabilities of that device. When each UDP packet is first received by the Edge server, it proceeds through a series of stages to address the request, while using the edge server's resources in the process. As UDP packets are delivered, each one will contain

TABLE 6. Statistics of normal instances included in Edge-IIoTset dataset.

N°	Edge Server (@IP)	IoT node sensor (@IP)	Access point (@IP)	IoT Device type	MQTT Topic	Data Profile	Data type	Data range	Records (PCAP size)	Records (CSV size)
1	192.168.0.128	192.168.0.101	192.168.0.1	DHT11 Sensor	Temperature_ and _ Humidity	Temperature and Humidity	Periodic	The temperature range is from 0 to 50 degrees Celsius while the humidity range is from 20 to 80%	- 1629749 records - 329 Mo	- 1629749 records - 879 Mo
2	192.168.1.128	192.168.1.101	192.168.1.1	HC-SR04 Ultrasonic Sensor	Distance	Distance of the object from the sensor	Periodic	Generating the high-frequency-sound waves (around 40kHz) for measuring distance	- 1143948 records - 81,4 Mo	- 1143540 records - 349 Mo
3	192.168.2.116	192.168.2.194	192.168.2.1	pH-sensor PH-4502C	phValue	Measure the ph of water	Periodic	Detection Range: 0 14. It is determined from the amount of free hydrogen ions (H +) contained in the substance	- 759665 records - 53,9 Mo	- 746908 records - 226 Mo
4	192.168.3.12	192.168.3.18	192.168.3.1	Heart Rate Sensor	Heart_Rate	Monitor the heart rate using pulse sensor	Periodic	The data value can range from 0-1024	- 170435 records -12,2 Mo	- 165319 records 50,1 Mo
5	192.168.4.30	192.168.4.73	192.168.4.1	Water sensor	Water_Level	Perceive the depth of water	Periodic	Width of detection: 40mm×16mm	- 2302497 records -164 Mo	- 2295288 records - 697 Mo
6	192.168.5.46	192.168.5.47	192.168.5.1	IR (Infrared) Receiver Sensor	IR_Receiver	Remote control detection for smart devices	Random	"POWER", "FUNC/STOP", "VOL+", "FAST BACK", "PAUSE", "FAST FORWARD" etc.	- 1308238 records -92,4 Mo	- 1307778 records - 393 Mo
7	192.168.6.100	192.168.6.56	192.168.6.1	LM393 Sound Detection Sensor	Sound_Sensor	Detects sound waves and converting it to electrical signals	Random	Analog output signal	- 1513318 records -106 Mo	- 1512883 records - 456 Mo
8	192.168.7.55	192.168.7.62	192.168.7.1	Flame Sensor based on the G1006 sensor	Flame_Sensor	Used for short range fire detection	Random	"Close Fire" , "Distant Fire" , "No Fire"	- 1070510 records -76 Mo	- 1070196 records - 327 Mo
9	192.168.0.128 192.168.7.55	192.168.0.101 192.168.7.62	192.168.0.1 192.168.7.1	Modbus/TCP server	Modbus_topic	Read and write to registers	Random	- Write Multiple Registers - Read Holding Registers - Input Register - Single Holding Register - Scattered Register Read - Read Device Identification	- 159514 records - 20 Mo	- 159502 records - 64,7 Mo
10	192.168.8.104	192.168.8.163	192.168.8.1	Soil Moisture Sensor v1.2	Soil_Moisture	Display the soil moisture value in percentage (%)	Random	The soil moisture range is from 1% to 100%	- 1200751 records - 85,6 Mo	- 1192777 records - 363 Mo

the IP address of the source device. During this kind of DDoS attack, an attacker will typically not use his real IP address but will impersonate the source IP address of the UDP packets instead, preventing the attacker's real place from being revealed and possibly flooded with the target server's response packets. The offensive systems with the following IP addresses: 190.123.219.128, 16.226.184.201, 153.125.214.15, and 91.184.12.91 were used to send manipulated UDP packets using the tool *hping3*-based python script.

- *HTTP flood DDoS attack*: This is a type of distributed denial of service (DDoS) attack that is intended to flood a particular target server with HTTP queries. After the target has been flooded with demands and is incapable of serving normal traffic, a denial of service attack will take place for further demands from actual users. Through the use of multiple malware-infected devices, attackers will employ or commonly build botnets to ensure that they reach the maximum effect of their offense. Two different types of HTTP flooding attacks are available, including, HTTP POST attack and HTTP GET attack. The offensive systems with the following IP addresses: 192.168.0.170 and 216.58.198.74 were used to apply 200000 connections with GET requests using the slowhttptest tool.
- *ICMP flood DDoS attack*: An Internet Control Message Protocol (ICMP) flood DDoS attack is a popular denial of service (DoS) attack where an attacker tries to

flood a targeted device through ICMP echo queries (pings). Technically, ICMP echo query and echo reply packets are employed to ping a network device to help diagnose the state of device health and connection between the source and the destination device. Flooding the destination with query packets, the network is constrained to reply with an identical number of reply packets. This makes the destination unavailable to regular network traffic. The offensive systems with the following IP addresses: 213.117.18.213, 183.223.100.122, 166.153.227.121, 49.81.59.152, and 227.117.33.125 were used to send manipulated ICMP packets using the tool *hping3*-based python script.

2) INFORMATION GATHERING

Obtaining intelligence about the targeted victim is always the first step in any successful attack. In our work, we consider three important steps that malicious actors generally do as a part of the information gathering stage, namely port scanning, OS fingerprinting, and vulnerability scanning.

- *Port Scanning*: The ports of IoT devices connected to a network are automatically scanned. The purpose is to discover which ports are open, closed or which of them have a security protocol. According to this analysis, intruders can obtain the composition of a network's architecture, the operating system, active security devices like firewalls, etc. This attack provides an easy access point for cyber-attackers. Once they manage to penetrate a network via port scanning, they will be able to extract sensitive information such as personal data, access to passwords, etc. The offensive systems with the following IP address: 192.168.0.170, were used to discover active hosts using the *Nmap* and *Netcat* tools.
- *OS Fingerprinting*: Once an attacker can identify the operating system (OS) type of a targeted device, he can then attack the vulnerabilities contained in that operating platform. Operating system fingerprinting is used by both attackers and security professionals to effectively and efficiently map remote networks, and to identify exploitable vulnerabilities. In addition, this attack operates only for packages with a TCP connection that has an ACK, SYN/ACK, and SYN. The offensive systems with the following IP address: 192.168.0.170 were used to apply an active operating system fingerprinting tool, named *xprobe2*.
- *Vulnerability scanning attack*: This is an automated procedure for conducting proactive detection of application and network vulnerabilities. Vulnerability scanning is usually conducted by attackers attempting to discover potential entry points into the network. This type of attack can be categorized based on three categories, including, environmental vulnerability scans, internal vulnerability scans, and external vulnerability scans. Specifically, the external vulnerability scans consist of scanning applications that are accessed by external users. The internal vulnerability scans consist of scanning and identifying the vulnerabilities inside the network, while the environmental vulnerability scans are based on the specific environment of IoT devices operations. The offensive systems with the following IP addresses: 192.168.0.170, 142.250.200.205, 172.217.19.35, and 142.250.201.10, were used for performing comprehensive tests against web servers using a web server scanner tool, named *Nikto*.

3) MAN IN THE MIDDLE ATTACKS

This attack is intended to compromise and alter the flow of communication between two sides who assume to be in direct communication with each other. we focus on using this attack by targeting a couple of the most commonly used protocols in almost every system today, DNS and ARP.

- *DNS Spoofing attack*: The attacker uses the weaknesses of the DNS (Domain Name System) protocol and/or its implementation through the domain name servers. There are two main DNS Spoofing attacks: DNS ID Spoofing and DNS Cache Poisoning. Specifically, the attacker's objective is to associate the IP address of a machine under his control with a real and valid name of a public machine. When an IoT device wants to communicate with the edge server, the IoT device needs the IP address of the edge server. However, the IoT device may only have the name of the edge server. In this case, the IoT device will use the DNS protocol to obtain the IP address of the edge server from its name. The DNS ID Spoofing attack consists of capturing the ID number (i.e., when a DNS request is sent to a DNS server) in order to send a forged response before the DNS server and this by sniffing when the attack is performed on the same physical network. Since the DNS servers have a cache that keeps the correspondence between an IoT device name and its IP address for a certain time, the DNS Cache Poisoning consists of corrupting this cache with false information. The offensive systems with the following IP addresses: 192.168.0.101, 192.168.0.152, 172.217.19.35, and 192.168.0.170, were used to sniff and spoof DNS system using the *Ettercap* tool.
- *ARP Spoofing attack*: This is a MitM attack that allows attackers to intercept communications between network devices. The attacker requires prior access to the IoT network. Once on the targeted network, he scans the network to determine the IP addresses of at least two IoT devices. The attacker chooses his target and then sends false ARP responses where he sends ARP packets through the IoT network that contain the target's IP address and the attacker's MAC address. The false responses state that the correct MAC address for the two IP addresses, which are owned by the router and the target IoT device, is the attacker's MAC address. As the other IoT devices store the spoofed ARP packets, the data sent by these devices to the victim will be forwarded to the attacker instead. Based on this attack, the attacker can steal data or launch a more sophisticated tracking

attack. The offensive systems with the following IP address: 192.168.0.101, 192.168.0.152, 172.217.19.35, and 192.168.0.170, were used to sniff and spoof ARP systems using the *Ettercap* tool.

