

An Unsupervised Machine Learning Approach for IoT Device Categorization

Faiçal Sawadogo, John Violos, Aroosa Hameed, Aris Leivadeas

Department of Software and IT Engineering, École de technologie supérieure, Montreal, Canada

Email: faycalsawadogo@gmail.com, {ioannis.violos.1, aroosa.hameed.1}@ens.etsmtl.ca, aris.leivadeas@etsmtl.ca

Abstract—Internet of Things (IoT) along with the advances in the recently emerged Edge Computing environment, have allowed the introduction of new and very diverse applications that can facilitate our everyday life. However, one intrinsic characteristic of IoT is the heterogeneity of the IoT devices that are continuously connected and disconnected, creating a highly volatile communication environment. In addition to that, new types of IoT devices are constantly manufactured making a supervised categorization approach not applicable due to the lack of historical data. Nonetheless, the classification or type identification of the IoT devices is important for the management and the decision making of the IoT applications, and can be used for traffic characterization, density prediction, network planning and security reasons among others. Accordingly, in this paper we propose for the first time an unsupervised machine learning methodology for the IoT device categorization that leverages traffic characteristics obtained at the network level. To this end, we tackle the limitation of requiring an annotated dataset, while our model could also work efficiently with new and not previously detected IoT devices. To do so, we experimentally evaluate our approach using two clustering algorithms namely, the K-Means and the BIRCH in a real dataset. The experimental evaluation presents promising results that enhance the applicability of unsupervised approaches for the IoT device categorization problem.

Index Terms—Internet of Things, Device Categorization, Unsupervised Learning

I. INTRODUCTION

The Internet of Things (IoT) has seen tremendous growth in recent years and more IoT manufacturers are emerging offering a variety of new types of devices to improve the well-being and every-day activities of our society. These devices can range from ordinary household items to healthcare sophisticated tools with all having Internet access capabilities. The number of IoT devices is fast increasing with over 7 billion IoT devices connected today, while the experts expect that number to reach 22 billion by 2025 [1].

IoT devices continuously generate data while they have significant requirements for computational and storage resources making the network engineers to resort in Cloud and Edge computing solutions [2]. However, this end-to-end communication solution can be affected by the type of the IoT devices and their intermittent association with the application that is hosted in these remote computing platforms. Thus, it is of great importance to be able to quickly identify the type of device and associate it with the application counterpart.

This work was supported in part by the CHIST-ERA-2018-DRUID-NET project "Edge Computing Resource Allocation for Dynamic Networks"

Additionally, an accurate classification of the device can result in better identifying the traffic characteristics of each type of device, more efficiently allocating the resources of a dynamic and resource limited Edge Computing environment [3], and indicating malicious devices in the IoT network.

Thus, a new problem is generated, called IoT device classification, that tries to map each device to a specific category. This classification can be facilitated by leveraging historical data that contain labeled and pre-specified categories of IoT devices and by using supervised machine learning approaches [4]. However, new applications are constantly introduced and accompanied with new types of devices. Additionally, recent trends of virtualization on the IoT device level allow the devices to dynamically change their type according to the subset of sensors and functionalities they support at each time [5]. All these count towards in favor of resorting to less explored unsupervised learning classification techniques for the IoT device classification problem.

Hence, the research gap we focus to eliminate in this work, is the automatic categorization of the type of IoT devices without prior knowledge of the number and type of classes. Specifically, in our research we use a clustering algorithm in order to group the network traces we get by the IoT devices. The clusters of the IoT traces represent the different IoT types of the devices, whereas every device is characterized by the cluster in which its traces are assigned. In order to perform this clustering we used and adapted for the problem at hand two well known unsupervised algorithms, namely the K-Means and BIRCH algorithms. Furthermore, to estimate the number of the IoT device types the elbow method was utilized. If a new type of device is identified in the infrastructure, it can be estimated by an outlier detection technique or comparing the centroids locations over time. As we will discuss later a centroid is a feature vector that represents a cluster and the outliers are the instances that are dissimilar with the previous observations. To the best of our knowledge, this is the first research endeavor that deploys an unsupervised learning technique for the IoT device classification problem.

