Transferability of Machine Learning Algorithm for IoT Device Profiling and Identification

Priscilla Kyei Danso, Sajjad Dadkhah, *Member, IEEE*, Euclides Carlos Pinto Neto, *Member, IEEE*, Alireza Zohourian, Heather Molyneaux, Rongxing Lu, *Fellow, IEEE*, and Ali A. Ghorbani, *Senior Member, IEEE*

***Abstract*—The lack of appropriate cyber security measures deployed on Internet of Things (IoT) makes these devices prone to security issues. Consequently, the timely identification and detection of these compromised devices become crucial. Machine learning (ML) models which are used to monitor devices in a network have made tremendous strides. However, most of the research in profiling and identification uses the same data for training and testing. Hence, a slight change in the data renders most learning algorithms to work poorly. In this article, we study a transferability approach based on the concept of transductive transfer learning for IoT device profiling and identification. Notably, this type of transfer learning works by explicitly assigning labels to the test data in the target domain by using the test feature space in the target domain, with training data from the source domain. Specifically, we propose a three-component system comprising: 1) the device type identification; 2) the vulnerability assessment; and 3) the visualization module. The device type identification component uses the underlying concept of transductive transfer learning where the trained model is transferred to a remote lab for testing. A variety of ML models are evaluated with respect to accuracy, precision, recall, and F1-score in order to determine which are the most suitable for the proposed transferability profiling. Furthermore, the vulnerability of the predicted device type is also assessed by using three vulnerability databases: 1) Vulners; 2) National Vulnerability Database (NVD); and 3) IBM X-Force. Finally, the results from the vulnerability assessment are visualized and displayed on a dashboard.**

***Index Terms*—Internet of Things (IoT), machine learning (ML), security, transferability, visualization, vulnerability assessment.**

I. Introduction

IT IS gradually becoming apparent that the Internet of Things (IoT) is bringing about a paradigm shift in the digital age [23]. IoT has been regarded as a technological revolution that is affecting all facets of life and businesses [1]. According to estimates, there will be 75 billion IoT-connected devices in use by the year 2025, making IoT a part of our everyday life [9]. Between 2021 and 2025, the global deployment of IoT devices is predicted to reach 30.9 billion units, up from 13.8 billion deployed in 2021. Smart home appliances, connected industrial equipment, and connected automobiles are examples of common IoT connections.

Compared to IoT connections, non-IoT connections, such as those between smartphones, desktops, and laptops, are expected to total 10 billion by 2025, which is one-third less than connections between IoT devices [8]. With these staggering statistics, McKinsey Global Institute projects that by 2025, the potential economic impact might reach $11.1 trillion [25]. Nevertheless, numerous security and privacy risks are pervasively ingrained by the enormous surge of digitally linked devices sharing large volumes of data [24]. IoT devices are also vulnerable to a variety of harmful passive and active attacks that compromise their security and privacy, which might potentially impair their operation [40]. Particularly as IoT becomes prevalently interwoven in the day-to-day life of people, users must have confidence that these devices are safe from threats [33].

IoT communications must also be protected since large volumes of data can be transmitted between devices as well as between those devices and servers or customers. If the confidentiality, integrity, or availability (CIA) data is compromised, it could have catastrophic consequences. Inadequately protected data streams on these devices act as possible access for attack and vulnerability to data theft [33].

Because of the power consumption, memory usage, connectivity, and battery or power availability, IoT devices cannot offer the same level of security as traditional computers. In this sense, machine learning (ML) models used to monitor devices in a network and make predictions by differentiating between normal and illegitimate devices have made tremendous strides.

This article proposes the use of the ML algorithm based on the concept of transferability for IoT device type identification. This transferability significantly improves the learning algorithm performance while avoiding costly target data labeling. The main contributions of this research are as follows.

1) An implementation of a complete system comprising device identification and profiling, vulnerability assessment, and visualization dashboard module.

Manuscript received 28 September 2022; revised 17 February 2023; accepted 27 June 2023. Date of publication 5 July 2023; date of current version 8 January 2024. This work was supported in part by the Canadian Institute for Cybersecurity (CIC); in part by the National Research Council of Canada's Artificial Intelligence for Logistics Program; and in part by NSERC under Discovery Grant RGPIN 231074. The work of Ali A. Ghorbani was supported by the Tier 1 Canada Research Chair. *(Corresponding author: Priscilla Kyei Danso.)*

Priscilla Kyei Danso, Sajjad Dadkhah, Euclides Carlos Pinto Neto, Alireza Zohourian, Rongxing Lu, and Ali A. Ghorbani are with the Canadian Institute for Cybersecurity, Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: pdanso@unb.ca; sdadkhah@unb.ca; e.neto@unb.ca; alireza.zohourian@unb.ca; rlu1@unb.ca; ghorbani@unb.ca).

Heather Molyneaux is with the Cybersecurity Team, Digital Technologies Research Center, National Research Council, Fredericton, NB E3B 9W4, Canada (e-mail: heather.molyneaux@nrc-cnrc.gc.ca).

Digital Object Identifier 10.1109/JIOT.2023.3292319

2) A demonstration of the use of the concept of transductive transfer learning for IoT device identification and profiling.
3) An evaluation of the proposed transferability approach on three publicly available IoT data sets: a) Canadian Institute for Cybersecurity (CIC) data set (for training); b) IMC 2019 Payload data set (for testing); and c) IoT sentinel (for testing).
4) A vulnerability assessment of the IoT device identified and profiled by the classifier.
5) An implementation of a visualization dashboard to display vulnerability assessed on each IoT device type or IoT vendor.

The remainder of this article has been organized as follows. Section II presents previous works in the domain of IoT device profiling and identification as well as vulnerability assessment and visualization. We further explain the terminology "transferability" which our research is hinged on in Section III. Section IV describes the overall framework which compromises a three-component system solution for an IoT architecture. The experimental system flow of the proposed device type identification framework is reviewed in Section V. A thorough description of the CIC IoT data set 2022, IMC 2019 Payload data set, and IoT Sentinel data set are done in Section VI. Section VII presents the experimental results of the proposed models. Section VIII assesses the process flow of the vulnerability assessment used in this study. Section IX outlines the visualization of the vulnerability assessed for each device. Finally, Section X concludes our work.

II. Related Work

In order to identify unauthorized IoT devices connected to a network, Meidan et al. [26] used TCP/IP traffic network data for categorization by ML. The authors assume that the data set adequately reflected each device type on the white list. In order to effectively identify IoT device types from the white list, features from network traffic data were extracted using supervised ML, more especially random forest (RF). Nine different IoT device categories totalling 17 unique IoT devices were gathered and manually labeled in order to train and test multiclass classifiers. 60 features were generated and eliminated several based on the following criteria: zero variance, irrelevance, or a propensity for model overfitting; and only used the list of top characteristics to profile and identify the device type. One IoT device type was excluded from the white list to signify an unauthorized type. In order to verify the classifier's transportability, the model was trained on two security cameras of the same type (SimpleHomeXCS71001) in Lab A and tested with a different security camera in Lab B. (Withings WBP02WT9510). The trained classifiers obtained an average of 96% accuracy in the detection of illegitimate IoT device types.

Anand et al. [2] incorporated vulnerability analysis and power storage techniques. This integration is intended to offer a workable way to overcome the difficulties in smart agriculture. The collector, the analyzer, the quantifier, and the mitigator make up the vulnerability assessment component. The collector uses Nmap to acquire data about the network and IoT devices. The analyzer juxtaposes the data from the National Vulnerability Database (NVD) using the data it has collected as input. The quantifier assesses the vulnerabilities based on the risk that a certain IoT poses to the system using the CVSSv3 metric from the analyzer. Using the vulnerabilities and the associated score, the mitigator proposes ways to resolve the vulnerability. The mitigator also provides manufacturers and IoT users with information about vulnerabilities and their remedies. In the event that known vulnerabilities are mitigated promptly, the IoT device is less likely to become the target of the potential adversary.

Fürst et al. [13] proposed that information infusion aims to quickly transfer knowledge used during the training phase across tasks rather than transferring taught knowledge to another task (as in transfer learning). In order to accommodate asynchronously to a new situation by querying various knowledge sources, the authors used adaptive weak and strong knowledge functions that interface with a knowledge graph and external knowledge sources. Hence, weak and strong knowledge functions were proposed as different forms of knowledge functions. A "white-box" knowledge model was created by combining these knowledge functions into a weak ensemble and a strong ensemble. The model was trained and tested using a variety of classifiers, with RFs showing the best results with an F1-score of 80.2%.

Multilayer vulnerability keywords matching is a method that Lou and Wang [21] developed to assess the studied vulnerabilities of IoT devices. From the IoT device, Nmap was used for reconnaissance. Identifying open ports and the services that are currently using them requires the usage of Nmap. To accurately extract vulnerability search terms, the authors use two techniques.

1) First, information on the class of products, operating systems, and software versions affected by the vulnerability are taken from the CPE during the usual extraction step.
2) The second stage involves retrieving additional keywords from the common vulnerabilities and exposure (CVE) description by analyzing its syntactic patterns.

III. Transferability

Transfer is defined as a model, piece of knowledge, or theory about behavior established in one domain that is applied to characterize the related behavior in another situation. Hence, Koppelman and Wilmot [19] defined the transferability of the model as the application of a model that was developed and evaluated in one domain to another. It follows that a model may be transferrable if it can offer helpful data about the behavior or phenomena relevant to the application domain. When models are evaluated on historical data and employed to make predictions about the future, the transferability characteristic is frequently triggered unintentionally.