4) INJECTION ATTACKS

These attacks aim at compromising the integrity and confidentiality of the targeted system. We used three different approaches, namely XSS, SQL injection, and uploading attacks.

- *Cross-site Scripting (XSS) attack*: This is a type of security vulnerability of websites. Specifically, malicious scripts are injected into websites in order to attack users' systems. These scripts are created in scripting languages (e.g. JavaScript), which are run in the Internet browser. The potential threat of cross-site scripting is the possibility of uploading user data to the browser without any verification. The diversity of XSS-based attacks is practically unrestrained, but they generally involve forwarding vulnerable data, such as session information or cookies, to the attacker, forwarding the target to the attacker's controlled web content. The XSS attacks can usually be divided into two different types: reflected and stored. The offensive systems with the following IP addresses: 192.168.0.170, 172.217.19.42, 104.16.87.20, were used to detect, exploit and report XSS vulnerabilities using the *xsser* tool in a PHP/MySQL web applications (DVMA application).
- *SQL Injection*: This type of attack operates on the security vulnerabilities of an application that interacts with databases. The SQL attack involves the modification of a running SQL query by the injection of an unexpected query fragment, usually through a web form. The attacker can then access the database, but also change the information contained within it, thus damaging the safety of the application. The types of SQL injection can be classified into four types, including, error-based, stacked queries, blind-based, and union-based. The error-based method inserts fragments that return what the hacker is trying to extract from the database, field by field. With the stacked queries method, the attacker not only retrieves data but can also get data directly from the database, by injecting another SQL query. The blind-based injection type inserts fragments that return what the attacker is trying to extract from the database character by character. The union-based SQL injection inserts fragments that return a set of data directly retrieved from the database. The offensive systems with the following IP address: 192.168.0.170 were used to detect and exploit SQL injection flaws using the *sqlmap* tool.
- *Uploading attack*: There are many websites where users can upload documents (e.g., financial documents, resume, profile picture,etc.). Once the attacker successfully uploads a malware program file into the webserver, he/she can obtain administrative privileges. Based on this attack, the attacker can upload and run a web shell, filtrate potentially confidential data, upload a permanent XSS as well as a phishing page. The offensive systems with the following IP address: 192.168.0.170 were used to create a PHP backdoor using *Metasploit* framework and uploading through a PHP/MySQL web application (e.g., Damn Vulnerable Web App (DVWA)).

5) MALWARE ATTACKS

These kinds of attacks are the ones that have gone publicly viral in the last few years, not just because of the extensive damage they have caused, but also because of the reported losses involved. We used three types of such attacks, namely backdoor, password crackers, and ransomware attacks.

- *Backdoor attack*: This malicious software is used to provide attackers with unauthorized remote access to an infected IoT device by exploiting vulnerabilities in the system. An attacker can use the backdoor attack to sniff a user, manage his or her files, attack other hosts, install additional software or malware, as well as monitor the whole system. The offensive systems with the following IP address: 192.168.0.170 were used to create a python script backdoor using the *Metasploit* framework and then transferring it using the *curl* tool.
- *Password cracking attack*: This attack consists of trying to find a password or a key through successive attempts. This means that the password is broken by trying successive combinations until the right one is found. This can range from alphanumeric attempts: a, aa aaa, ab, abb, abbb, etc., or from a dictionary of the most commonly used passwords. The offensive systems with the following IP address: 192.168.0.170 were used to lunch this attack. The *CeWL* tool is used as a Ruby app for creating a list of words (password crackers) and email addresses (usernames).
- *Ransomware attack*: This is a type of malware that takes hostage files or IoT devices. Specifically, the attacker demands a ransom in exchange for restoring access or decrypting files. The cybercriminals behind this attack will contact the victim with their demands, promising to unlock the IoT device or decrypt the files once pay a ransom, which is usually in Bitcoin. The offensive systems with the following IP address: 192.168.0.170 were used to lunch this attack. After applying the Backdoor attack, the OpenSSL cryptography toolkit is used for creating RSA public/private keys and encrypting and decrypting victim files.

C. THE DIRECTORIES OF THE EDGE-IIoTset DATASET

As published in [1], the directories of the Edge-IIoTset datasets contain 49 files, which are organized into three sub-directories as follows:

1) NORMAL TRAFFIC OF IoT AND IIoT APPLICATIONS

This subdirectory is named Normal traffic, which contains the following 10 files.

TABLE 7. The list of extrapolated features obtained from different sources, including alerts, system resources, logs, IoT and IIoT network traffic.

N°	Name	Prot. Layer	Type	Description
1	frame.time	Frame	Date and time	Arrival Time
2	ip.src_host	IP	Character string	Source Host
3	ip.dst_host	IP	Character string	Destination Host
4	arp.dst.proto_ipv4	ARP	IPv4 address	Target IP address
5	arp.opcode	ARP	Unsigned integer	Opcode
6	arp.hw.size	ARP	Unsigned integer	Hardware size
7	arp.src.proto_ipv4	ARP	IPv4 address	Sender IP address
8	icmp.checksum	ICMP	Unsigned integer	Checksum
9	icmp.seq_le	ICMP	Unsigned integer	Sequence Number
10	icmp.transmit_timestamp	ICMP	Unsigned integer	Transmit Timestamp
11	icmp.unused	ICMP	Sequence of bytes	Unused
12	http.file_data	HTTP	Character string	File Data
13	http.content_length	HTTP	Unsigned integer	Content length
14	http.request.uri.query	HTTP	Character string	Request URI Query
15	http.request.method	HTTP	Character string	Request Method
16	http.referer	HTTP	Character string	Referer
17	http.request.full_uri	HTTP	Character string	Full request URI
18	http.request.version	HTTP	Character string	Request Version
19	http.response	HTTP	Boolean	Response
20	http.tls_port	HTTP	Label	Unencrypted HTTP protocol detected over encrypted port
21	tcp.ack	TCP	Unsigned integer	Acknowledgment Number
22	tcp.ack_raw	TCP	Unsigned integer	Acknowledgment number (raw)
23	tcp.checksum	TCP	Label	Checksum
24	tcp.connection.fin	TCP	Label	Connection finish (FIN)
25	tcp.connection.rst	TCP	Label	Connection reset (RST)
26	tcp.connection.syn	TCP	Label	Connection establish request (SYN)
27	tcp.connection.synack	TCP	Label	Connection establish acknowledge (SYN+ACK)
28	tcp.dstport	TCP	Label	Destination Port
29	tcp.flags	TCP	Label	Flags
30	tcp.flags.ack	TCP	Boolean	Acknowledgment
31	tcp.len	TCP	Unsigned integer	TCP Segment Len
32	tcp.options	TCP	Sequence of bytes	TCP Options
33	tcp.payload	TCP	Sequence of bytes	TCP payload
34	tcp.seq	TCP	Unsigned integer	Sequence Number
35	tcp.srcport	TCP	Unsigned integer	Source Port
36	udp.port	UDP	Unsigned integer	Source or Destination Port
37	udp.stream	UDP	Unsigned integer	Stream index
38	udp.time_delta	UDP	Time offset	Time since previous frame
39	dns.qry.name	DNS	Character string	Name
40	dns.qry.name.len	DNS	Unsigned integer	Name Length
41	dns.qry.qu	DNS	Boolean	"QU"" question"
42	dns.qry.type	DNS	Unsigned integer	Type
43	dns.retransmission	DNS	Boolean	Retransmission
44	dns.retransmit_request	DNS	DNS query retransmission	Label
45	dns.retransmit_request_in	DNS	Frame number	Retransmitted request. Original request in
46	mqtt.conack.flags	MQTT	Unsigned integer	Acknowledge Flags
47	mqtt.conflag.cleansess	MQTT	Boolean	Clean Session Flag
48	mqtt.conflags	MQTT	Unsigned integer	Connect Flags
49	mqtt.hdrflags	MQTT	Unsigned integer	Header Flags
50	mqtt.len	MQTT	Unsigned integer	Msg Len
51	mqtt.msg_decoded_as	MQTT	Character string	Message decoded as
52	mqtt.msg	MQTT	Sequence of bytes	Message
53	mqtt.msgtype	MQTT	Unsigned integer	Message Type
54	mqtt.proto_len	MQTT	Unsigned integer	Protocol Name Length
55	mqtt.protoname	MQTT	Character string	Protocol Name
56	mqtt.topic	MQTT	Character string	Topic
57	mqtt.topic_len	MQTT	Unsigned integer	Topic Length
58	mqtt.ver	MQTT	Unsigned integer	Version
59	mbtcp.len	Modbus/TCP	Unsigned integer	Length
60	mbtcp.trans_id	Modbus/TCP	Unsigned integer	Transaction Identifier
61	mbtcp.unit_id	Modbus/TCP	Unsigned integer	Unit Identifier
62	Attack_label	/	Number	0 indicates normal and 1 indicates attacks
63	Attack_type	/	Character string	Attack categories

- File 1.1 (Distance): This file includes two documents, namely, Distance.csv and Distance.pcap. The IoT sensor (Ultrasonic sensor) is used to capture the IoT data.