The rest of the paper is structured as follows: Section II presents a brief review of the related work. Section III explains the challenges. Section IV discusses the proposed model. Section V describes the experimental evaluation. Finally, Section VI concludes the paper with an overview of a future work.

II. RELATED WORK

In the pertinent literature, there are various studies for the device categorization such as fingerprinting based models, aggregated traffic models and supervised machine learning approaches. The IoT device fingerprinting models identify the type of IoT devices used within an IoT application by analyzing their network traffic i.e., packet-level information. Regarding the fingerprinting based approaches, Bezawada et al. [6] proposed both a static and a dynamic behavioral fingerprinting model. Specifically, the authors used a number of features from the packet header and payload such as entropy, length and window size. Similarly, Meidan et al. [7] proposed a device categorization of IoT devices using continuous classification. For this, firstly the set of authorized devices are defined called "white list", and then, their network traces are gathered. 300 network attributes are used from each TCP traffic session for the device classification using majority voting for every 20 consecutive sessions.

Miettinen et al. proposed a security based classification system called IoT sentinel in which a device type identification technique is provided that can recognize and identify the IoT devices immediately after they are connected to the network using a single attribute vector with 276 network features [8]. There are also solutions that applied aggregated traffic models to provide the device identification. Laner et al. [9] proposed a Coupled Markov Modulated Poisson Processes (CMMPP) framework to capture the traffic behavior of a single machine-type communication along with the collective behavior of tens of thousands of M2M devices. In [10], a classification strategy is designed for a fleet management use case incorporating three different classes of M2M traffic states, namely periodic update, event-driven, and payload exchange.

Regarding machine learning models, there is a great focus on applying supervised methods to solve the device categorization problem. For instance, the authors in [11] proposed a Network Traffic Classifier (NTC) based on a combination of deep learning models. The authors experimented with various combinations of a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) with varying architectural details such as different number of layers and different number of neurons in each layer. A representation learning technique was proposed in [12] to identify the different devices. The purpose of this was to identify the IoT devices in a network and the devices that do not belong in a predefined white list. Similarly, Meidan et al. [13] classified the connected devices in an organization's network based on an analysis of the network traffic. Firstly, the authors differentiate the IoT from non-IoT devices using network traffic and HTTP ownership. Secondly, the device presence is detected using its network behavior.

Sivanathan et al. [14] introduced an IoT device classification framework using statistical attributes, signaling patterns and cipher suites along with two stage machine learning models. At the first stage, the naive bayes algorithm is applied that provides the tentative classification and at second stage, random forest algorithm is applied that provides the final classification result. Hameed et al. [15] also proposed a two stage IoT device

multi classification framework that utilized both network and statistical features to characterize the IoT devices in the context of a smart city. The logistic regression was applied at stage 1 and a gradient boosting algorithm was provided at stage 2 for the final classification. This work was further extended in [4] by resorting to a deep learning technique, that leveraged a more extended feature set and by enhancing the classification accuracy through a feature correlation technique.

The above described works are limited in a way that it is required to have the labelled dataset as the approaches are based on supervised learning methods. Furthermore, the historical data of IoT devices are also required for the training of the models. Therefore, in this paper, we propose the use of a clustering and summarization machine learning model in order to address the limitations of categorizing IoT devices without previous data and without labelled data. For this purpose, we examine two different clustering techniques in order to provide an accurate and timely IoT device characterisation.

III. CHALLENGE AND APPROACH OF REAL TIME IOT DEVICE CATEGORIZATION

Our research goal is to design a model that identifies the type of the IoT devices connected to an Edge environment. This process takes place without historical data for training the model since it is not feasible or realistic to have a training dataset for any possible type of IoT device. The model takes as input the network traces of the devices by a network monitoring system and outputs for every connected device its category/type. In this unsupervised approach it is not provided the manufacturer or commercial name of the device type. Instead, we use the following four elements to represent an IoT device type: i) *Device Type ID*: a unique identification number; ii) *Summarization*: a summary of the device's network characteristics; iii) *Number of Devices*: the number of devices that have been categorized to belong in the Device Type ID; iv) *Cluster Centroid*: a feature vector that represents each device type in the N-dimensional feature space.