The common assumption in transfer learning is that the training and testing data sets share a certain level of commonality and identifying such common structures is of key importance [31]. We identify this commonality using transductive

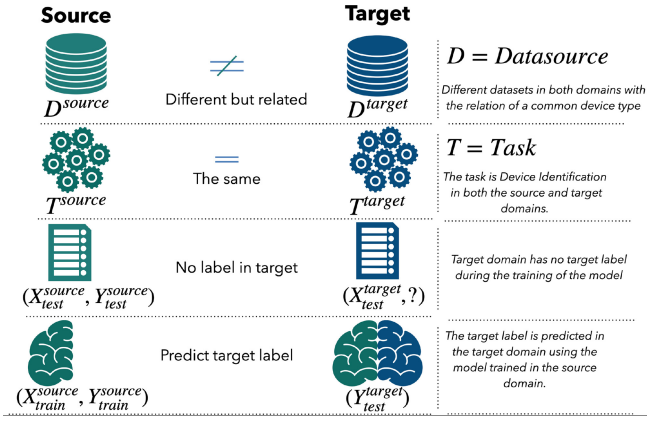

Fig. 1. Transductive transfer learning concept.

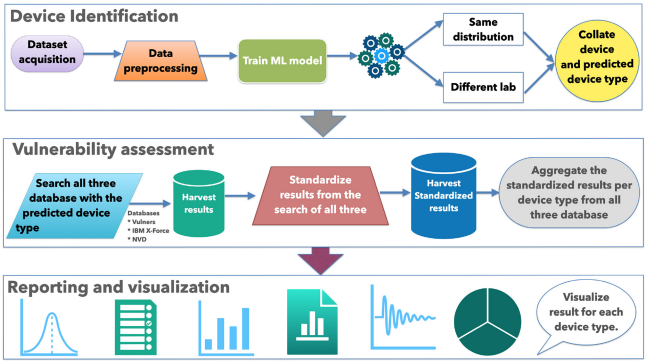

Fig. 2. Overall system framework of the proposed system.

transfer learning to identify the IoT device types in the network. Transductive transfer learning in simple terms refers to situations where the label information only comes from the source domain [41], the source and target tasks are the same, while the source and target domains are different [30]. Fig. 1 clearly encapsulates the concept of transductive transfer learning.

Transductive transfer learning assumes that X_{train} and X_{test} are from the same distribution, D. The transferability approach put forward in this research uses data sets different from the data used for training. The data set in both the source and target domains are considered but related. This means that despite having the common device type category "Camera" in both the source and the target the devices that make up that category would be different. The source domain can have devices like Amcrest camera and Luohe camera, which belong to the Camera category. In contrast, the target domain can have devices like the Lefun camera and Yi camera comprising devices in the camera category. Hence, the data sources are different but related in terms of the device type category. As a result, we presume that data can be derived from two distinct distributions: 1) D^{source} and 2) D^{target}. This particular type of transfer learning works by explicitly assigning labels $Y_{\text{test}}^{\text{target}}$ to test data $X_{\text{test}}^{\text{target}}$ which is typically derived from D^{target}, with training data ($X_{\text{train}}^{\text{source}}$, $Y_{\text{train}}^{\text{source}}$) drawn from D^{source}. The translation of this concept and its application to our study is as follows.

1) The goal in the target domain with unlabeled is to predict the device type using the knowledge from the trained model in the source domain.
2) The knowledge from the task performed in the source domain is transferred to the target domain.
3) The task performed in this study using transductive transfer learning is IoT device type identification.
4) The data in the source and target domain are different but related with device type being commonality.

IV. Overall Proposed Framework

The framework proposed in this research is composed of three components: 1) the device identification and profiling; 2) vulnerability assessment; and 3) the visualization dashboard module illustrated (Fig. 2). The motivation for the proposed framework is to have a compound system responsible for identifying the devices and device types in a network using ML and simultaneously assessing and visualizing the vulnerabilities of the predicted device type.

Algorithm 1 highlights the lists of formal steps of our proposed framework, and below is a brief description.

1) Extract features from each IoT device.
2) Features extracted are used to train the ML model after data preprocessing.
3) If the data set is from the same distribution, train with the percentage (70%) of the ML model with a percentage and test the trained model with the remaining percentage (30%).
4) Assuming the training data (source) is a different distribution from the testing data (target), use the trained model from the source domain in the device identification of the target domain.
5) For each instance in steps c and d the predicted device type by the classifier is collated.
6) The predicted device type is used as a keyword to search all three databases (NVD, IBM X-Force, and Vulners) for vulnerability assessment.
7) The results from the search are harvested in a local database.
8) For uniformity, the harvested search results from IBM X-Force, and Vulners are standardized using the NVD database.
9) The standardized results in then grouped per device type, i.e., audio, camera, home automation, and smart hub.
10) The resultant aggregate is visualized and displayed on a dashboard.

V. Experimental System Flow of the Proposed Device Type Identification Framework

This study proposes a transferability approach based on the underlying concept of transductive transfer learning by employing the use of several supervised ML techniques to train a model in one lab (source) and transfer the knowledge of the trained model to another lab (target) with the aim of enhancing the learning process.

In this section, an overview of the proposed device type identification framework is analyzed. A detailed explanation

Algorithm 1: Algorithm for the Proposed Framework

Input: IoT device feature set
Output: Vulnerability of the IoT devices
Data: Testing set x(same distribution) and y(different distribution)

$DL = dl1, dl2, \cdots, dl_n$ /* where each element represents the feature set per device. */
Function MLTraining(*deviceFeatures*):
  if *dataset from the same distribution* **then**
    trainedmodel =train(dl_i in *data*(*DS*)) /* Train ML model with 70% of the data */
    predicteddevicetypes = *trainedmodel*(*x*) /* Test trained ML model 30% of the same data used for training */
  else if *dataset is from different distribution* **then**
    predicteddevicetypes = *trainedmodel*(*y*)
    /* Test the trained ML model with another dataset */
  else
    None
  return *predicteddevicetypes*
    /* Collate device and predicted device type identification. */
Function VulnerAssess(*predicteddevicetype*):
  foreach *predicteddevicetype in predicteddevicetype* **do**
    vulsearch = searchDB(*predicteddevicetypes*)
    /* Search all three databases(NVD, IBM X-Force, and Vulners) for vulnerabilities with the predicted device type as a keyword. */
    harvestVul = harvestResults(*vulsearch*)
    /* Store all search results from the three databases(NVD, IBM X-Force, and Vulners) into a local database */
    standardizeSearch = standardSearch(*harvestVul*)
    /* Standardize the harvested IBM X-Force, and Vulners search results by using the NVD database for uniformity */
    aggregateVul = aggregate(*standardizeSearch*)
    /* Aggregate the standardized results per device type from all three standardized databases. */
  return *aggregateVul*
Function VisualizeVul(*aggregateVul*):
  foreach *vulneraility in aggregateVul* **do**
    visualVul = displayVisualization(*vulneraility*)
  return *visualVul*

of the feature selection and classifier modeling methods used is provided. Fig. 3 depicts the proposed device type identification framework. The framework is divided into four modules: data preprocessing, feature selection, model training, and transferability of the classifier.

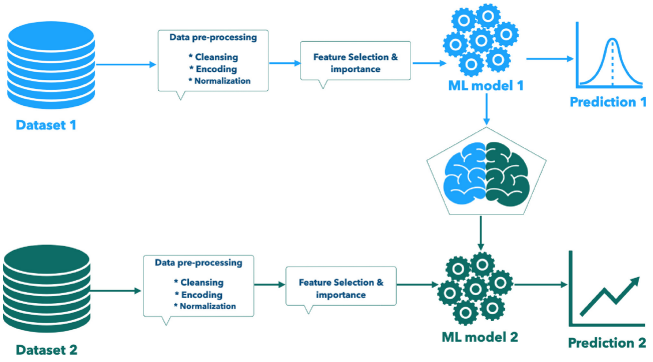

Fig. 3. Proposed device type identification framework.

The following is a summary of each module.

1) *Data Preprocessing:* During this stage, the feature set is synthesized statistically to ensure that the selected preprocessing approach is relevant to accommodate missing values. Before training, the preprocessed feature set is scaled into a regularized range using normalization. Additionally, the feature set is transformed and encoded for categorical values. To improve the overall system performance of our approach, we extracted a collection of the most statistically important features for our device identification and profiling framework.
2) *Feature Selection Module:* This phase is responsible for reducing the number of feature sets used for training the ML model with the goal of lowering computational cost and increasing the efficiency of the model. This happens by selecting a fraction of the important feature set and only using those relevant features to train the ML model.
3) *Model Training:* This study employs a supervised ML technique to profile and identify the different types of devices in the network. Several ML models are investigated to determine the most effective method for our approach.
4) *Transferability of Classifier:* This research employs a transferability approach for device type identification based on the concept of transductive transfer learning. To test and predict the target label that corresponds to the device type, the knowledge from the trained model in the source domain is transferred to the target domain.