- File 1.2 (Flame_Sensor): This file includes two documents, namely, Flame_Sensor.csv and Flame_Sensor.pcap. The IoT sensor (Flame Sensor) is used to capture the IoT data.
- File 1.3 (Heart_Rate): This file includes two documents, namely, Flame_Sensor.csv and Flame_Sensor.pcap. The IoT sensor (Flame Sensor) is used to capture the IoT data.
- File 1.4 (IR_Receiver): This file includes two documents, namely, IR_Receiver.csv and IR_Receiver.pcap. The IoT sensor (IR (Infrared) Receiver Sensor) is used to capture the IoT data.
- File 1.5 (Modbus): This file includes two documents, namely, Modbus.csv and Modbus.pcap. The IoT sensor (Modbus Sensor) is used to capture the IoT data.
- File 1.6 (phValue): This file includes two documents, namely, phValue.csv and phValue.pcap. The IoT sensor (pH-sensor PH-4502C) is used to capture the IoT data.
- File 1.7 (Soil_Moisture): This file includes two documents, namely, Soil_Moisture.csv and Soil_Moisture.pcap. The IoT sensor (Soil Moisture Sensor v1.2) is used to capture the IoT data.
- File 1.8 (Sound_Sensor): This file includes two documents, namely, Sound_Sensor.csv and Sound_Sensor.pcap. The IoT sensor (LM393 Sound Detection Sensor) is used to capture the IoT data.
- File 1.9 (Temperature_and_Humidity): This file includes two documents, namely, Temperature_and_Humidity.csv and Temperature_and_Humidity.pcap. The IoT sensor (DHT11 Sensor) is used to capture the IoT data.
- File 1.10 (Water_Level): This file includes two documents, namely, Water_Level.csv and Water_Level.pcap. The IoT sensor (Water sensor) is used to capture the IoT data.

2) ATTACK TRAFFIC OF IoT AND IIoT APPLICATIONS

This subdirectory is named Attack traffic, which contains the following 28 files, including 14 CSV files and 14 PCAP files.

- Attack traffic (CSV files): This 14 files includes Backdoor_attack.csv, DDoS_HTTP_Flood_attack.csv, DDoS_ICMP_Flood_attack.csv, DDoS_TCP_SYN_Flood_attack.csv, DDoS_UDP_Flood_attack.csv, MITM_attack.csv, OS_Fingerprinting_attack.csv, Password_attack.csv, Port_Scanning_attack.csv, Ransomware_attack.csv, SQL_injection_attack.csv, Uploading_attack.csv, Vulnerability_scanner_attack.csv, XSS_attack.csv. Each file is specific for each attack.
- Attack traffic (PCAP files): This 14 files includes Backdoor_attack.pcap, DDoS_HTTP_Flood_attack.pcap, DDoS_ICMP_Flood_attack.pcap, DDoS_TCP_SYN_Flood_attack.pcap, DDoS_UDP_Flood_attack.pcap, MITM_attack.pcap, OS_Fingerprinting_attack.pcap, Password_attack.pcap, Port_Scanning_attack.pcap, Ransomware_attack.pcap, SQL_injection_attack.pcap, Uploading_attack.pcap, Vulnerability_scanner_attack.pcap, XSS_attack.pcap. Each file is specific for each attack.

3) SELECTED DATASET FOR ML AND DL

This directory contains two CSV files namely, DNN-EdgeIIoT-dataset.csv and ML-EdgeIIoT-dataset.csv. The DNN-EdgeIIoT-dataset.csv contains a selected dataset for the use of evaluating deep learning-based intrusion detection systems. The ML-EdgeIIoT-dataset.csv contains a selected dataset for the use of evaluating traditional machine learning-based intrusion detection systems.

V. EXTRAPOLATED FEATURES

To extract flow features from the network packets (i.e., PCAP files), we have analyzed different sources, including alerts, system resources, logs, and network traffic. Therefore, we have analyzed the attributes of each protocol recorded in the network, namely, frame, Internet Protocol Version 4 (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Domain Name System (DNS), MQ Telemetry Transport Protocol (MQTT), Modbus/TCP (mbtcp). Then, we have used two networks protocols analyzers, namely, the *Zeek* tool and *TShark* tool, to extract and filter the features. To identify features with high correlation, we have developed a python script based on the *Yellowbrick* package. The 61 extrapolated features are demonstrated in Tab. 7.

We indicate that an authentic labeling operation was performed to label all records, whether normal or attack. Specifically, we have added two new attributes, namely, Attack_label and Attack_type. The Attack_label contains 0 or 1, which is used for the binary classification model (i.e., 0 indicates normal and 1 indicates attacks). The Attack_type presents the attack categories, that are used for the multiclass classification model (i.e., a classification task with more than two classes).

A. INTERNET PROTOCOL VERSION 4 (IP)

The Internet Protocol delivers the transport feature of the network layer (layer 3), which is deployed to transmit data packets from one IP address to other addresses. The end-user of the network layer will provide a remote IP address with a packet, which IP is required to forward the packet to that particular host. Based on the network protocol analyzer tool, we have found 138 attributes (e.g., Source or Destination Address, Destination Address, Destination Host, Timestamp, Transmission Control Code, IPv4 Fragment, Version, etc.). We have identified and selected two features with high correlation, including, ip.src_host and ip.dst_host.

B. ADDRESS RESOLUTION PROTOCOL (ARP)

The address resolution protocol is employed to determine the address assignment between a Layer 3 (protocol) address and a Layer 2 (hardware) address in a dynamic manner.

Based on the network protocol analyzer tool, we have found 51 attributes (e.g., Target hardware address, Target protocol address, Target IP address, Hardware size, Hardware type, Sender MAC address, Sender protocol size, Sender protocol address, Sender IP address, etc.). We have identified and selected four features with high correlation, including, arp.dst.proto_ipv4, arp.opcode, arp.hw.size, and arp.src.proto_ipv4.

C. INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

This protocol is used by IP to transfer control messages between IP hosts. Based on the network protocol analyzer tool, we have found 107 attributes (e.g., Address entry size, Address Mask, Checksum, Timestamp from ICMP data, ICMP Extensions, Sequence Number, Address Family Identifier, Interface Index, Name Length, Length of the original datagram, UDP tunneling, Gateway Address, Request frame, Response time, etc.). We have identified and selected four features with high correlation, including, icmp.checksum, icmp.seq_le, icmp.transmit_timestamp, and icmp.unused.

D. HYPERTEXT TRANSFER PROTOCOL (HTTP)

This is a text-based request-response client-server protocol, where an HTTP request is sent to an HTTP server (e.g. the Apache HTTP server) by an HTTP client (e.g. a web browser such as Mozilla). Based on the network protocol analyzer tool, we have found 83 attributes (e.g., Full request URI, Request URI Query, Status Code, Sec-WebSocket-Accept, Time since request, Set-Cookie, Sec-WebSocket-Extensions, Proxy-Authenticate, Proxy-Authorization, Proxy-Connect-Hostname, Proxy-Connect-Port, File Data, etc.). Based on the network protocol analyzer tool, we have found 83 attributes (e.g., Full request URI, Request URI Query, Status Code, Sec-WebSocket-Accept, Time since request, Set-Cookie, Sec-WebSocket-Extensions, Proxy-Authenticate, Proxy-Authorization, Proxy-Connect-Hostname, Proxy-Connect-Port, File Data, etc.). We have identified nine features with high correlation, including, http.file_data, http.content_length, http.request.uri.query, http.request.method, http.referer, http.request.full_uri, http.request.version, http.response, and http.tls_port.

E. TRANSMISSION CONTROL PROTOCOL (TCP)

The TCP protocol offers a connection-oriented data transfer based on the flow of data. Based on the network protocol analyzer tool, we have found 267 attributes (e.g., Acknowledgment Number, SEQ/ACK analysis, TCP Analysis Flags, TCP window update, Checksum, Proxy-Authenticate, Conversation completeness, Connection finish (FIN), TCP segment data, TCP Flags, MD5 digest, Multipath TCP Data ACK, etc.). We have identified fifteen features with high correlation, including, tcp.ack, tcp.ack_raw, tcp.checksum, tcp.connection.fin, tcp.connection.rst, tcp.connection.syn, tcp.connection.synack, tcp.dstport, tcp.flags, tcp.flags.ack, tcp.len, tcp.options, tcp.payload, tcp.seq, and tcp.srcport.