In most cases, IoT application providers and network planners are mostly interested in the summarization of each device type and the number of devices assigned to each type. This summarization includes information such as the protocol of communication of a device type, the average packet length, Time to Live (TTL), average Window Size (WS), the Source and Destination Ports, etc. Thus, with this manner, someone could extract how many devices are connected in the infrastructure, their type, and an estimation of the devices' behaviour in terms of the network characteristics.

Additionally, in this work, we also use the Cluster Centroids for the classification, which are estimated by the clustering algorithm. They are the mathematical representations, specifically feature vectors, formed in order to assign each device in a specific type. The assignments take place using a distance or similarity function between the device traces, which are also represented as feature vectors and Centroids. In the event that a new incoming trace has a long distance from the other

centroids, it can be assumed that a new type of device has been connected to the infrastructure. This event will trigger a new clustering and summarization in the observed traces in order to include the new incoming device type.

From the above description it is obvious that our analysis takes into consideration the time component. This analysis is appropriate to the particularities of IoT and Edge computing, since it manipulates their dynamic behaviour and captures the types of connected devices that may change over time. For someone that wants to benefit from such an unsupervised approach, it does not have to be aware of all the Machine Learning elements included, such as the Cluster Centroids, similarities, unsupervised algorithms, etc. In contrast, the interested parties will only have to follow the the output of our model, which is the number of connected devices, their types, a summarization for these types and last but not least the evolution of this information over time.

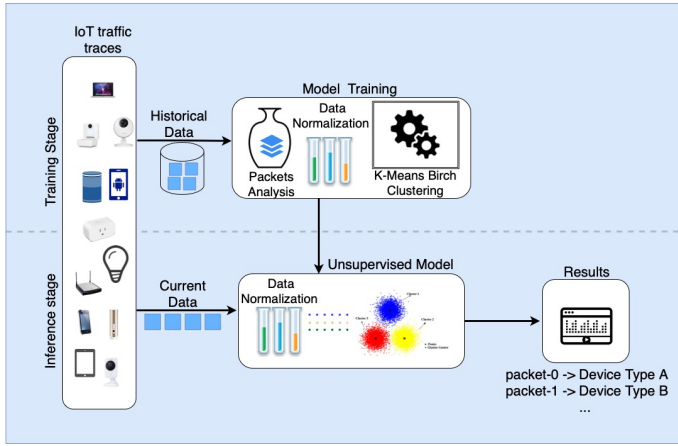


Fig. 1: Architecture of the solution

IV. PROPOSED METHODOLOGY AND UNSUPERVISED MODELS

In this section, we describe the main components and layers of the architecture of our solution. The architecture is depicted in Fig.1. On the left side we see that the devices connected to the Edge infrastructure produce traffic traces. These traces are available by a network monitoring tool such as the tcpdump. After we gather an initial number of packets, a preprocessing step takes place that maps the fields of the packets into feature vectors. The feature vectors are also subject to data cleaning and data normalization before being inputted to a clustering algorithm. The cluster algorithm provides a number of clusters, and their centroids. The number of clusters can be predefined or estimated in runtime by using an appropriate method, such as the elbow method. Each cluster is related with an IoT device type and using descriptive data analysis we can have a summarization of the device behaviour.

Generally speaking, the number of instances in order to extract the first set of clusters varies based on the algorithm and the use case. For the IoT device categorization, we have seen experimentally, it can span from 1,000 packets to 5,000

packets. After the first set of clusters has been calculated the new incoming packets are also assigned in a cluster using a distance metric between the cluster centroid and the packet feature vector. If the packet feature vectors have a distance from all the centroids bigger than the diameter of all clusters then a new type of device has been detected. In this case the clustering algorithm should recalculate the clusters and the new centroids including the ones produced from the new packets. In the next subsections, we give more details in the main components of this methodology.