A. Data Preprocessing

Because of the concept of "garbage in, garbage out," data preprocessing is a crucial stage in any ML task. This means that using unclean data to train your model will inevitably result in an incorrectly trained model that is irrelevant to the analysis. All deficient data records for the measured variables in the data set are handled at this stage. To scale our data into a regularized range, we used Z-score normalization, also known as standardization. Z-score normalization rescales features in accordance with the standard normal distribution property with $\mu = 0$ and $\sigma = 1$, where μ is the mean (average) and σ is the standard deviation from the mean. $z = ([x_i - \mu]/\sigma)$ is used to compute the standard score or z-score of the sample. One of the common obstacles faced in any ML problem is the unequal number of samples in each class distribution. This

scenario, known as the imbalanced problem, has an impact on the performance of many ML algorithms. A model trained with such a data set is mostly biased toward the minority target label in the data set due to their fewer number hence misclassifying the minority as belonging to a majority class. To prevent overfitting, an equal number of records are sampled from each class in the target variable during the training of the classifier.

B. Feature Selection

Finding a subset of input variables that can effectively represent all of the input data while minimizing the influence of noise or unnecessary attributes yet delivering accurate prediction results is the aim of feature selection [16]. Additionally, feature selection lowers computation demands, lessens the impact of the "curse of dimensionality" and enhances generalization performance [7]. In this study, the analysis of variance (ANOVA) feature selection method was employed. ANOVA is a data analysis technique in which one or more dependent variables are evaluated under different conditions identified that are determined by one or more measurements [36]. ANOVA presupposes that the variables have a Gaussian distribution and that there is a linear relationship between the feature and the target [12]. F-test is employed in ANOVA to determine whether there are any significant differences between the groups. The F-ratio of an ANOVA will be close to 1 if there is no significant contrast between the groups and all variances are identical. Equal variation across groups indicates that this attribute has no effect on response and cannot be used in model training. The SelectKBest module chooses the best feature set for the training using the score values from the ANOVA module.

C. Model Training

In this phase, six supervised ML techniques will be used in training our model to profile and identify the different device types. Decision tree (DT) [15], [20], RF [6], gradient boosting machine (GBM) [29], support vector machine (SVM) [17], *k*-nearest neighbor (*k*NN) [37], and multilayer perceptron (MLP) [14] are the ML models explored in this study.

D. Transferability of Classifiers

In this study, the IoT traffic data was collected from three completely different labs located in distant countries, denoted as Lab A—CIC Lab (source), Lab B—Mon(IoT)r Lab (target), and Lab C—IoT Sentinel Lab (target). The data set for Lab B and C is to verify the proposed transferability approach for device profiling and identification. Additionally, each of these labs has a completely different network configuration and is populated with different devices. However, there were some devices that were common in the source and each of the target labs and not a single common device in all three labs. It is important to highlight that our work may be impacted by the various privacy and security compliance legislation of these different labs situated in completely different locations from each other, such as GDPR for the EU, and FTC for the U.S. Lab A which is used for training is located in Canada; Lab B is located in the U.S. and Lab C is located in Europe.

Using the input variables from each of the target sources and the trained model from the source domain, the objective at this stage is to assign labels, which are the device types, to each of these target feature sets. As a result, the trained model from the source will be used for testing rather than retraining the model in the target domain. We compared the performance of classifiers when they were trained on IoT device data acquired in one lab and then evaluated data collected from other IoT devices as part of our analyses of the transferability using a range of various IoT device types from each lab.

VI. Data Set Description

The data sets utilized for the evaluation of the transferability approach for device type identification based on the idea of transfer learning are detailed in this section.

A. CIC IoT Data Set 2022

The CIC IoT data set 2022[1] was generated by capturing the network traffic by running the following experiments [35].

1) *Power:* Each device was powered on individually and started capturing the network traffic separately.
2) *Idle:* During what is termed as idle time, the whole network traffic was recorded from late at night to early in the morning. The entire lab was evacuated during this time, and no human interactions ensued.
3) *Interactions:* Every conceivable functionality of IoT devices has been extracted, along with the related network activity and transmitted packets for each functionality or activity.
4) *Scenarios:* The network activity inside a smart house was simulated using a collection of devices in six distinct types of scenario experiments. These tests were conducted to observe how several devices functioned simultaneously.
5) *Active:* All day long, the entire network communications were captured. Users were permitted to use the lab to generate both active and passive traffic for devices interacted with.
6) *Attacks:* Two distinct attacks, Flood and RTSP-Brute Force were conducted on certain devices while simultaneously recording the network data generated by these attacks.

B. IMC 2019 Payload Data Set

Network traffic from 81 devices recorded during manual and automated trials makes up the data set.[2] Power, interaction, idle, and uncontrolled experiments were the four types of experiments that were carried out [32].

1) *Power* experiments involve turning on the device and capturing the network traffic.
2) *Interaction* experiments capture the network transfer while devices are actively in use.

[1]https://www.unb.ca/cic/datasets/iotdataset-2022.html
[2]https://github.com/NEU-SNS/intl-iot

TABLE I
DEVICES IN THE DIFFERENT LABS AND THE CORRESPONDING DEVICE TYPE

	CIC IoT dataset 2022	
No.	**Device Name**	**Device Type**
01.	Amazon Echo Dot	Audio
02.	Amazon Echo Spot	Audio
03.	Amazon Echo Studio	Audio
04.	Google Nest Mini	Audio
05.	Sonos One	Audio
06.	Amcrest Camera	Camera
07.	ArloQ Camera	Camera
08.	Borun Camera	Camera
09.	DLink Camera	Camera
10.	HeimVision Camera	Camera
11.	HomeEye Camera	Camera
12.	Luohe Camera	Camera
13.	Nest Camera	Camera
14.	Netatmo Camera	Camera
15.	SimCam	Camera
16.	Arlo Base Station Camera	Camera
17.	Amazon Plug	Home Automation
18.	Globe Lamp	Home Automation
19.	Gosund Plug	Home Automation
20.	Heimvision Lamp	Home Automation
21.	Teckin Plug	Home Automation
22.	Yutron Plug	Home Automation
23.	D-Link Water sensor	Home Automation
24.	Philips Hue	Smart Hub
25.	Ring Basestation	Smart Hub
26.	Eufy Homebase	Smart Hub
27.	Atomi coffee maker	Appliance
28.	iRobot roomba	Appliance
29.	Smart board	Appliance

	IMC 2019 Payload dataset	
No.	**Device name**	**Device type**
01.	Amcrest Cam	Camera
02.	Blink Cam	Camera
03.	Lefun Cam	Camera
04.	Luohe Cam	Camera
05.	Microseven Cam	Camera
06.	Wansview Cam	Camera
07.	Yi Cam	Camera
08.	Sengled hub	Smart Hub
09.	Smartthings	Smart Hub
10.	Blink Hub	Smart Hub
11.	Insteon hub	Smart Hub
12.	Wink 2 hub	Smart Hub
13.	Philips Hue	Smart Hub
14.	Flux Bulb	Home Automation
15.	TP-Link Bulb	Home Automation
16.	Wemo Plug	Home Automation
17.	Amazon Echo Dot	Audio
18.	Amazon Echo Spot	Audio
19.	Amazon Echo Plus	Audio
20.	Google Home Mini	Audio
21.	Behmor Brewer	Appliance
22.	Samsung Washer	Appliance
23.	Xiaomi Cleaner	Appliance
24.	Xiaomi Rice Cooker	Appliance

	IoT Sentinel	
No.	**Device name**	**Device type**
01.	D-Link Cam	Camera
02.	D-Link Day Cam	Camera
03.	EdimaxCam 1	Camera
04.	EdimaxCam 2	Camera
05.	EdnetCam 1	Camera
06.	EdnetCam 2	Camera
07.	D-Link Switch	Smart Hub
08.	Hue Switch	Smart Hub
09.	Ednet Gateway	Smart Hub
10.	Max Gateway	Smart Hub
11.	WeMo Insight Switch 1	Smart Hub
12.	WeMo Insight Switch 2	Smart Hub
13.	WeMo Switch 1	Smart Hub
14.	WeMo Switch 2	Smart Hub
15.	Lightify	Smart Hub
16.	HomeMatic plug Switch	Smart Hub
17.	D-Link Home hub	Smart Hub
18	TP-LinkPlugHS210	Home Automation
19	TP-LinkPlugHS110	Home Automation
20	Edimax Plug 1	Home Automation
21	Edimax Plug 2	Home Automation
22	Smarter coffee	Appliances

3) *Idle* experiments record the network traffic of an IoT device not in use.
4) *Uncontrolled* experiments allowed 36 research participants the opportunity to use IoT devices for their intended use in a studio apartment in the U.S. Lab only.

C. IoT Sentinel Data Set

IoT Sentinel data set[3] consists of a selection of daily used IoT devices from the European market to conduct this research. The network configuration took into account both WiFi and Ethernet-connected devices that joined directly to the network as well as ZigBee and *Z*-Wave devices connected to the network using an intermediary hub. Using an app given by the vendor, the IoT device was configured to connect directly over WiFi or Ethernet, and the network's WiFi credentials were sent to the device via this connection. After that, the device resets and uses the supplied credentials to access the network. The authors concentrated on tracking the transfer or intermediary traffic produced by the IoT gateway for the ZigBee and *Z*-Wave devices. The normal device setup procedure was carried out 20 times for each tested device in order to collect enough fingerprints [27].

VII. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experiments

This section evaluates the data collection, data sampling or reduction technique used in this research work as well as the feature selection technique. Additionally, we describe our approach in the training and testing of the ML models and the performance metrics used on the training data set.