Algorithm 1: Dataset Processing and Analyzing

```
Processing ($CSV_1, \dots, CSV_n$):
    $Joined\_files = CSV_1, \dots, CSV_n$
    $Selected\_Features = F_1, \dots, F_n$
    for $i$ = 1,..,$n$ do
        $add\_column(CSV_i,$ “$Attack\_label''$)
        $add\_column(CSV_i,$ “$Attack\_type''$)
    end
    if IoT traffic is normal then
        for $i$ = 1,..,$n$ do
            $add\_values(CSV_i,$ “$Attack\_label''$, “0″)
            $add\_values(CSV_i,$ “$Attack\_type''$, “$Normal''$)
        end
    end
    if IoT traffic is attack then
        for $i$ = 1,..,$n$ do
            $add\_values(CSV_i,$ “$Attack\_label''$, “1″)
            $add\_values(CSV_i,$ “$Attack\_type''$, “$Type''$)
        end
    end
    $CSV_x = concat(joined\_files)$
    $CSV_x = drop(CSV_x, Selected\_Features)$
    $CSV_x = drop\_duplicates(CSV_x)$
    $CSV_x =$
      $encode\_text\_dummy(CSV_x, Selected\_Features)$
    $CSV\_train, CSV\_test = split\_dataset(CSV_x)$
    $OneHotEncoder(CSV\_train, CSV\_test)$
    $StandardScaler(CSV\_train, CSV\_test)$
    $SMOT(CSV\_train, CSV\_test)$
```

Algorithm 2: Centralized Learning

```
Edge Server ($x$):
    $\mathcal{B} \leftarrow$ Split($\mathcal{P}, B$)
    for $i$ = 1,..,$E$ do
        for $b \in \mathcal{B}$ do
            $x \leftarrow x - \eta \nabla f_e(x, b)$
        end
    end
```

F. USER DATAGRAM PROTOCOL (UDP)

The UDP layer offers transport layer (layer 4) functionality based on connectionless datagrams. Based on the network protocol analyzer tool, we have found 30 attributes (e.g., Checksum, Bad checksum, Destination Port, Length, Payload, Source or Destination Port, Destination process ID, Source process ID, Source Port, Stream index, Location, PDU Size, etc.). We have identified three features with high correlation, including, udp.port, udp.stream, and udp.time_delta.

G. DOMAIN NAME SYSTEM (DNS)

DNS is the system used to solve the storage of domain name information, including mail servers, IP addresses, and other

TABLE 8. Notation for the discussion of algorithms.

Notation	Description
η	Local learning rate
C	Fraction of nodes used at each iteration for each node
K	Total number of clients
R	Total number of the federated rounds
x	Initial weights
S_t	Clients per round
$\mathcal{B}$	The local minibatch size
k	The K clients are indexed by k
E	Number of local updates
$f_C(.,.)$	Client function
$f_e(.,.)$	Edge function
$add\_column(.,.)$	Function for adding a column
$add\_values(.,.,.)$	Function for adding a value
CSV	CSV file
$concat(.)$	Function for merging the CSV files
$drop(.,.)$	Function for drop the columns
$drop\_duplicates(.)$	Function for drop the duplicated rows
$encode\_text\_dummy(.,.)$	Function for encoding text values to dummy variables
$split\_dataset(.)$	Function for split dataset into random train and test subsets
$OneHotEncoder(.,.)$	Function for encoding categorical features as a one-hot numeric array
$StandardScaler(.,.)$	Function for standardizing features
$SMOT$	Function for performing over-sampling using SMOTE
I_F	The importance of the features
$Found\_Features\_Importance(Model)$	Function for finding the importance of the features
L_F	The lowest importance of the features
$Found\_Features\_lowest(Model)$	Function for finding the lowest importance of the features
$Features(.)$	Function for finding the number of features
T_F	The number of the target features

Algorithm 3: Federated Learning (FedAvg) [31]

```
Edge Server (K, C, R):
    $x_1 \leftarrow GenericModel()$
    for $t = 1, .., R$ do
        $S_t \leftarrow$ Subset(max($C \cdot K$, 1), “random″)
        Parallel.for $k \in S_t$ do
            $x_{t+1}^k \leftarrow IoTDevice(x_t, k)$
        end
        $x_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} x_{t+1}^k$
    end
IoT device (x, k):
    $\mathcal{B} \leftarrow$ Split($\mathcal{P}$, $B$)
    for $i$ = 1,..,E do
        for $b \in \mathcal{B}$ do
            $x \leftarrow x - \eta \nabla f_c(x, b)$
        end
    end
    Send $x$ to Edge Server
```

Algorithm 4: Feature Selection

```
Feature_selection ($CSV_x$, $T_F$):
    $Model \leftarrow GenericModel(RandomForest)$
    while $Features(CSV_x) > T_F$ do
        $CSV\_train, CSV\_test = split\_dataset(CSV_x)$
        Train Model with CSV_train
        $I_F \leftarrow Found\_Features\_Importance(Model)$
        $L_F \leftarrow Found\_Features\_lowest(Model)$
        $drop(CSV_x, L_F)$
    end
```

details. Based on the network protocol analyzer tool, we have found 357 attributes (e.g., Address, Hostname, Prefix Length, Issuer Critical, Report URL, Certificate, Key Tag, Digest, Conflict, DNS Gateway, IPv4 Gateway, Source Netmask, etc.). We have identified seven features with high correlation, including, dns.qry.name, dns.qry.name.len, dns.qry.qu, dns.qry.type, dns.retransmission, dns.retransmit_request, and dns.retransmit_request_in.

H. MQ TELEMETRY TRANSPORT PROTOCOL (MQTT)

This is the most important protocol for IoT applications, which is used in a wide variety of industries. The MQTT protocol is constructed as an ultra-lightweight publish/subscribe messaging transport, which is suitable for interconnecting remote devices using a small code footprint and low network bandwidth. Based on the network protocol analyzer tool, we have found 78 attributes (e.g., Reason Code, Client ID, Client ID Length, Acknowledge Flags, QoS Level, User Name Flag, Connect Flags, Keep Alive, Msg Len, Message, Password, QoS, etc.). We have identified thirteen features with high correlation, including, mqtt.conack.flags, mqtt.conflag.cleansess, mqtt.conflags, mqtt.hdrflags, mqtt.len, mqtt.msg_decoded_as, mqtt.msg, mqtt.msgtype, mqtt.proto_len, mqtt.protoname, mqtt.topic, mqtt.topic_len, and mqtt.ver.

TABLE 9. Settings for deep learning classifier.

	Parameter	Value
Centralized	Batch size	800
	Total epochs	25
Federated	Local epochs	3
	Global epochs	10
	batch_size	100
*	Hidden layers	2
	Hidden nodes	90
	Regularization	L2
	Activation function	ReLu
	Classification function	softmax
	Optimizer	adam

I. MODBUS/TCP (MBTCP)

The Modbus/TCP protocol is commonly adopted in IIoT as a local interface to manage IIoT devices, which is the Modbus RTU protocol with a TCP interface. This protocol uses a client/server architecture (i.e., runs on Ethernet). Based on the network protocol analyzer tool, we have found 65 attributes (e.g., Length, Data, diagnostic code, Broadcast Received, Character Overrun, Communication Error, Slave Abort Exception Sent, status, Number of Objects, Read Device ID, Protocol Identifier, Transaction Identifier, function code, etc.). We have identified three features with high correlation, including, mbtcp.len, mbtcp.trans_id, and mbtcp.unit_id.

VI. THE PERFORMANCE EVALUATION

This section discusses the experimental results of the proposed Edge IIoT dataset, using centralized and federated deep learning-based intrusion detection with common evaluation metrics. Fig. 3 illustrates the main difference between these two learning approaches. Tab. 8 presents the notations list used within the proposed algorithms.

Firstly, we have used four conventional machine learning algorithms namely, Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), as well as the most popular Deep Neural Network (DNN) for cyber-attack detection to evaluate models accuracy, using both centralized and federated learning against the proposed large-scale and heterogeneous dataset. Tab. 9 shows the different parameters applied for the implemented deep learning classifiers. Therefore, as familiar with machine learning workflow, we start with preparing data and cleaning our data from duplicates and missing values such as NAN (Not A Number) or 'INF' (Infinite Value). Then, we drop unnecessary flow features such as IP addresses, ports, timestamp, and payload information (i.e frame.time, ip.src_host, ip.dst_host, arp.src.proto_ipv4, arp.dst.proto_ipv4, http.file_data, http.request.full_uri, icmp.transmit_timestamp, http.request.uri.query, tcp.options, tcp.payload, tcp.srcport, tcp.dstport, udp.port, mqtt.msg). After that, we perform label-encoding by mapping the remaining categorical features (non-numeric) to numeric values. We apply feature scaling using a standardization algorithm. We split the data to produce Train sets for training and validating and Test sets for the final model evaluation. The statistics of normal and attacks involved in the dataset are described in Tab. 10.

TABLE 10. Statistics of normal and attacks in Edge-IIoTset.

IoT traffic	Class	Records	Total
Normal	Normal	11223940	11223940
Attack	Backdoor attack	24862	9728708
	DDoS_HTTP attack	229022	
	DDoS_ICMP attack	2914354	
	DDoS_TCP attack	2020120	
	DDoS_UDP attack	3201626	
	Fingerprinting attack	1001	
	MITM attack	1229	
	Password attack	1053385	
	Port_Scanning attack	22564	
	Ransomware attack	10925	
	SQL_injection attack	51203	
	Uploading attack	37634	
	Vulnerability_scanner attack	145869	
	XSS attack	15915	
Total			20952648

TABLE 11. Statistics of total selected observation for training and testing.