A. Data Preprocessing

Data Preprocessing is the first step in the pipeline of the IoT device classification. Data preprocessing includes all the appropriate data transformations and filtering methods in order to map the information included in the network packets to feature vectors. Data Preprocessing in our case includes data verification, data selection, and data normalization. In the data verification we check for the accuracy and inconsistencies in the feature vectors. We also check and manipulate missing values, duplicate records and that the values are inside their acceptable range.

Data selection eliminates certain information in the data that are not necessary in the clustering process or they introduce bias such as the IP address, packet identification number, and the MAC address. Regarding the data transformation, we converted the no-numerical values into a numerical form as most of machine learning algorithms cannot work with nominal values.

Most of the clustering algorithms use distance measurements to determine whether an observation should belong to a certain cluster or not. Euclidean distance is used to measure these distances. So if a variable has significantly higher values, it can dominate the distance measures, suppressing other variables with small values. This can impact the efficiency of the clustering algorithm. To avoid this problem, we use the normalization technique of giving a large equal value or equal weight to all variables so that no variable steers the performance of the model in one direction simply because it contains a larger variable. Normalization is a rescaling of the data from the original range so that all values are within the new range of 0 and 1. A variable can be normalized by the Eq 1 where max is the maximum value of the feature and min is the minimum value.

$$y = \frac{x - \min}{\max - \min} \quad (1)$$

B. Clustering

Clustering is an unsupervised machine learning method that groups data objects based on information found only in the data that describes the objects and their relationships. The goal is for the objects of a group to be similar (or related) to each other and different (or not related) to the objects of other groups. In our work we group the network packets of the IoT devices and we categorize each device based on the group in which the majority of its packets are assigned to.

There are various clustering algorithms in the literature. However, for the particular problem, we should select those that satisfy the particularities of the IoT device categorization. Specifically most clustering algorithms divide the data objects into groups but cannot efficiently assign new objects later. In other words, these clustering algorithms cannot incrementally update their clusters, but instead the clusters should be calculated from scratch. Two algorithms that do not have this limitation and they are capable of predicting the cluster of new objects after the initial cluster formation are the K-Means and BIRCH.

1) *K-Means*: K-Means belongs to the Expectation-Maximization algorithms with two main iterative steps: (a.) the assignment of objects to their closest centroids and (b.) the recomputation of the cluster centroids. The algorithm begins with K initial centroids. For the selection of the number K , we use the elbow method which will be discussed in the next subsection. The centroids represent the clusters and are updated iteratively based on the objects assigned to their clusters. After a series of iterations and when the centroids do not change anymore, the K-means converges and the clustering is considered to be finished. The steps are also given in Algorithm 1.

Algorithm 1 K-Means Algorithm

Input: $D = \{d_1, d_2, \dots, d_n\}$ // Set of n data items.
 K // Number of desired clusters
Output:
A set of k clusters
Step:

1. Select K points as initial centroids.
2. Repeat:
3. Form k clusters by assigning each point to its closest centroid.
4. Recompute the centroid of each cluster.
5. Until centroids do not change.

2) *BIRCH*: BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) has the ability to incrementally and dynamically cluster incoming data objects. It is based on the Clustering Feature (CF) tree which is a height-balanced tree. CF is the triple (N, LS, SS) where N is the number of objects in the cluster, LS is the linear sum of the vector representation of the objects and SS is the square sum of these vectors. As each data object is encountered the CF tree is traversed and is assigned to the closest node at each level up to a leaf node. Leaf nodes have the restriction that must have a diameter that is less than a threshold T . If a new cluster has a diameter greater than T , the leaf node must be split. The steps of BIRCH are given in Algorithm. 2 [16]. For further details regarding the BIRCH and the clustering methodology followed, we refer the reader who is not familiar with data science to [17].