TABLE II
NUMBER OF RECORDS FOR EACH DEVICE TYPE IN THE CIC DATA SET [35] AFTER DATA PREPROCESSING AND THE NUMBER OF SAMPLED RECORDS USED FOR TRAINING THE ML MODEL

No.	**Device Type**	**Total no. of records**
1	Camera	191812
2	Audio	19580
3	SmartHub	12711
4	HomeAutomation	8593
5	Appliances	1635

No.	**Device Type**	**Sampled no. of records**
1	Camera	1635
2	Audio	1635
3	SmartHub	1635
4	HomeAutomation	1635
5	Appliances	1635

1) Data Collection: CIC IoT data set 2022 [35] functions as the training data (source) while IMC 2019 Payload data set [32] and IoT Sentinel data set [27] are used to evaluate our proposed framework as the test data (target). Table I lists all the devices and their corresponding device types for each data set. 48 features were extracted from each of the pcap files of the different data sets with output as a CSV file. The CIC IoT data set 2022[4] contains the feature set that makes up the generated CSV files, as well as descriptions and definitions of each feature. The collection of features created for the source domain and the target domain are identical. The resulting CSV files are used as input for the detailed Section V-A data preprocessing stage.

2) Data Sampling: An unbalanced data set has a detrimental effect on the precision of class predictions in most classification problems [38]. This skewness was addressed in this study by randomly selecting an equal number of records for each class to guarantee that each distribution class had an equal probability of being chosen. Table II highlights the data sampled for each class in our training data set.

3) Feature Selection: The findings of the feature selection used in this study, as described in Section V-B, are reviewed in this section. Fig. 4 shows the top 20 features selected. The

[3]https://research.aalto.fi/en/datasets/iot-devices-captures

[4]https://www.unb.ca/cic/datasets/iotdataset-2022.html

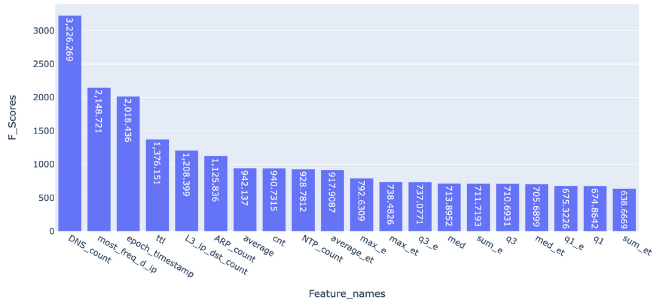

Fig. 4. 20 top features using ANOVA technique explained in Section V-B and their respective score on the CIC data set [35].

topmost feature appears to be *DNS_count* which is the Domain Name Server request made by a particular device, followed by the most frequent destination IP (*most_freq_d_ip*). The number of destination IP addresses that a device contact is an important feature because some devices have less communication, consequently, an attribute that can be utilized to distinguish one type of device from another. A camera, for example, would have very few destination IP addresses, whereas a speaker would have much more outbound communication whether in interaction, power or idle mode.

4) Training and Testing of the ML Methods: Scikit learn,[5] a Python library, is used in the implementation of the device profiling and identification component in our system. To determine which ML technique performs the best, six ML models are utilized to train and test the data set for both when the data is from the same distribution or a different distribution. When using data from the same distribution, we first cross-validate each of our six ML classifiers using a 7:3 split, which involves training on 70% of the data that was randomly chosen and testing on the remaining 30%.

5) Performance Measuring Metrics: In this section, we outline five benchmark performance indicators for assessing how well our classifiers performed on the data set with the training and test data taken from the set. Macro averaging is employed for the overall performance of precision, recall, and F1-score. For data sets with training and testing data drawn from a different distribution (lab), a different metric is used to evaluate the device identification performance. In this article, we adopted the testing metric used in [35] called the inference percentage (IP). More detail on the metrics for assessing data sets based on the same distribution is discussed in a later section.

Confusion matrix is a visual depiction of the model's performance in predicting the various classes.

Accuracy is the proportion of accurate predictions to all observations

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

Precision is calculated as the number of correctly identified device types by the number of correctly or incorrectly predicted device types by the classifier

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

[5]https://scikit-learn.org/stable/

TABLE III
PERFORMANCE OF THE SEVERAL CLASSIFIERS ON THE CIC IOT DATA SET 2022 ACTING AS THE SOURCE DOMAIN AND USED FOR TRAINING

Metric	RF	DT	GBC	KNN	SVM	MLP
Accuracy	99.87	99.46	**100**	95.18	98.50	98.52
Precision	99.62	99.48	**99.92**	95.47	96.10	98.53
Recall	99.63	99.49	**99.93**	95.23	95.69	98.58
F1-score	99.63	99.48	**99.93**	95.28	95.81	98.55
Time(sec)	1.20	**0.06**	65.97	0.98	2.20	9.33

Detection Rate (DR) shows how many actual device types the classifier classified properly

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

F1-score is a geometric average of precision and recall

$$\text{F1} - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}.$$

When a trained model is transferred to a different lab for testing, the IP is the metric for evaluation which Sajjad et al. [35] described as follows;

Inference percentage is computed for predicted IoT device. We identified some correlation with what we are terming as IP in our work with accuracy in that the classifier's prediction of the actual value is the accuracy of the device.

1) Pass a preprocessed pcap file of a specific unlabeled device to the trained model transferred to the target domain.
2) The model is to predict the device type of the specific device.
3) To ascertain the results of the prediction by the transferred model, the IP metric by Sajjad et al. [35] is used.
4) Following the prediction by the model, a unique count of the target label prediction is compiled.
5) The compilation is in the form of key-value pair, which contains the name of the particular device type and its associated count.
6) The result is sorted in decreasing order of magnitude based on the unique count.
7) The name of the device type with the greatest value is the device type that is assigned to a specific feature space.

During the device type identification in the target domain, we have modified our development based on the premise of the real-world operation of the ML pipeline. Because ML techniques are data-sensitive or data-dependent, a little variation in the inputdatais expected to significantly influence these features and, consequently, on the performance of the model and its associated predictions [5]. The assumption that data preprocessing techniques like normalization and feature selection will be employed in the target domain in a real-world transductive transfer learning setting is merely unproven. Hence, we trained the model in the source domain without normalization and feature selection and used the RF algorithm. Due to the noise-tolerant nature of RF classifier [22]

TABLE IV
DEVICE TYPE PREDICTED BY THE DIFFERENT CLASSIFIERS WITH THE CORRECTLY PREDICTED DEVICES HIGHLIGHTED IN GREEN WHEN THE TRAINED CIC IOT DATA SET 2022 WAS TRANSFERRED TO THE U.S. LAB AND TESTED WITH THE IMC 2019 PAYLOAD DATA SET [32] USING THE INTERACTION EXPERIMENT CONDUCTED

		Device type prediction Classifier					
No.	**Device name**	**RF**	**DT**	**GBC**	**KNN**	**SVC**	**MLP**
1.	Amazon Cloudcam	Camera	Camera	Home Automation	Home Automation	Smart Hub	Home Automation
2.	Amcrest Cam	Camera	Camera	Home Automation	Home Automation	Audio	Home Automation
3.	Blink Cam	Home Automation	Home Automation	Home Automation	Home Automation	Audio	Audio
4.	Lefun Cam	Camera	Camera	Home Automation	Home Automation	Smart Hub	Smart Hub
5.	Luohe Cam	Camera	Camera	Home Automation	Home Automation	Audio	Audio
6.	Microseven Cam	Camera	Smart Hub	Home Automation	Home Automation	Audio	Audio
7.	Wansview Cam	Camera	Camera	Home Automation	Camera	Audio	Audio
8.	Yi Cam	Camera	Camera	Home Automation	Home Automation	Audio	Audio
9.	Insteon	Home Automation	Home Automation	Home Automation	Home Automation	Audio	Audio
10.	Philips Hue	Audio	Home Automation	Home Automation	Home Automation	Audio	Audio
11.	Smartthings hub	Audio	Home Automation	Home Automation	Home Automation	Audio	Audio
12.	Sengled Hub	Home Automation	Home Automation	Home Automation	Home Automation	Audio	Home Automation
13.	Wink 2 hub	Camera	Camera	Home Automation	Home Automation	Audio	Audio
14.	TP-Link Plug	Camera	Home Automation	Home Automation	Home Automation	Audio	Audio
15	Flux Bulb	Home Automation	Home Automation	Home Automation	Home Automation	Home Automation	Home Automation
16.	Wemo Plug	Home Automation	Camera	Home Automation	Home Automation	Audio	Audio
17.	Amazon Echo Dot	Audio	Home Automation	Home Automation	Home Automation	Audio	Audio
18.	Amazon Echo Spot	Audio	Home Automation	Home Automation	Home Automation	Audio	Audio
19.	Amazon Echo Plus	Audio	Home Automation	Home Automation	Home Automation	Audio	Audio
20.	Google Home Mini	Camera	Home Automation	Home Automation	Audio	Audio	Audio
21.	Behmor Brewer	Home Automation	Home Automation	Home Automation	Audio	Audio	Smart Hub
22.	Samsung Washer	Audio	Home Automation	Home Automation	Home Automation	Audio	Audio
23.	Xiaomi Cleaner	Home Automation	Home Automation	Home Automation	Home Automation	Audio	Home Automation
24.	Xiaomi Rice Cooker	Home Automation	Home Automation	Home Automation	Home Automation	Audio	Home Automation

it was selected to test and predict the device types of the IoT devices.