Class	Total	Train	Test
Normal	24301	19281	4820
Backdoor attack	10195	7892	1973
DDoS_HTTP attack	10561	8396	2099
DDoS_ICMP attack	14090	10477	2619
DDoS_TCP attack	10247	8198	2049
DDoS_UDP attack	14498	11598	2900
Fingerprinting attack	1001	682	171
MITM attack	1214	286	72
Password attack	9989	7978	1994
Port_Scanning attack	10071	7137	1784
Ransomware attack	10925	7751	1938
SQL_injection attack	10311	8225	2057
Uploading attack	10269	8171	2043
Vulnerability_scanner attack	10076	8050	2012
XSS attack	10052	7634	1909

Tab. 11 illustrates the randomly selected subsets data for ML algorithms and the resulting Train and Test sets after data cleaning and splitting. For the DNN we have selected a greater portion of data for more accuracy. We also used SMOTE for oversampling minority classes (MITM, Fingerprinting) to enhance the overall model efficiency.

We conducted three experiments using Binary, 6-class, and 15-class classification to better study both traffic predictability and detection efficiency of various cyber-attacks and threat models. Furthermore, we studied centralized and federated learning to evaluate detection accuracy when considering privacy, heterogeneity, and the availability of data issues.

The availability of cloud solutions to overcome the shortcomings of resource limitation, centralized learning characterized by the availability of rich data promotes higher detection capabilities against complicated and large-scale attack patterns. Thus, we have studied various centralized detection approaches using google Colab resources.

For machine learning algorithms, we implement a workflow pipeline composed of: features selection using Random

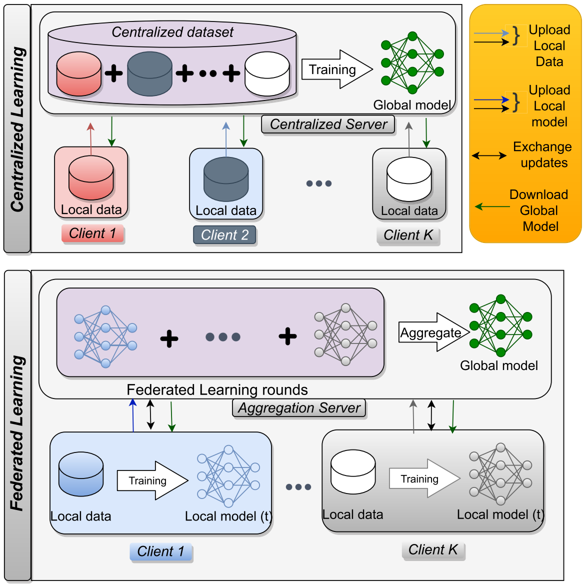
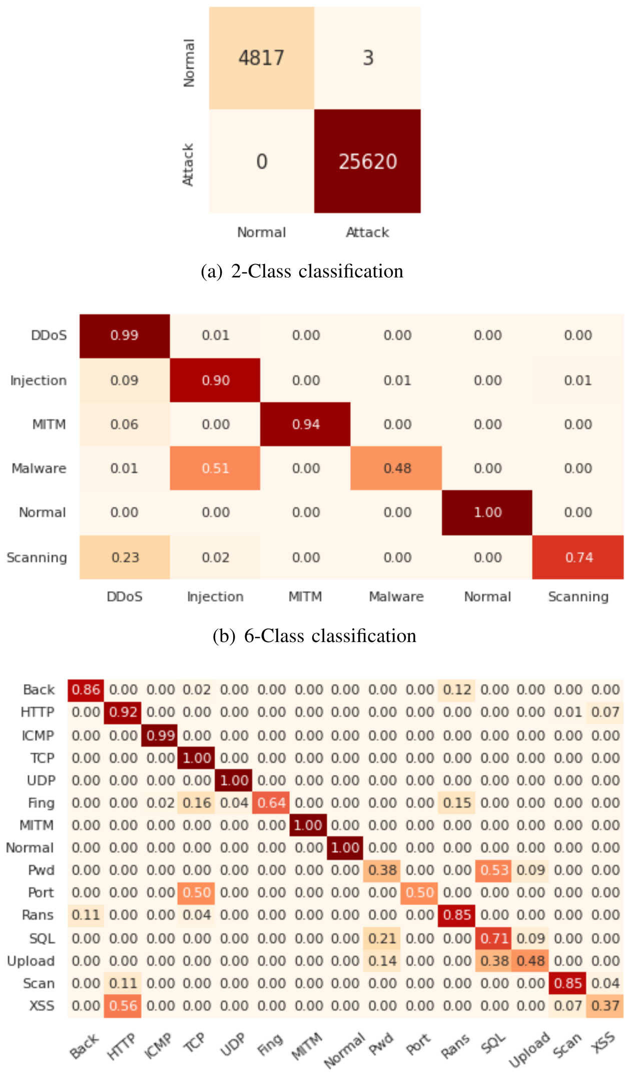


FIGURE 3. Centralized vs. Federated learning modes.



(a) 2-Class classification

(b) 6-Class classification

(c) 15-Class classification

FIGURE 4. Confusion matrix of DNN using binary and multi-class classification (2-class, 6 class, and 15 class).

Forest, model initialization, and hyper-parameter tuning using Grid Search with stratified cross-validation technique to finally obtain a generalized and more efficient model. The algorithms 1, 2, 3, and 4 are used in the performance evaluation of the Edge-IIoTset dataset, for 1) dataset processing and analyzing, 2) centralized learning approach, 3) federated learning (FedAvg) approach [31], 4) feature selection method, respectively. The resulted models were then evaluated using the Test data and considering the following detection metrics:

- *Accuracy*: is used to determine the proportion of correct classifications to the total number of entries, which is given by :

$$Acc = \frac{TP_{Attack} + TN_{Normal}}{TP_{Attack} + TN_{Normal} + FP_{Normal} + FN_{Attack}} \tag{1}$$

- *Precision*: denotes the proportion of correct attack classes to the total amount of predicted attack results, which can be given by :

$$Pr = \frac{TP_{Attack}}{TP_{Attack} + FP_{Normal}} \tag{2}$$

- *Recall*: denotes the proportion of proper attack classifications relative to the overall count of all samples that ought to have been identified as attacks, it is given by :

$$Rc = \frac{TP_{Attack}}{TP_{Attack} + FN_{Attack}} \tag{3}$$

- F_1-Score: reports the Harmonic Mean between Precision and Recall, which is given by:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

A. CENTRALIZED MACHINE LEARNING

The confusion matrix of the DNN model of the Edge-IIoTset datasets (i.e., 2-class, 6-class, and 15-class) are depicted in Fig. 4. Fig. 5 presents the centralized model performance of the accuracy of machine learning techniques (DT, RF, SVM, KNN, DNN) in multiclass classification (i.e., 15-Class and 6-Class) and binary classification (i.e., 2-Class). For multiclass classification (15-Class), the highest accuracy was obtained using the DNN classifier which achieved 94.67%, while the lowest accuracy was obtained using the DT classifier with 67.11%, RF classifier with 80.83%, SVM classifier with 77.61%, and KNN classifier with 79.18%. For multiclass classification (6-Class), the highest accuracy was obtained using the DNN classifier which achieved 96.01%, while the lowest accuracy was obtained using the DT classifier with 77.90%, RF classifier with 82.90%, SVM classifier with 85.62%, and KNN classifier with 83.39%. For binary

TABLE 12. Classification report for 2-class of deep learning (Centralized model performance).

Epoch	Learning rate	Error %	Accuracy %	Validation_Error %	Validation_Accuracy %	Training time	Validation time
1	0.5	07.36	97.07	0.64	99.98	450ms	18ms
	0.1	12.61	96.76	0.95	99.99	469ms	19ms
	0.01	5.93	80.48	32.77	97.95	451ms	18ms
2	0.5	0.28	99.99	0.26	99.9	67ms	3ms
	0.1	0.49	100	0.47	100	65ms	3ms
	0.01	24.34	99.49	17.83	99.95	63ms	3ms
3	0.5	0.09	100	0.32	100	66ms	3ms
	0.1	0.29	100	0.32	100	62ms	2ms
	0.01	14.2	99.98	11.41	99.99	70ms	3ms

TABLE 13. Classification report for 6-class of traditional machine learning as well as deep learning (Centralized model performance).