Algorithm 2 BIRCH Algorithm

Input:
 $D = \{d_1, d_2, \dots, d_n\}$ // Set of elements
 T // Threshold for CF tree construction
Output:
A set of k clusters
Step:

1. For each $d_i \in D$ do:
2. Determine the correct leaf node for d_i insertion.
3. if threshold condition is not violated then
4. add d_i to cluster and update CF triples.
5. else
6. if room to insert d_i then
7. insert d_i as single cluster and update CF triples.
8. else
9. split leaf node and redistribute CF features;

C. Number of IoT Device Types

The parameter K in K-means is the number of different types of IoT devices connected to the network. This number most of the times is not given in an unsupervised learning environment. To this end, appropriate heuristics should be used to estimate it. One well known heuristic is the Elbow method [17]. The elbow method runs the clustering algorithm for a range of values of K (say, K from 5-10) and then for each K value it computes an average score for all of the clusters. This score can be the sum of square distances from each point to the centroid of its cluster. To determine the right value of K , we select the value after which the sum of square distances start decreasing in a linear fashion.

D. Summarization of Groups

The clustering algorithm assigns the devices in clusters that represent categories of IoT devices. As described in Section III the categories include: the device type ID which is a unique identifier assigned to each cluster we found, the number of devices in this category, and also a summarization element. The summarization comes with a description, which is the output of a descriptive data analysis and includes several statistical metrics for the network characteristics of a device such as mean, median, standard deviation, max, min and the 25% and 75% percentiles. A summary of the statistical description of an IoT device, named Netatmo Welcome device, which is included in our dataset under consideration [18], is provided in Table II. The statistics are extracted for each of the device feature and their values vary based on the data distribution of each feature. These statistics are useful for applications such as task offloading, traffic analysis and intrusion detection.

This summarization description and depending on the type of features included (i.e. packet length, protocol used, etc.) can be of utmost importance for the service and network provider, since they can have more insights on the type of traffic generated by each device category. This way, they can better calculate the QoS expectations (i.e. throughput, delay,

TABLE I: Internal and External Evaluation performance

	Silhouette	Precision	Recall	F1-Score
K- Means	0.555	0.724	0.722	0.720
BIRCH	0.590	0.730	0.673	0.698

etc.) of the applications, more efficiently allocate spectrum and computing resources to each category of devices, and identify any network pattern that resembles to possible security threats (i.e. DDoS attacks, etc.).

E. New Type of IoT Devices Detection

IoT is characterized as a dynamic environment in which new types of IoT devices are connected and disconnected over time. In case that a new type of IoT device is connected, the model should be able to identify it and not to assign it in an existing cluster. In order to achieve this, we propose an outlier detection approach. In case that the new incoming data objects have a distance from every centroid, which is bigger than the diameters of the cluster, then a new type of device is assumed to have been detected and a recalculation of the clusters should take place including the new objects.

V. EXPERIMENTAL EVALUATION

For the evaluation of the proposed methodology we used the IoT Traces dataset collected by a group of researchers based at the School of Electrical Engineering and Telecommunication (University of New South Wales) [18]. The dataset is the collection of passive packet level of network traffic from 28 IoT devices over the course of 20 weeks. The types of devices are including cameras, lights, motion sensors, health devices and monitors. We used the Wireshark tool for the offline analysis of the pcap files. The pcap files include the packet data with the network characteristics and consists of more than 500,000 observations. The proposed methodology developed in the Python 3.8 programming language and we used the libraries NumPy and Pandas and Scikit-learn. The experimentation and implementation of our solution was performed in an HP Z420 PC with an Intel Xeon processor and 32 GB of memory running on a Windows 10 operating system.