B. Results and Analysis

1) CIC IoT Data Set 2022: Using CIC IoT data set 2022 [35] as our source domain and training data set, Table III compares the performance of the RF, DT, GBC, KNN, SVM, and MLP with respect to the performance metrics: Accuracy, Precision, Recall, F1-score, and processing time (in seconds). RF and GBC achieved an accuracy of 99.87% and 100%, respectively, while the time processing time is 1.20 and 65.97 s, respectively. GBC achieves the highest precision of 99.92% followed by RF, DT, MLP, SVM, and KNN with values 99.62%, 99.48%, 98.53%, 96.10%, and 98.53%, respectively. GBC again achieved a Recall also referred to as DR with a score of 99.93% outperforming RF, DT, MLP, SVM, and KNN. Gradient boosting is an ensemble technique which works by successively improving the performance of the classifier after every iteration. This outstanding performance on the training data set is not replicated when used in both target domains for testing, as indicated in the next section. GBC is regarded as computationally expensive and takes the longest amount of time during training because of successive iterations.

2) IMC 2019 Payload Data Set: To test the transferability technique based on the concept of transductive transfer learning proposed in this study, the IMC 2019 Payload data set was employed. The model was trained in the source domain which is located in Canada at the CIC Lab using the CIC IoT data set 2022 [35] and tested in the target domain which is located in the U.S. at Northeastern University's Mon(IoT)r Lab. Table IV demonstrates our results using the metric, IP explained in Section VII-A5.

From Table IV, the best performing classifier identifying different device types when the model was trained in one lab and transferred to a different lab for testing was the RF classifier. RF correctly predicted 12 out of the 24 devices. However, all classifiers failed to predict devices that belong to the category smart hub and appliances. The network packets generated from these devices appear to have some underlying similarities with home automation hence, almost the classifiers wrongly predict smart hub and appliances as belonging to the category home automation. Flux bulb was correctly predicted by all six classifiers while Insteon hub, Philips Hue, Sengled hub, Wink 2 hub, Behmor brewer, Samsung washer, Xiaomi cleaner, and Xiaomi rice cooker were unidentified by all six classifiers. RF correctly classified seven of eight cameras and three of four audio devices. The most diverse classifier appears to be RF classifying camera, audio, and home automation devices but failed to classify smart hub and appliances.

3) IoT Sentinel Data Set: For the verifiability of our proposed transferability approach, we test our work on the IoT Sentinel data set. The devices in this data set were purchased in the European market, hence it is assumed the lab in the study is located in Europe. Table V displays the findings for the devices drawn from the data set utilized in this study.

The data set was limited in terms of the diversity in the devices compared to that used for training. KNN appears to outperform the other classifiers in terms of the number of correctly predicted devices as well as the diversified number of predictions. KNN predicted in the category camera, smart hub, and appliance but failed to predict devices in the category home automation. MLP and SVC correctly predicted the smarter coffee which is an Appliance while RF failed to classify any of the 22 devices correctly. Compared to the IMC 2019 Payload data set [32] where RF is the best performing classifier, KNN appears to be the best performing classifier on the IoT Sentinel data set [28]. The best performing classifier in our work is defined as a classifier that correctly predicted a device type when a model was trained on the CIC IoT 2022 data set [35] and tested on [28] and [32].

TABLE V
DEVICE TYPE PREDICTED BY THE DIFFERENT CLASSIFIERS WITH THE CORRECTLY PREDICTED DEVICES HIGHLIGHTED IN GREEN WHEN THE TRAINED CIC DATA SET WAS TRANSFERRED TO THE U.K. LAB AND TESTED WITH THE IOT SENTINEL DATA SET [28]

		Device type prediction Classifier					
No.	**Device name**	**RF**	**DT**	**GBC**	**KNN**	**SVC**	**MLP**
1.	D-Link Cam	Home Automation	Camera	Home Automation	Camera	Home Automation	Home Automation
2.	D-Link Day Cam	Home Automation	Home Automation	Home Automation	Home Automation	Home Automation	Home Automation
3.	D-Link Switch	Camera	Home Automation	Home Automation	Smart Hub	Appliances	Appliances
4.	Edimax cam1	Home Automation	Home Automation	Home Automation	Home Automation	Home Automation	Home Automation
5.	Edimax cam2	Home Automation	Home Automation	Home Automation	Home Automation	Home Automation	Home Automation
6.	Edimax plug1	Camera	Camera	Home Automation	Smart Hub	Appliances	Appliances
7.	Edimax plug2	Camera	Home Automation	Home Automation	Smart Hub	Appliances	Appliances
8.	Ednet cam1	Home Automation	Home Automation	Home Automation	Home Automation	Audio	Audio
9.	Ednet cam2	Home Automation	Home Automation	Home Automation	Home Automation	Audio	Audio
10.	Ednet gateway	Home Automation	Camera	Home Automation	Smart Hub	Appliances	Appliances
11.	HomeMaticPlug switch	Home Automation	Home Automation	Home Automation	Camera	Camera	Camera
12.	Hue switch	Camera	Home Automation	Home Automation	Smart Hub	Appliances	Appliances
13.	D-Link Home hub	Camera	Home Automation	Camera	Smart Hub	Appliances	Appliances
14.	TP-LinkPlugHS110	Camera	Camera	Home Automation	Appliances	Appliances	Appliances
15.	TP-LinkPlugHS100	Camera	Camera	Home Automation	Appliances	Appliances	Appliances
16.	Lightify	Home Automation	Home Automation	Home Automation	Camera	Appliances	Appliances
17.	Max gateway	Home Automation	Home Automation	Home Automation	Smart Hub	Appliances	Camera
18.	Smarter coffee	Home Automation	Home Automation	Home Automation	Appliances	Appliances	Appliances
19.	WeMoInsight Switch1	Camera	Camera	Camera	Smart Hub	Appliances	Appliances
20.	WeMoInsight Switch2	Camera	Camera	Home Automation	Smart Hub	Appliances	Appliances
21.	Wemo switch	Camera	Camera	Home Automation	Smart Hub	Appliances	Appliances
22.	Wemo switch2	Camera	Camera	Home Automation	Smart Hub	Appliances	Appliances

Comparatively, using the concept of transferability, the RF classifier outperformed the other ML techniques on the IMC 2019 Payload data set [32] while KNN outperformed the other ML techniques on the IoT sentinel [28].

C. Comparison

Table VI shows the results on 24 devices sampled from the IMC 2019 Payload data set [32] using RF for training the model on the CIC IoT data set 2022 [35]. Using the concept of transferability, the classifier was able to correctly predict all cameras but one (Blink security camera) with an average accuracy of 90.8%. Hence, out of eight cameras, seven of them were correctly identified with average high accuracy. In comparison to the existing works by Meidan et al. [26] who performed a case study on the concept of transportability where the classifier was trained on two security cameras of the same kind (SimpleHomeXCS71001) in Lab A, and testing was performed in Lab B with a completely different security camera (Withings WBP02WT9510). The accuracy of the classifier was recorded at 100% when the security cameras were transported. The same concept of transportability was employed on TV and this achieved an accuracy of 85%. With almost near-perfect accuracy, it goes to say the model generalized one type of camera from a manufacturer (and model) to another across a different location. The possible reason for the generalizability of the work by Meidan et al. [26] is 1) two devices were employed to test the concept of transportability proposed, 2) user datagram protocol (UDP) a predominant protocol in devices like cameras hence if a camera is from a different manufacturer with a different model or unique identifier, due to the underlying configuration of the device which translates to the packets generated, this will yield outstanding results in terms on accuracy. Undeniably, our work equally yielded perfect accuracy for two cameras and near-perfect accuracy of 99.99% and 99.87% for Yi camera and Wansview camera, respectively. Comparatively, Kolcun et al. [18], trained a model using data from one period while the trained model was evaluated on the data from different periods. The results of this study demonstrated that while these models' accuracy is outstanding when tested against data obtained at the same time as the training data, it decreases with time when tested against data gathered during a different time frame. The models' accuracy degraded on average by between 12%–21%, with an average of 17%. The accuracy achieved in our work aligns with the degradation of accuracy when training and test data are drawn from different distributions as discussed by Kolcun et al. [18]. In our experiment, out of the 12 correctly predicted devices 6 of them had an accuracy ranging from 54% to 74%. There is a huge drop in terms of accuracy as compared to that of the results in Table III when both the training and test were from the same distribution. Du et al. [10] proposed a lightweight device identification scheme based on traffic analysis. Using the extremely randomized trees (ET) algorithm, the model attains an accuracy of 95.8% in 1-min time intervals. The authors anticipate their model will attain an accuracy of 99.3% over longer time intervals like 3 min. Comparing our work with Wang et al. [39] and their work on an ML-assisted approach for IoT device identification. In their work, SVM shows the best accuracy of 78% on both training and testing data sets. RF achieved an accuracy of 74%, while DT and Logistic Regression Classifiers achieved similar accuracies of an average of 47% in device identification and are much lower than the SVM and the RF. Fan et al. [11] used a semi-supervised approach for device identification with about 5% of labeled data and obtained an average accuracy of 99.81%.

VIII. VULNERABILITY ASSESSMENT

Our work does not end after the IoT device type identification by the ML classifier. We further analyze the possible vulnerability of the predicted device type. The vulnerability assessment implemented in this work is in four steps.

1) Search all three databases (NVD, Vulners, and IBM X-Force) with the predicted device type by the classifier.