Alg	Metric	Normal	DDoS attacks	Injection attacks	MITM attacks	Malware attacks	Scanning attacks
DT	Pr	1.00	0.73	0.61	1.00	0.85	0.82
	Rc	1.00	0.92	0.57	0.99	0.69	0.62
	F1	1.00	0.81	0.59	0.99	0.76	0.71
RF	Pr	1.00	0.98	0.67	1.00	0.73	0.76
	Rc	1.00	0.83	0.82	1.00	0.72	0.80
	F1	1.00	0.90	0.74	1.00	0.73	0.78
SVM	Pr	1.00	0.96	0.67	1.00	0.97	0.74
	Rc	1.00	0.83	0.91	1.00	0.68	0.92
	F1	1.00	0.89	0.77	1.00	0.80	0.82
Knn	Pr	1.00	0.88	0.63	0.99	0.93	0.82
	Rc	1.00	0.90	0.86	1.00	0.73	0.58
	F1	1.00	0.89	0.73	0.99	0.82	0.68
DNN	Pr	1.00	0.92	0.66	1.00	0.97	0.96
	Rc	1.00	0.99	0.90	0.94	0.48	0.74
	F1	1.00	0.95	0.76	0.97	0.64	0.84

Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Deep Neural Network (DNN), Precision (Pr), Recall (Re), F_1-Score (F1)

classification (2-Class), the highest accuracy was obtained using three classifiers namely, RF, SVM, KNN, DNN which achieved 99.99%, while the DT classifier obtained an accuracy of 99.98%. These obtained results prove that the deep learning approach is efficient for intrusion detection compared to traditional machine learning techniques (DT, RF, SVM, KNN) in centralized mode, especially with big data availability.

Tab. 12 provides a summary of DNN model learning for attack detection (Binary classification). We implemented a shallow DNN with only 47 trainable parameters. The model converges quickly achieving higher detection accuracy of 99.99%. As depicted in the classification report (Fig. 4(a)). The normal class pattern was well discriminated from all the attack patterns, due to the nature of IIoT physical objects that are typically task-oriented and maintain relatively stable data distribution which enhances both the effectiveness and the efficiency of attack detection with real-time capabilities. Tab. 13 provides the obtained centralized model results of machine learning techniques (DT, RF, SVM, KNN, DNN) in terms of F1-score, Recall, Precision, under multi-class classification (6 class). It can be seen that the DNN classifier gives the highest precision rate for Normal traffic and three types of attacks, namely, MITM attacks 100%, Malware attacks 97%, and Scanning attacks 94%, while for DDoS attacks and Injection attacks, the highest precision rate are given by RF classifier with 98% and 67%, respectively. We note that all machine learning algorithms produce no false positives for the Normal class, which means that the precision rate is 100%. Therefore, we observe that the DNN classifier can give a higher recall for two types of attacks, namely, DDoS attacks with 99% and MITM attacks with 100%. For the Injection attacks and Scanning attacks, the SVM classifier gives a

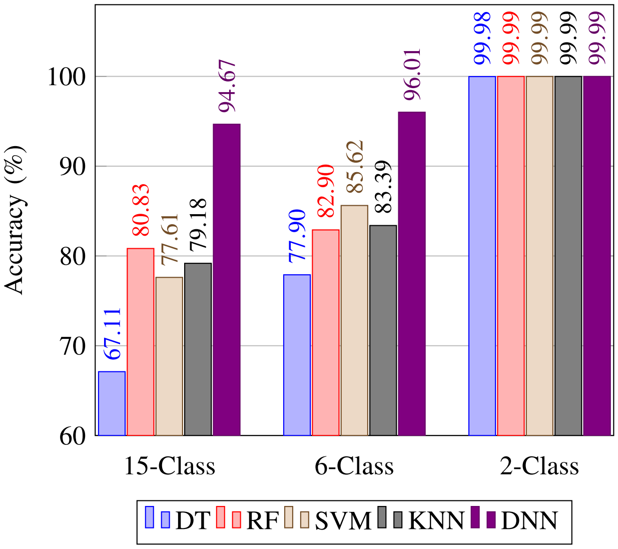


FIGURE 5. Centralized model performance.

TABLE 14. Classification report for 15-classes of traditional machine learning as well as deep learning (Centralized model performance).

Alg	Metr	Normal	Back	HTTP	ICMP	TCP	UDP	Fing	MITM	Pwd	Port	Rans	SQL	Upload	Scan	XSS
DT	Pr	1.00	0.86	0.44	1.00	0.66	1.00	0.00	1.00	0.00	0.71	0.80	0.34	1.00	1.00	0.15
	Rc	1.00	0.78	0.63	1.00	1.00	1.00	0.00	1.00	0.00	0.48	0.86	0.96	0.02	0.00	0.23
	f1	1.00	0.82	0.52	1.00	0.80	1.00	0.00	1.00	0.00	0.58	0.83	0.50	0.04	0.00	0.18
RF	Pr	1.00	0.99	0.64	0.96	1.00	1.00	0.77	1.00	0.41	0.63	0.96	0.76	0.66	1.00	0.65
	Rc	1.00	0.92	0.82	1.00	0.60	1.00	0.10	1.00	0.81	1.00	0.93	0.21	0.51	0.81	0.58
	f1	1.00	0.95	0.72	0.98	0.75	1.00	0.18	1.00	0.54	0.77	0.95	0.33	0.58	0.90	0.61
Knn	Pr	1.00	0.96	0.69	1.00	0.76	1.00	0.79	0.97	0.45	0.74	0.95	0.50	0.63	0.88	0.49
	Rc	1.00	0.94	0.74	0.99	0.80	1.00	0.70	1.00	0.56	0.73	0.94	0.49	0.49	0.48	0.69
	F1	1.00	0.95	0.72	1.00	0.78	1.00	0.74	0.99	0.50	0.73	0.95	0.50	0.55	0.62	0.57
SVM	Pr	1.00	0.63	0.86	1.00	1.00	1.00	0.80	1.00	0.60	0.64	0.69	0.42	0.66	0.95	0.61
	Rc	1.00	0.77	0.60	0.99	0.59	1.00	0.66	1.00	0.23	1.00	0.51	0.82	0.40	0.86	0.88
	f1	1.00	0.69	0.71	0.99	0.74	1.00	0.72	1.00	0.34	0.78	0.59	0.55	0.50	0.90	0.72
DNN	Pr	1.00	0.95	0.76	1.00	0.82	1.00	0.59	1.00	0.55	1.00	0.73	0.47	0.67	0.96	0.53
	Rc	1.00	0.86	0.92	0.99	1.00	1.00	0.64	1.00	0.38	0.50	0.85	0.71	0.48	0.85	0.37
	F1	1.00	0.90	0.83	1.00	0.90	1.00	0.61	1.00	0.45	0.66	0.79	0.57	0.56	0.90	0.43

Backdoor attack (Back), HTTP flood DDoS attack (HTTP), ICMP flood DDoS attack (ICMP), TCP SYN Flood DDoS attack, (TCP), UDP flood DDoS attack (UDP), OS Fingerprinting attack (Fing), Man in the middle attack (MITM) , Password cracking attack(Pwd) , Port Scanning attack (Port), Ransomware attack (Rans) , SQL Injection (SQL), Upload attack (Upload), Vulnerability scanning attack (Scan), Cross-site Scripting attack (XSS), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Deep Neural Network (DNN), Precision (Pr), Recall (Re), F_1-Score (F1), Metr: metrics.

TABLE 15. The evaluation results of the federated deep learning approach.

		1^{st} round						10^{th} round					
		IID			Non IID			IID			Non IID		
Dataset	**Clients**	**B**	**W**	**G**	**B**	**W**	**G**	**B**	**W**	**G**	**B**	**W**	**G**
2-Class	$K = 5$	98.55	98.33	98.45	99.98	99.97	99.97	100.00	99.99	100.00	100.00	99.99	99.99
	$K = 10$	99.98	99.97	99.98	99.98	99.15	99.99	100.00	99.99	99.99	99.99	99.98	99.98
	$K = 15$	99.97	99.92	99.95	99.92	76.73	96.70	100.00	99.99	99.99	99.98	99.97	99.98
6-Class	$K = 5$	95.32	95.2	95.18	93.06	03.71	86.46	95.94	95.82	95.98	93.13	03.71	86.86
	$K = 10$	95.33	95.26	95.34	95.21	03.72	87.35	96.00	95.89	95.99	95.22	07.45	91.10
	$K = 15$	95.31	94.51	95.11	93.31	03.72	86.53	95.63	95.08	95.38	95.05	03.75	92.95
15-Class	$K = 5$	93.30	92.90	93.04	91.44	01.06	77.04	93.62	93.36	93.59	91.77	39.10	91.33
	$K = 10$	93.32	93.03	93.14	91.13	08.23	73.78	93.86	93.26	93.89	92.52	11.28	91.45
	$K = 15$	93.00	93.02	93.22	88.19	10.75	71.42	93.38	92.91	93.37	90.83	13.61	91.74