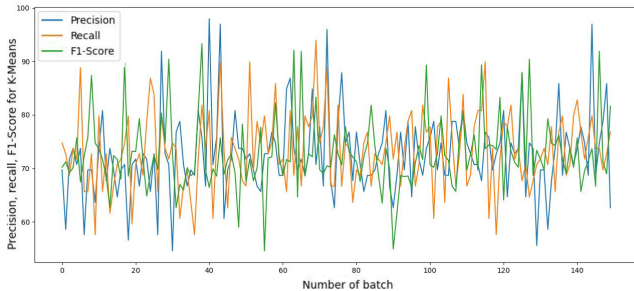


Fig. 2: Precision, Recall and F1-Score for K-Means

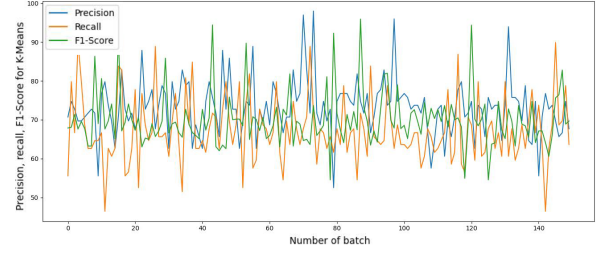


Fig. 3: Precision, Recall and F1-Score for BIRCH

A. Experimental Evaluation Protocol

We reorganized the dataset in time ordered mini-batches. Every mini-batch includes 100 packets and they come sequentially in the proposed model. In order to formulate the first set of clusters we used IoT traces from the first day of the experiment. Next, we provide sequentially and in time order the rest mini-batches from the following days and assign the corresponding IoT devices in clusters that represent the types of the IoT devices. For the performance evaluation of the clustering, we use the following internal and external evaluation metrics [19]. The performance in our experimental evaluation quantifies the ability of the clustering algorithms to group together in an unsupervised way devices that belong in the same category.

1) *Internal Evaluation Metrics*: Internal evaluation metrics do not use external knowledge and the ground truth categories of the devices. Instead, they rely on the dimensional distances of the clusters and the objects. Specifically they quantify the degree of whether there is a low distance within the inter-cluster objects, also named cohesion, or a high distance between intra-cluster objects, also named separation. Inter-cluster objects are the data observations assigned to the same cluster. While intra-cluster objects are the observations assigned to different clusters.

To express this distance, we have used the Silhouette coefficient, which expresses how close are the objects to their own cluster objects compared to the other clusters objects. Eq. 2 gives the Silhouette coefficient, where a is the average distance of the objects to the objects in their own cluster and b is the minimum average distances of the objects to the objects in other clusters. The coefficient ranges from -1 to $+1$.

$$s = \frac{b - a}{\max(a, b)} \quad (2)$$

2) *External Evaluation Metrics*: External evaluation methods introduce external information that targets the real labels of the objects and consequently the real categories. We use the external evaluation metrics, the Precision, Recall and F-measure with the terms True Positive (TP), False Positive (FP), and False Negative (FN). TP is the number of object pairs found together in the predicted cluster and in the ground truth category. FP is the number of object pairs found in the same cluster but in different categories in the ground truth. FN is

TABLE II: Summarization of Device type Netatmo Welcome

Device Features	Mean	Standard Deviation	Minimum	25%	Median (50%)	75%	Maximum
Communication Protocol	1.47749	0.63485	1.00000	1.00000	1.09013	2.00000	3.00000
Average Packet Length	110.05958	228.62041	42.00000	43.00000	66.00000	66.00000	1510.00000
Time to Live (TTL)	64.58941	50.20704	0.00000	47.00000	47.00000	52.00000	255.00000
Average Window Size	3243.35787	8318.26967	0.00000	836.00000	963.00000	2549.00000	65535.00000

the number of objects assigned in different clusters but they belong in the same category.

Precision expresses the percentage of the data objects that are relevant and properly grouped together (TP) out of the total objects the model grouped together (TP + FP). Recall evaluates the percentage of data objects correctly grouped together and identified as relevant (TP) out of the total objects that should be grouped together (TP + FN). F1-Score is the harmonic mean of the Precision and Recall and combines these two metrics in one. The values of all these metrics are calculated between [0,1] with 1 indicating the best and 0 the worst performance.

B. Experimental Outcomes

The IoT Traces dataset provides the real categories of the IoT devices. So, we apply both the external and internal evaluation metrics. We evaluate the performance for each sequential mini-batch and the time plots of K-Means and BIRCH are depicted in Fig. 2 and Fig. 3 respectively. The average performance is around 70%, with a minimum of around 50% and a maximum of almost 100%. We have 60 to 80% performance almost all the time. These results show that K-Means and BIRCH offer a good performance for the real time categorization of IoT devices. An additional interesting result is that packets from the same device or similar type of devices are more likely to be grouped together.