TABLE VI
EXPERIMENTAL RESULTS ON THE DEVICE TYPE PREDICTED BY RF CLASSIFIER WITH THE CORRECTLY PREDICTED DEVICES HIGHLIGHTED IN GREEN, INCORRECTLY PREDICTED DEVICES IN RED ACCURACY IN ( ) WHEN THE TRAINED CIC IOT DATA SET 2022 WAS TRANSFERRED TO THE U.S. LAB AND TESTED WITH THE IMC 2019 PAYLOAD DATA SET [32] USING THE INTERACTION EXPERIMENT CONDUCTED. THE PREDICTED DEVICE TYPE LABELS ARE C=CAMERA, HA=HOME AUTOMATION, SH=SMART HUB, AU=AUDIO, AND AP=APPLIANCE

Amazon Cloudcam (74.2%)		
Type	**count**	**IP**
C	2226	74.2%
HA	768	25.6%
SH	4	0.13%
Au	2	0.07%

Amazon Echodot (54.1%)		
Type	**count**	**IP**
Au	1433	54.12%
SH	888	33.53%
HA	168	6.34%
C	159	6.00%

Amazon Echospot (65.2%)		
Type	**count**	**IP**
Au	1956	65.20%
HA	504	16.80%
C	411	13.70%
SH	129	4.30%

Amazon Echoplus (60.2%)		
Type	**count**	**IP**
Au	1805	60.166667
C	669	22.300000
HA	437	14.566667
SH	89	2.966667

Amcrest Camera (100%)		
Type	**count**	**IP**
C	3000	100%

Behmor Brewer (0.12%)		
Type	**count**	**IP**
SH	512	62.21%
HA	260	31.59%
C	50	6.08%
Au	1	0.12%

Blink Camera (16.62%)		
Type	**count**	**IP**
HA	4626	56.98%
SH	1921	23.66%
C	1349	16.62%
Au	221	2.72%
AP	2	0.02%

Flux Bulb (100%)		
Type	**count**	**IP**
HA	5915	100.00%

Google home mini (0.2%)		
Type	**count**	**IP**
C	7466	98.70%
HA	59	0.78%
SH	24	0.32%
Au	15	0.20%

Insteon Hub (21.72%)		
Type	**count**	**IP**
HA	6202	52.214177
C	3061	25.770332
SH	2580	21.720828
Au	35	0.29466

Lefun Cam (98.59%)		
Type	**count**	**IP**
C	32128	98.59%
HA	455	1.40%
Au	3	0.01%
SH	1	0.003%

Luohe camera (100%)		
Type	**count**	**IP**
HA	5915	100.00%

Microseven camera (70.84%)		
Type	**count**	**IP**
C	9598	70.84%
HA	2416	17.83%
Au	1532	11.31%
SH	3	0.02%

Phillips Hue Hub (23.11%)		
Type	**count**	**IP**
Au	5000	46.70%
SH	2476	23.11%
C	1750	16.33%
HA	1490	13.90%

Samsung washer (0%)		
Type	**count**	**IP**
Au	9036	73.847663
C	2370	19.369075
HA	676	5.524681
SH	154	1.258581

Sengled hub (0%)		
Type	**count**	**IP**
HA	2831	99.964689
C	1	0.035311

Smartthings Hub (0.94%)		
Type	**count**	**IP**
Au	6099	72.72%
C	1926	22.96%
HA	283	3.37%
SH	79	0.94%

TP-Link Plug (4.41%)		
Type	**count**	**IP**
C	2569	72.12%
Au	763	21.42%
HA	157	4.41%
SH	73	2.05%

Wansview camera (99.87%)		
Type	**count**	**IP**
C	22140	99.87%
HA	28	0.13%
SH	1	0.00%

Wemo plug (59.77%)		
Type	**count**	**IP**
HA	10858	59.77%
Au	6981	38.43%
C	175	0.96%
SH	152	0.84%

Wink 2 Hub (0.31%)		
Type	**count**	**IP**
C	2843	43.73%
Au	2239	34.44%
HA	1399	21.52%
SH	20	0.31%

Xiaomi cleaner (0%)		
Type	**count**	**IP**
HA	353	92.89%
C	27	7.11%

Xiaomi ricecooker (0%)		
Type	**count**	**IP**
HA	192	54.55%
C	160	45.45%

Yi camera (99.9%)		
Type	**count**	**IP**
C	38714	99.99%
HA	3	0.00%

2) Harvest each individual keyword search into respective local databases.
3) Standardize results from the harvested search for Vulners, and IBM X-Force to have the same attributes as that of NVD.
4) Aggregate the standardized results per device type from all three databases.

A. *Vulnerability Database*

In this section, we describe the three publicly available databases used in this study.

1) *NVD:* NVD is the de facto standard vulnerability information source [34]. The database is managed and regulated by the National Institute of Standards and Technology (NIST). NIST provides an ongoing analysis of CVEs and assigns common vulnerability scoring system (CVSS) base metrics for each vulnerability, and will update a score if more information becomes available. Ongoing analysis and scoring help NVD users understand the potential severity of each issue and help users to prioritize vulnerability management activities. NIST works directly with vendors and researchers to improve the quality of the published data and to provide the public with accurate scoring data [4].
2) *Vulners:* Vulners [34] gathers data from a myriad of sources about cyber breaches, including ICS-CERT advisories by vendor, cybersecurity-focused posts, also a majority coming from vulnerability databases. The current service has about 131 sources from NVD, JVNDB, US-CERT, and ZDI. The platform is constantly updated with new sources and a revision of the current source.

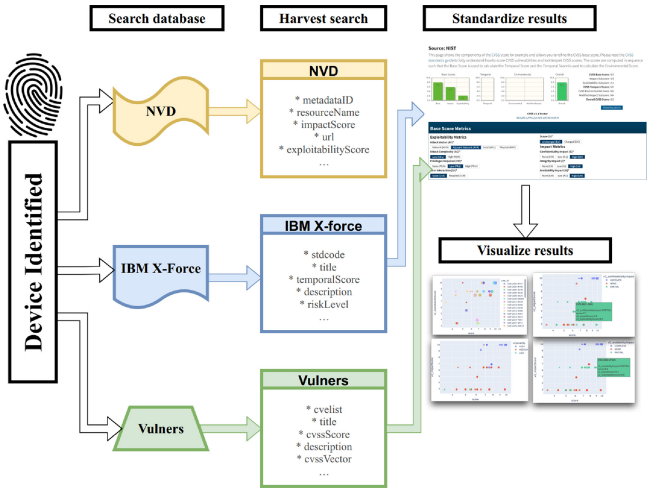

Fig. 5. Vulnerability assessment overview.

Hence, the data validity and integrity can differ as there is sometimes omitted information from the original entry. Vulners offers an API as a means to have the data locally available to manage and maintain. Vulners, in contrast to the other two sources, is a premium service with restricted free noncommercial use of an API and free use of the website's search engine. Vulners only permits 100 API calls per month; if this amount is exceeded, users are asked to upgrade their service.

3) *IBM X-Force:* The IBM X-Force database is gradually working its way to becoming one of the largest databases for vulnerability assessment and threat detection. Developers and researchers at IBM spent endless hours putting this together making IBM incorporate this data into their services and products. X-Force is as a result of information gathered from different sources like the Internet, research, and IBM Internet Security Systems software, among other related software. The IBM X-Force database has approximately 40000 distinct vulnerabilities. The limitation is that IBM X-Force allows 5000 records per month and a maximum of 200 records displayed per every API call.

B. Vulnerability Assessment Overview

Following successful device type identification by our ML classifier, the predicted device type is passed as a keyword to each of the three vulnerability databases stated in Section VIII-A. The following step entails gathering information from the three sources and storing it in their respective local database storage. The information is then standardized in the following step. The standardization works on Vulners and IBM X-Force search results. This works by taking the CVE-ID from each search result and passing it to the NVD database to retrieve the JSON response for that specific CVE. The NVD database is assumed to be the "master source," with most or all CVEs approved and verified. The steps are depicted in Fig. 5.

1) Vulnerability Keyword Search: The classifier predictions are passed to the API endpoints as keywords, and a JSON response is returned from the API based on the keyword predicted by the classifier.

In order to obtain vulnerabilities, an API call is used to access the IBM X-Force and vulnerability database. In contrast, the NVD data stream[6] is scraped for each individual year. To synchronize the local database and the NVD website's feed, a cron job is executed to update the data feed every time it is updated on the NVD website. The scraped feeds are in a zipped JSON format, hence a script is run every year to unzip them. Our NVD database is constructed by combining the unzipped data feeds for every year together. This data is searched for any potential vulnerabilities related to the predicted device type from the classifier.