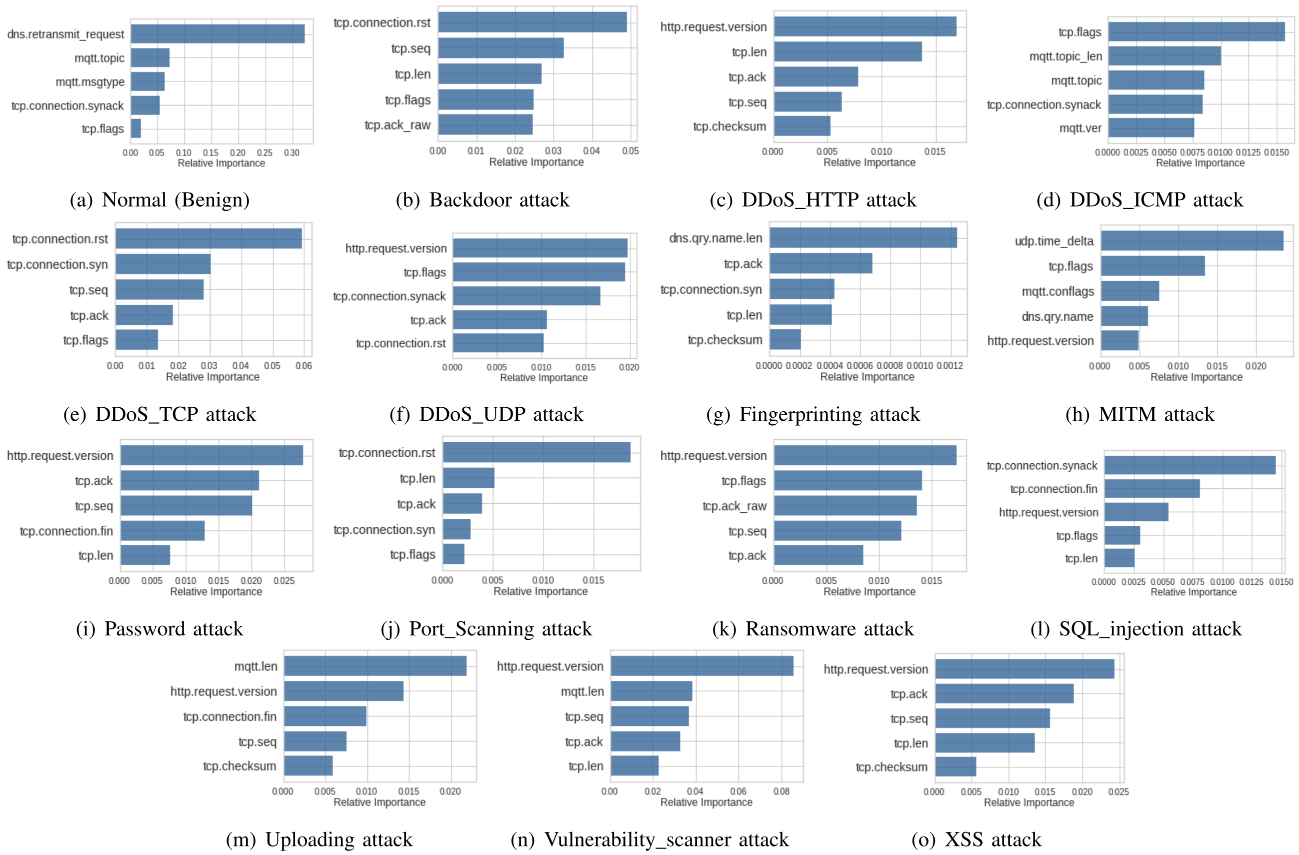
(**B**): **B**est client accuracy; (**W**): **W**orst client accuracy; (**G**): **G**lobal model accuracy;

higher recall with 91% and 91%, respectively. We note also that all machine learning algorithms produce no false positives for the Normal class, which means that the recall rate is 100%. Fig. 6 illustrates the five more important features for each class based on the interpretation of Random Forest prediction which is helpful for further forensic analysis. We can see that different protocols information contributed well to identifying a variety of attacks.

Tab. 14 provides the obtained centralized model results of machine learning techniques (DT, RF, SVM, KNN, DNN) in terms of F1-score, Recall, Precision, under multi-class classification (15 class). It can be seen that the DNN classifier gives the highest precision rate for Normal traffic and six types of attacks, namely, Backdoor attack 99%, ICMP flood DDoS attack 100%, UDP flood DDoS attack 100%, MITM attack 100%, Port Scanning attack 100%, and SQL Injection 91%. The SVM classifier gives the highest precision rate for Normal traffic and three types of attacks, namely, HTTP flood DDoS attack 86%, OS Fingerprinting attack 80%, and Password cracking attack 61%. The RF classifier gives the highest precision rate for Normal traffic and three types of attacks, namely, TCP SYN Flood DDoS attack 100%, Ransomware attack 96%, and Cross-site Scripting (XSS) attack 65%. The DT classifier gives the highest precision rate for Normal traffic and two types of attacks, namely, Upload attack 100% and Cross-site Scripting (XSS) attack 65%. Therefore, we observe that the KNN classifier can give a higher recall for three types of attacks, namely, Backdoor attack 94%, OS Fingerprinting 70%, and Ransomware attack 94%. The DNN classifier can give a higher recall for five types of attacks, namely, HTTP flood DDoS attack 92%, TCP SYN Flood DDoS attack 100%, UDP flood DDoS attack 100%, MITM attack 100%, and Password cracking attack 91%. The DT classifier can give a higher recall for two types of attacks, namely, ICMP flood DDoS attack 100% and SQL Injection 96%. The SVM classifier can give a higher recall for three types of attacks, namely, Port Scanning 100%, Vulnerability scanning attack 86%, and Cross-site Scripting (XSS) attack 88%. Finally, The RF classifier can give only a higher recall for Upload attack 51%.

B. FEDERATED MACHINE LEARNING

The evaluation results of the federated deep learning approach for three types of classification, namely, 2-class (binary classification), 6-class (multi-classification), and 15-class (multi-classification), are presented in Tab. 15. In particular, the results present the three types of accuracy metric, namely, global model accuracy, worst client accuracy, and best client accuracy. These all accuracies are obtained for the first and the 10^{th} round of deep learning network



(a) Normal (Benign)

(b) Backdoor attack

(c) DDoS_HTTP attack

(d) DDoS_ICMP attack

(e) DDoS_TCP attack

(f) DDoS_UDP attack

(g) Fingerprinting attack

(h) MITM attack

(i) Password attack

(j) Port_Scanning attack

(k) Ransomware attack

(l) SQL_injection attack

(m) Uploading attack

(n) Vulnerability_scanner attack

(o) XSS attack

FIGURE 6. Features importance for each class by random forest.

under the federated learning mode. In addition, the results are obtained for two modes, namely, 1) non-independent and identically distributed (Non-IID) and 2) independent and identically distributed (IID).

For 2-class (i.e., binary classification), the best results in the first round of deep learning network under the federated learning mode are obtained when the number of clients $k = 5$ and $k = 10$ and with the mode of non-independent and identically distributed (Non-IID), where the best client accuracy achieves 99.98%, the worst client accuracy achieves 99.97%, the global model accuracy achieves 99.97%. However, with the 10^{th} round of deep learning network, the best results are obtained when the number of clients $k = 5$, $k = 10$ and, $k = 15$ and with the mode of independent and identically distributed (IID), where the best client accuracy achieves 100%, the worst client accuracy achieves 99.99%, and the global model accuracy achieves 100%.

For 6-class (i.e., multi-classification), the best results in the first round of deep learning network under the federated learning mode are obtained when the number of clients $k = 10$ and with the mode of independent and identically distributed (IID), where the best client accuracy achieves 95.33%, the worst client accuracy achieves 95.26%, and the global model accuracy achieves 95.34%. With the 10^{th} round, the best results of deep learning network are obtained when the number of clients $k = 10$ and with the mode of independent and identically distributed (IID), where the best client accuracy achieves 96.00%, the worst client accuracy achieves 95.89%, and the global model accuracy achieves 95.99%.

For 15-class (i.e., multi-classification), the best results in the first round of deep learning network under the federated learning mode are obtained when the number of clients $k = 15$ and with the mode of independent and identically distributed (IID), where the best client accuracy achieves 93.00%, the worst client accuracy achieves 93.02%, and the global model accuracy achieves 93.22%. With the 10^{th} round, the best results of deep learning network are obtained when the number of clients $k = 15$ and with the mode of independent and identically distributed (IID), where the best client accuracy achieves 93.38%, the worst client accuracy achieves 92.91%, and the global model accuracy achieves 93.37%.

From these results, we first observe that with federated deep learning, the performance of all global models are able to approximate the centralized model's performance. The second finding is that under the IID data distribution strategy, the Best, Worst, and Global models are strongly matched to each other in a consistent manner throughout all parameters and datasets (i.e., 2-class, 6-class, 15-class). The third observation is that with the Non-IID case, the clients are able to benefit from the federated learning strategy. A clear illustration of a good example is with a 15-class dataset, where $K = 15$, the best accuracy of the client was 71.42%, but with 10^{th} of federated learning rounds, the client achieved an accuracy of 91.74%.