In Table I, we provide the aggregated internal and external evaluation metrics of our experiments. We see a Silhouette coefficient of 0.590 for the BIRCH and 0.555 for the K-Means. These coefficients declare that the clusters are well-separated and cohesive. Comparing K-Means and BIRCH we see that BIRCH has better results and formulates more modular clusters. The Table II also provides the summarization of a detected device type.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a methodology to categorize IoT devices based on the classical clustering algorithms K-Means and BIRCH. Both algorithms give an accuracy performance of about 70%. We aim to explore further clustering algorithms with more data transformation and processing methods in order to increase the performance. As future work we will also examine the multiple applications of our proposed model in different types of network problems such as the workload modeling and the task offloading.

REFERENCES

- [1] T. Mustafa and A. Varol, "Review of the Internet of Things for Healthcare Monitoring," in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, Jun. 2020, pp. 1–6.
- [2] F. Saeik, M. Avgeris, D. Spatharakis, N. Santi, D. Dechouniotis, J. Violos, A. Leivadeas, N. Athanasopoulos, N. Mitton, and S. Papavassiliou, "Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions," *Computer Networks*, vol. 195, p. 108177, Aug. 2021.
- [3] D. Dechouniotis, N. Athanasopoulos, A. Leivadeas, N. Mitton, R. Jungers, and S. Papavassiliou, "Edge computing resource allocation for dynamic networks: The druid-net vision and perspective," *Sensors*, vol. 20, no. 8, 2020.
- [4] A. Hameed, J. Violos, and A. Leivadeas, "A deep learning approach for iot traffic multi-classification in a smart-city scenario," *IEEE Access*, vol. 10, pp. 21 193–21 210, 2022.
- [5] I. AlShiab, A. Leivadeas, and M. Ibnkahla, "Virtual sensing networks and dynamic rpl-based routing for iot sensing services," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.
- [6] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral fingerprinting of iot devices," in *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*, ser. ASHES '18. ACM, 2018, p. 41–50.
- [7] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized iot devices using machine learning techniques," Tech. Rep. arXiv:1709.04647, 2017.
- [8] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2177–2184.
- [9] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," in *ISWCS*, 2013.
- [10] M. Laner, N. Nikaein, P. Svoboda, M. Popovic, D. Dragic, and S. Krco, "Traffic models for machine-to-machine (m2m) communications: types and applications," in *Machine-to-machine (M2M) Communications*. Woodhead Publishing, 2015, pp. 133–154.
- [11] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," *IEEE Access*, vol. 5, pp. 18 042–18 050, 2017.
- [12] J. Kotak and Y. Elovici, "IoT device identification using deep learning," in *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*. Springer International Publishing, aug 2020, pp. 76–86. [Online]. Available: https://doi.org/10.1007/2F978-3-030-57805-3_8
- [13] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "Profiliot: A machine learning approach for iot device identification based on network traffic analysis," in *Proceedings of the Symposium on Applied Computing*, ser. SAC '17. New York, NY, USA: ACM, 2017, p. 506–509.
- [14] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2019.
- [15] A. Hameed and A. Leivadeas, "Iot traffic multi-classification using network and statistical features in a smart environment," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020, pp. 1–7.
- [16] S. Zhu, S. Yang, X. Gou, Y. Xu, T. Zhang, and Y. Wan, "Survey of Testing Methods and Testbed Development Concerning Internet of Things," *Wireless Personal Communications*, vol. 123, no. 1, pp. 165–194, Mar. 2022. [Online]. Available: <https://doi.org/10.1007/s11277-021-09124-5>
- [17] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Pearson, 2016.
- [18] University of new souths wales, "IoT Traffic Traces". Accessed: 2022-05-01. [Online]. Available: <https://iotanalytics.unsw.edu.au/iottraces>
- [19] J.-O. Palacio-Niño and F. Berzal, "Evaluation Metrics for Unsupervised Learning Algorithms," Tech. Rep. arXiv:1905.05667, 2019.