2) Vulnerability Harvesting: Three vulnerability databases are searched for each of the five predicted device types (audio, appliance, camera, home automation, and smart hub). As a result, there will be 15 iterations of $3^{vd} * 5^{pd}$, where 3^{vd} stands for the three vulnerability databases and 5^{pd} for the five predicted device types. We identified a small number of IoT device vulnerabilities based on a preliminary examination of the NVD data feed. As a result, rather than the actual devices themselves, our vulnerability evaluation concentrated on the device types. Some modifications were made in order to generate vulnerability data for the relevant device kinds. The predicted device types and the keyword searched are *Audio*{Audio and Speaker}, *Camera*{Camera and Video}, *Home Automation*{Plugs, Bulb, and Lamp}, *Appliances*{Brewer, Kettle, Microwave, Vacuum, Washer, Dryer}, and *Smart Hub*{Hubs, Smarthub, and Smartthings}. This was to ensure there is a comprehensive list of vulnerabilities that encompasses the individual devices that belong to each device type category. In order to meet the requirements for storage, the obtained search results were normalized and synthesized before being saved in a PostgreSQL database.[7]

3) Vulnerability Standardization: The attributes and representation of the search results from the various databases varied. The gathered search results were made more uniform and reliable by standardizing them. The NVD is presumed to be a more reliable source for vulnerability assessment [3], yet the other sources discovered vulnerabilities that NVD had missed and vice versa. Using the NVD database, the data obtained from Vulners and IBM X-Force was standardized. Parsing the CVEID from the IBM X-force and Vulners database to the NVD database to retrieve the results produced is one of the data processing operations involved in this stage. The standardized search results are stored in a local database.

4) Vulnerability Aggregation: In this step, the standardized vulnerabilities that were received from each data source are combined into a single device type. The NVD data is synchronized with the gathered standardized Vulners and IBM X-Force data. For example in order to represent the Camera for further research, $\text{NVD}_{\text{camera}}$, $\text{Vulners}_{\text{camera}}$, and $\text{XForce}_{\text{camera}}$ are combined into one vulnerability data source. The other device types are subject to this aggregation.

[6]https://nvd.nist.gov/vuln/data-feeds
[7]https://www.postgresql.org/

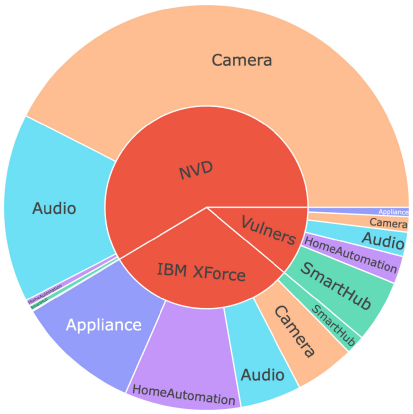

Fig. 6. Sunburst chart showing hierarchical results for the respective databases and the predicted device types.

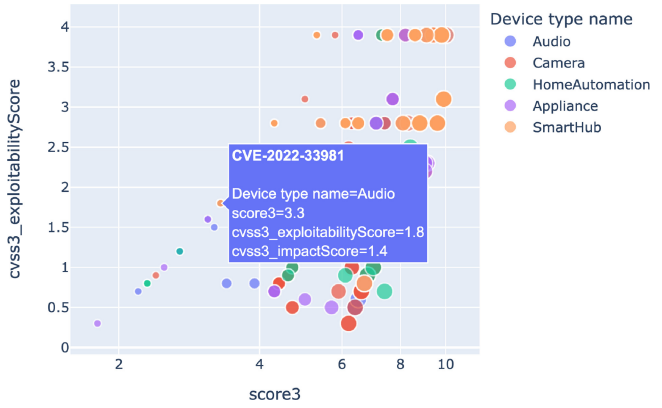

Fig. 7. Scatter chart of device types with score3 on the x-axis and exploitability score on the y-axis.

IX. Reporting and Dashboarding

The proposed framework includes a dashboard depicting the vulnerabilities found for each type of predicted device. To improve the presentation of the evaluation results, we created a Web-based visualization dashboard interface. Plotly,[8] a Python library, is the programming language used to integrate all graphs and display them on the portal. The respective device type data had diverse attributes that are compelling to plot. However, we picked the attributes that best articulate the recent vulnerabilities and the scope of this research.

Exploitability score reflects the simplicity and technical feasibility of exploiting the vulnerability.

Impact Score depicts the immediate result of an effective exploit.

score3 is the version 3 CVSS score value between the range of 1-10.

Charts displaying the visual representation of vulnerabilities represent the analysis of the searches. The test data used to perform the visualization is the IMC 2019 Payload data set [32]. When creating the graphs for this study, two possibilities were taken into account.

1) *Device Type Evaluation:* Visualizing the average vulnerability for each device type. Fig. 6 is a sunburst graph which is essentially a hierarchical multilevel chart that uses the different databases as the root and the predicted device types as the children. The value of the predicted device type for each database is computed as the average of the CSSV score3 for each device type. Hence, Camera in the NVD database has a larger portion than the other inferring Camera has a high-CSSV score3 average. Fig. 7 displays a scatter plot of the various CVEs, the type of device, the CSSV version 3 score, the exploitability score, and the impact score. Hovering over the scatter plot is an Audio device with the CVE_ID CVE-2022-33981, the CSSV version 3 score of 3.3, the exploitability score of 1.8, and the impact score of 1.4. Fig. 8 shows the bar chart which combines numerous data points per each device type per each database.

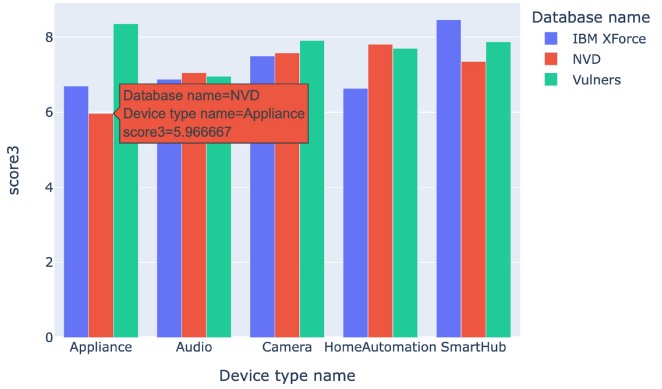

Fig. 8. Bar chart showing the average CSSV score3 of the different predicted device types in the three different databases.

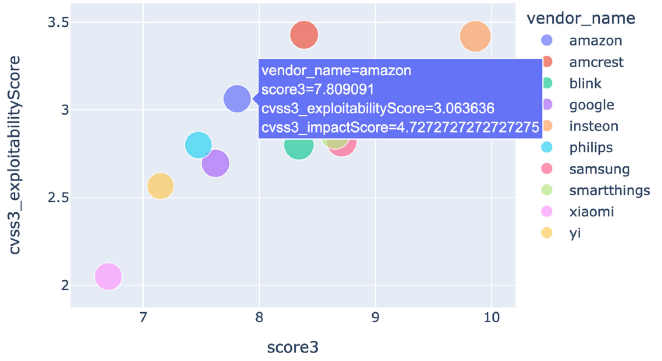

Fig. 9. Scatter plot of the different vendors with CSSV score3 on the x-axis and the CSSV exploitability score on the y-axis.

2) *Vendor Evaluation:* Displaying the vulnerability rating for each vendor per device. Fig. 9 depicts a scatter plot of the various vendors with CSSV score3 on the x-axis and the CSSV exploitability score on the y-axis. Amazon is listed as a vendor with an average CSSV score3 of 7.8, an average exploitability score of 3.1, and an average impact score of 4.7 in the current hover, Fig. 10 is a bar chart which computes the aggregate CSSV score3 score for each vendor. The current hover shows Insteon as a vendor with a CSSV score3 of 9.86. Using the CSSV score3 on the x-axis and the CSSV exploitability score on the y-axis.

[8] https://plotly.com/python/plotly-express/

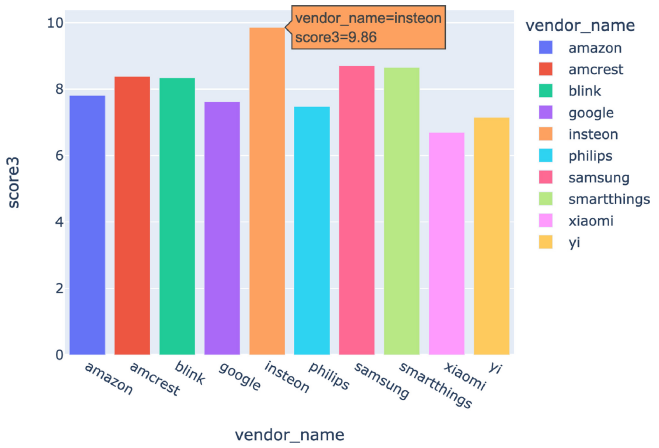

Fig. 10. Bar chart showing different vendor and their aggregated CSSV score3.

X. Conclusion and Future Works

This study proposed the use of a transferability approach based on the concept of transductive transfer learning for IoT device profiling and identification. To the best of our knowledge, this is the first work to use transductive transfer learning for IoT device identification. Forty eight features were extracted from each device's pcap files while selecting the best 20 features using a statistical feature selection technique called ANOVA. The 20 best features were used to train the different ML models. For verifiability, the two publicly available data sets were used to test our proposed approach. CIC IoT data set 2022 was used as our training data set (source domain) while our test data sets (target domain) are IMC 2019 Payload data set and IoT Sentinel data set. Experimenting with multiple ML models on the CIC IoT data set 2022 where the training and test data is drawn from the same distribution, the gradient boosting classifier achieved the best results in terms of accuracy, precision, recall, and F1-score. Experimenting with multiple ML models where the CIC IoT data set 2022 is the source and IMC 2019 Payload data set is the target domain MLP outperformed the other classifiers. Experimenting with multiple ML models where the CIC IoT data set 2022 is the source and the IoT Sentinel data set is the target domain Support Vector classifier outperformed the other classifiers. The predicted device type from the classifier was searched through three vulnerability databases (NVD, Vulners, and IBM X-Force) and the results were displayed on a visualization dashboard. Cameras in the NVD database have the highest CSSV score while the Insteon vendor has the highest vulnerability. From training and testing the data set with multiple classifiers when the training and test data is from a different distribution, the experiments indicate that some classifiers work best for a particular device type category. In future, this work will be extended to use a robust deep learning technique that can uniquely classify each device correctly. Additionally, this study will implement instance detection of multiple IoT devices in a network as well as individual device identification based on the concept of transductive transfer learning.