VII. CONCLUSION

In this paper, we proposed a new comprehensive realistic cyber security dataset of IoT and IIoT applications, called Edge-IIoTset, that cyber security researchers can use to evaluate their proposed machine learning-based intrusion detection systems in two different modes, namely, centralized and federated learning. The proposed testbed is organized into seven layers, including, Cloud Computing Layer, Network Functions Virtualization Layer, Blockchain Network Layer, Fog Computing Layer, Software-Defined Networking Layer, Edge Computing Layer, and IoT and IIoT Perception Layer. It addresses the limitations of the current data sets and is appropriate for the key requirements of IoT and IIoT applications, where we provided new emerging technologies In each layer, such as ThingsBoard IoT platform, OPNFV platform, Hyperledger Sawtooth, Digital twin, ONOS SDN controller, Mosquitto MQTT brokers, Modbus TCP/IP, etc. The IoT data are generated from various IoT devices (more than 10 types). This dataset is analyzed using a primary exploratory data analysis with the performance of machine learning approaches in both centralized and federated learning modes.

REFERENCES

[1] *Edge-Iiotset Dataset*. Accessed: Jan. 15, 2022. [Online]. Available: http://ieee-dataport.org/8939

[2] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6822–6834, Aug. 2019.

[3] N. Moustafa, M. Keshky, E. Debiez, and H. Janicke, "Federated TON_IoT Windows datasets for evaluating AI-based security applications," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Los Alamitos, CA, USA, Dec. 2020, pp. 848–855.

[4] *THE INTERNET OF THINGS 2020: Here's What Over 400 IoT Decision-Makers Say About the Future of Enterprise Connectivity and How IoT Companies Can Use it to Grow Revenue*. Accessed: Jan. 3, 2022. [Online]. Available: https://www.businessinsider.com/internet-of-things-report?IR=T

[5] *THE INTERNET OF THINGS 2020: Here's What Over 400 IoT Decision-Makers Say About the Future of Enterprise Connectivity and How IoT Companies Can Use it to Grow Revenue*. Accessed: Jan. 3, 2022. [Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/internet-of-thin%gs-iot-market-100307

[6] J. Sengupta, S. Ruj, and S. Das Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT," *J. Netw. Comput. Appl.*, vol. 149, Jan. 2020, Art. no. 102481.

[7] *Kaspersky: Attacks on IoT Devices Double in a Year*. Accessed: Jan. 3, 2022. [Online]. Available: https://iottechnews.com/news/2021/sep/07/kaspersky-attacks-on-iot-devices%-double-in-a-year/

[8] *Combating Ransomware. A Comprehensive Framework for Action: Key Recommendations from the Ransomware Task Force*. Accessed: Jan. 3, 2022. [Online]. Available: https://securityandtechnology.org/wp-content/uploads/2021/04/IST-Ransomwa%re-Task-Force-Report.pdf

[9] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Oct. 2018.

[10] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, 2019.

[11] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a new dataset for machine learning techniques on MQTT," *Sensors*, vol. 20, no. 22, p. 6578, Nov. 2020.

[12] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3962–3977, Mar. 2022.

[13] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.

[14] M. A. Ferrag, L. Shu, O. Friha, and X. Yang, "Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 3, pp. 407–436, Mar. 2022.

[15] I. Hafeez, M. Antikainen, A. Y. Ding, and S. Tarkoma, "IoT-KEEPER: Detecting malicious IoT network activity using online traffic analysis at the edge," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 45–59, Mar. 2020.

[16] S. Verma, Y. Kawamoto, and N. Kato, "A smart internet-wide port scan approach for improving IoT security under dynamic WLAN environments," *IEEE Internet Things J.*, early access, Dec. 3, 2021, doi: 10.1109/JIOT.2021.3132389.

[17] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, "Federated deep learning for cyber security in the Internet of Things: Concepts, applications, and experimental analysis," *IEEE Access*, vol. 9, pp. 138509–138542, 2021.

[18] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, and R. G. D'Oliveira, "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.

[19] Y. Al-Hadhrami and F. K. Hussain, "Real time dataset generation framework for intrusion detection systems in IoT," *Future Gener. Comput. Syst.*, vol. 108, pp. 414–423, Jul. 2020.

[20] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset)," in *Proc. Int. Netw. Conf.* Springer, 2020, pp. 73–84.

[21] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.

[22] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018.

[23] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2921–2926.

[24] *Thingsboard Open-Source IoT Platform*. Accessed: Dec. 29, 2021. [Online]. Available: https://thingsboard.io/

[25] *OPNFV: Open platform for NFV*. Accessed: Dec. 29, 2021. [Online]. Available: https://www.opnfv.org/

[26] *Hyperledger Sawtooth*. Accessed: Dec. 29, 2021. [Online]. Available: https://sawtooth.hyperledger.org/

[27] *Eclipse Ditto*. Accessed: Dec. 29, 2021. [Online]. Available: https://www.eclipse.org/ditto/

[28] *ONOS*. Accessed: Dec. 29, 2021. [Online]. Available: https://opennetworking.org/onos/

[29] *Eclipse Mosquitto*. Accessed: Dec. 29, 2021. [Online]. Available: https://mosquitto.org/

[30] *Node-Red Modbus TCP*. Accessed: Dec. 29, 2021. [Online]. Available: https://flows.nodered.org/node/node-red-contrib-modbus

[31] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Mach. Learn. Res.*, Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.

MOHAMED AMINE FERRAG received the bachelor's, master's, Ph.D., and Habilitation degrees in computer science from Badji Mokhtar—Annaba University, Annaba, Algeria, in June 2008, June 2010, June 2014, and April 2019, respectively.

Since October 2014, he has been a Senior Lecturer with the Department of Computer Science, Guelma University, Guelma, Algeria. Since July 2019, he has been a Visiting Senior Researcher with the NAU-Lincoln Joint Research Center of Intelligent Engineering, Nanjing Agricultural University, Nanjing, China. His research interests include wireless network security, network coding security, and applied cryptography. He has published over 90 papers in international journals and conferences in the above areas. He has been conducting several research projects with international collaborations on these topics. His current H-index is 24, i10-index is 41, and 3234 citations in Google Scholar Citation. He was a recipient of the 2021 IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT Best Paper Award. He is featured in Stanford University's list of the world's Top 2 % scientists for the years 2019 and 2020. Some of his research findings are published in top-cited journals, such as the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, IEEE ACCESS, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, *Sensors* (MDPI), *Journal of Information Security and Applications* (Elsevier), *Transactions on Emerging Telecommunications Technologies* (Wiley), *Telecommunication Systems* (Springer), *International Journal of Communication Systems* (Wiley), *Sustainable Cities and Society* (Elsevier), and *Journal of Network and Computer Applications* (Elsevier). He is currently serving on various editorial positions, such as Editorial Board Member in journals (Indexed SCI and Scopus) such as *ICT Express* (JCR IF 4.317), *IET Networks* (Citescore 4.1), *International Journal of Internet Technology and Secured Transactions* (Citescore 1.0), *Security and Communication Networks* (JCR IF 1.791), and *Journal of Sensor and Actuator Networks* (Citescore 6.2). He reviewed more than 1000 papers (verified by publons) for top-cited journals, including, *Nature*, IEEE TRANSACTIONS, Elsevier, Springer, and Wiley journals.

OTHMANE FRIHA received the master's degree in computer science from Badji Mokhtar—Annaba University, Algeria, in 2018, where he is currently pursuing the Ph.D. degree. His current research interests include network and computer security, the Internet of Things, and applied cryptography.

DJALLEL HAMOUDA received the master's degree in computer science from Guelma University, Guelma, Algeria, in 2020, where he is currently pursuing the Ph.D. degree. His current research interests include cyber security and industrial networks.

LEANDROS MAGLARAS (Senior Member, IEEE) received the B.Sc. (M.Sc. equivalent) degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 1998, the M.Sc. degree in industrial production and management and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Thessaly, in 2004, 2008, and 2014, respectively, and the Ph.D. degree in intrusion detection (SCADA systems) from the University of Huddersfield, in 2018. From September 2017 to November 2019, he was the Director of the National Cyber Security Authority of Greece. He is currently a Full Professor of cybersecurity with the School of Computer Science and Informatics, De Montfort University. He is an author of more than 160 papers in scientific magazines and conferences. He is featured in Stanford University's list of the world Top 2% scientists.

HELGE JANICKE (Member, IEEE) is currently the Research Director of the Cyber Security Cooperative Research Centre, Australia. He is affiliated with Edith Cowan University and holds a visiting professorship in cyber security at De Montfort University, U.K. He established DMU's Cyber Technology Institute and its Airbus Centre of Excellence in SCADA Cyber Security and Forensics Research. He has been the Head of the School of Computer Science, De Montfort University, U.K., before taking up his current position as the Research Director for the Cyber Security Cooperative Research Centre. He has founded the International Symposium on Industrial Control System Cyber Security Research (ICS-CSR) and contributed over 150 peer-reviewed articles and conference papers to the field that resulted from his collaborative research with industry partners such as Airbus, BT, Deloitte, Rolls-Royce, QinetiQ, and General-Dynamics. His research interests include cyber security, in particular with applications in critical infrastructures using cyber-physical systems, SCADA, and industrial control systems. His current research investigates the application of agile techniques to cyber incident response in critical infrastructure, managing human errors that lead to cyber incidents, and research on cyber warfare and cyber peacekeeping.

• • •