References

[1] A. M. Almomani and M. N. A. Rahman, "A literature review of the adoption of Internet of Things: Directions for future work," *Int. J. Contemp. Manage. Inf. Technol.*, vol. 2, no. 2, pp. 15–23, 2022.

[2] P. Anand, Y. Singh, A. Selwal, M. Alazab, S. Tanwar, and N. Kumar, "IoT vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges," *IEEE Access*, vol. 8, pp. 168825–168853, 2020.

[3] A. Anwar, A. Abusnaina, S. Chen, F. Li, and D. Mohaisen, "Cleaning the NVD: Comprehensive quality assessment, improvements, and analyses," 2020, *arXiv:2006.15074*.

[4] H. Booth, D. Rike, and G. Witte, *The National Vulnerability Database (NVD): Overview*, NIST, Gaithersburg, MD, USA, 2013.

[5] H. B. Braiek and F. Khomh, "On testing machine learning programs," *J. Syst. Softw.*, vol. 164, Jun. 2020, Art. no. 110542.

[6] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[7] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[8] *Internet of Things (IoT) and Non-IoT Active Device Connections Worldwide From 2010 to 2025 (in Billions) [Graph]*, Statista Res. Dept., Statista, Hamburg, Germany, 2020.

[9] *Internet of Things (IoT) Connected Devices Installed Base Worldwide From 2015 to 2025*, Statista Res. Dept., Statista, Hamburg, Germany, 2016.

[10] R. Du, J. Wang, and S. Li, "A lightweight flow feature-based IoT device identification scheme," *Security Commun. Netw.*, vol. 2022, Jan. 2022, Art. no. 8486080.

[11] L. Fan et al., "An IoT device identification method based on semi-supervised learning," in *Proc. 16th Int. Conf. Netw. Service Manage. (CNSM)*, 2020, pp. 1–7.

[12] D. A. Fitts, "Variable criteria sequential stopping rule: Validity and power with repeated measures anova, multiple correlation, manova and relation to chi-square distribution," *Behav. Res. Methods*, vol. 50, no. 5, pp. 1988–2003, Oct. 2018.

[13] J. Fürst, M. F. Argerich, B. Cheng, and E. Kovacs, "Towards knowledge infusion for robust and transferable machine learning in IoT," *Open J. Internet Things*, vol. 6, no. 1, pp. 24–34, 2020.

[14] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, no. 14, pp. 2627–2636, 1998.

[15] B. Gupta, A. Rawat, A. Jain, A. Arora, and N. Dhami, "Analysis of various decision tree algorithms for classification in data mining," *Int. J. Comput. Appl.*, vol. 163, no. 8, pp. 15–19, 2017.

[16] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[17] S. Kaplantzis, A. Shilton, N. Mani, and Y. Sekercioglu, "Detecting selective forwarding attacks in wireless sensor networks using support vector machines," in *Proc. 3rd Int. Conf. Intell. Sens., Sens. Netw. Inf.*, 2007, pp. 335–340.

[18] R. Kolcun et al., "Revisiting IoT device identification," 2021, *arXiv:2107.07818*.

[19] F. S. Koppelman and C. G. Wilmot, "Transferability analysis of disaggregate choice models," *Transp. Res. Rec.*, vol. 895, pp. 18–24, 1982.

[20] S. B. Kotsiantis, "Decision trees: A recent overview," *Artif. Intell. Rev.*, vol. 39, no. 4, pp. 261–283, 2013.

[21] J. Luo and J. Wang, "Vulnerability assessment of IoT devices through multi-layer keyword matching," in *Proc. Int. Conf. Comput., Internet Things Control Eng. (CITCE)*, 2021, pp. 138–143.

[22] A. E. Maas, F. Rottensteiner, and C. Heipke, "A label noise tolerant random forest for the classification of remote sensing data based on outdated maps for training," *Comput. Vis. Image Understand.*, vol. 188, Nov. 2019, Art. no. 102782.

[23] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A literature review," *J. Comput. Commun.*, vol. 3, no. 5, p. 164, 2015.

[24] V. Malik and S. Singh, "Security risk management in IoT environment," *J. Discr. Math. Sci. Cryptogr.*, vol. 22, no. 4, pp. 697–709, 2019.

[25] J. Manyika, *The Internet of Things: Mapping the Value Beyond the Hype*, McKinsey Global Inst., Atlanta, GA, USA, 2015.

[26] Y. Meidan et al., "Detection of unauthorized IoT devices using machine learning techniques," 2017, *arXiv:1709.04647*.

[27] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated device-type identification for security enforcement in IoT," 2016, *arXiv:1611.04880v2*.
[28] M. Miettinen et al., "IoT sentinel demo: Automated device-type identification for security enforcement in IoT," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2017, pp. 2511–2514.
[29] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, p. 21, Dec. 2013.
[30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[31] B. Quanz and J. Huan, "Large margin transductive transfer learning," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1327–1336.
[32] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information exposure from consumer IoT devices: A multidimensional, network-informed measurement approach," in *Proc. Internet Meas. Conf.*, 2019, pp. 267–279.
[33] K. Rose, S. Eldridge, and L. Chapin, "The Internet of Things: An overview," *Internet Soc. (ISOC)*, vol. 80, pp. 1–50, 2015.
[34] M. Rytel, A. Felkner, and M. Janiszewski, "Towards a safer Internet of Things—A survey of IoT vulnerability data sources," *Sensors*, vol. 20, no. 21, p. 5969, 2020.
[35] D. Sajjad, M. Hassan, K. D. Priscilla, Z. Alireza, A. T. Kevin, and A. G. Ali, "Towards the development of a realistic multidimensional IoT profiling dataset," in *Proc. 19th Annu. Int. Conf. Privacy, Security Trust (PST)*, Fredericton, NB, Canada, 2022, pp. 1–11.
[36] M. Sheikhan, M. Bejani, and D. Gharavian, "Modular neural-SVM scheme for speech emotion recognition using ANOVA feature selection method," *Neural Comput. Appl.*, vol. 23, no. 1, pp. 215–227, Jul. 2013.
[37] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICCS)*, 2019, pp. 1255–1260.
[38] S. Tyagi and S. Mittal, "Sampling approaches for imbalanced data classification problem in machine learning," in *Proc. ICRIC*, 2020, pp. 209–221.
[39] Y. Wang et al., "IoT device identification using supervised machine learning," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, 2022, pp. 1–6.
[40] N. Yousefnezhad, A. Malhi, and K. Främling, "Security in product lifecycle of IoT devices: A survey," *J. Netw. Comput. Appl.*, vol. 171, Dec. 2020, Art. no. 102779.
[41] F. Zhuang et al., "A comprehensive survey on transfer learning," 2019, *arXiv:1911.02685*.

Priscilla Kyei Danso is currently pursuing the M.Sc. degree in computer science with the University of New Brunswick, Fredericton, CA, Canada.

Her current research interests include the Internet of Things, data analysis, data mining, and machine learning.

Sajjad Dadkhah (Member, IEEE) received the Ph.D. degree from the Universiti Teknologi Malaysia, Skudai, Malaysia, in 2014.

His current research interests include Multimedia watermarking and security, cyber security, IoT Security, and security in machine learning techniques.

Euclides Carlos Pinto Neto (Member, IEEE) received the degree in computer science from the Federal Rural University of Pernambuco, Recife, Brazil, in 2016, and the M.Sc. and Ph.D. degrees in computer engineering from the Digital Systems Department, University of São Paulo, São Paulo, Brazil, in 2018 and 2021, respectively.

His research interest includes applications of Artificial Intelligence techniques on safety-critical systems.

Alireza Zohourian received the degree in B.Sc. degree in mathematics from Shiraz University, Shiraz, Iran, in 2016, and the M.Sc. degree in computer science from the University of Tehran, Tehran, Iran, in 2020.

His research interest includes the Internet of Things, IoT security, and IoT profiling.

Heather Molyneaux received the M.A. and Ph.D. degrees from the University of New Brunswick, Fredericton, CA, Canada, in 2002 and 2009, respectively.

She began her career with the National Research Council of Canada in the Human–Computer Interaction team and is currently employed with the Cybersecurity team. Her research interests include human-centred cybersecurity and usable security.

Rongxing Lu (Fellow, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2012.

He is currently the Mastercard IoT Research Chair, a University Research Scholar, and an Associate Professor with the Faculty of Computer Science, University of New Brunswick, Fredericton, CA, Canada. His research interests include applied cryptography, privacy-enhancing technologies, and IoT-big data security and privacy.

Ali A. Ghorbani (Senior Member, IEEE) received the M.Sc. degree in computer science from the George Washington University, Washington, DC, USA, in 1979, and the Ph.D. degree in computer science from the University of New Brunswick, Fredericton, CA, Canada, in 1995.

He is the co-inventor of three awarded patents in the areas of Network Security and Web Intelligence and has published over 260 peer-reviewed articles during his career. He is the Co-Founder of the Privacy, Security, Trust Network in Canada and its international annual conference.