 doi:10.20944/preprints202101.0273.v1

Device Classification-based Context Management for Ubiquitous Computing using Machine Learning

Nalini A. Mhetre

Department of Computer Engineering, Sinhgad College of Engineering, Pune, India

Email: nalini.mhetre@gmail.com

Arvind V. Deshpande

Department of Computer Engineering, SKN College of Engineering, Pune, India

Email: principal.skncoe@sinhgad.edu

Parikshit Narendra Mahalle

Department of Computer Engineering, SKN College of Engineering, Pune, India

Email: aalborg.pnm@gmail.com

Abstract

Ubiquitous computing comprises scenarios where networks, devices within the network, and software components change frequently. Market demand and cost-effectiveness are forcing device manufacturers to introduce new-age devices. Also, the Internet of Things (IoT) is transitioning rapidly from the IoT to the Internet of Everything (IoE). Due to this enormous scale, effective management of these devices becomes vital to support trustworthy and high-quality applications. One of the key challenges of IoT device management is automatic device classification with the logically semantic type and using that as a parameter for device context management. This would enable smart security solutions. In this paper, a device classification approach is proposed for the context management of ubiquitous devices based on unsupervised machine learning. To classify unknown devices and to label them logically, a proactive device classification model is framed using a k-Means clustering algorithm. To group devices, it uses the information of network parameters such as Received Signal Strength Indicator (rssi), packet_size, number_of_nodes in the network, throughput, etc. Experimental analysis suggests that the well-formedness of clusters can be used to derive cluster labels as a logically semantic device type which would be a context for resource management and authorization of resources. This paper fulfills an identified need of proactive device classification for device management.

Keywords context management, device classification, IoT device management, k-Means clustering, ubiquitous computing, unsupervised machine learning

1. Introduction

The term ubiquitous means “always existing everywhere” i.e., constantly available. In computer science, ubiquitous computing ('ubicomp') is a term where computing is accessible anytime and everywhere. Ubicomp can take place using any computing device, at any place, and in any format as opposed to desktop computing. Several ubicomp ideas have emerged in recent years which is nowadays termed as Internet of Things (IoT), a collection of smart, inter-connected objects using cutting edge internet technologies. Currently, the IoT is moving rapidly through various transformations and transitioning towards the Internet of Everything (IoE) [1]. A user interacts with the computer, which can exist in many different heterogeneous forms including laptops, computers, tablets, and everyday objects such as air conditioners, washing machines, stoves, fridges, TV, cars, even a pair of glasses, and several such physical objects (either smart or with an additional smart layer through various sensors). This rapid growth in devices and the availability of computing power to them leads to or demands context-aware and responsive application environments. Smart connectivity and context-aware computations are an important part of IoT [2] for various services.

The growth and spread estimate for the internet-connected devices is around 34 Billion [3] by the year 2020, which includes 20 Billion IoT devices. Reports also predict that IoT Device count will reach 76 Billion by the year 2025 [4], out of which 50 Billion will be connected to the internet. This explosion of devices [5] makes it crucial for systems to manage these devices to provide security and access control, which is an open issue [6]. One of the major challenges IoT faces is access control to its resources. Framing device-specific authorization policies would be difficult, time-consuming, and unscalable given a large number of heterogeneous IoT devices. Also, from a network security point of view, administrators may restrict the usage of certain types of devices. To address these problems, future solutions would require dealing with a group of devices rather than dealing with a single device. Thus, there is a need to classify devices having certain similarities. It would also be useful to automate the process of device grouping.

All major IoT devices or things in ubicomp are typically battery-operated or with limited energy. Also, the signal strength of the device has a significant impact on energy consumption and the communication it intends to do. Various computations, accessing resources, and more specifically the transmission of data consumes a lot of energy, and it is the main reason why energy consumption is one of the main constraints to consider when building ubicomp systems. Thus, to classify devices by taking energy and signal strength as primary parameters along with other networking parameters to study would be more useful from the context management perspectives. In recent times, the automatic classification of smart devices using different contexts including network packets [7], device ids [8] has been explored. However, the majority of the work focuses on identifying devices based on their consistent features but not considering the changing state of the device during the lifetime of communication and resource utilization.

In this paper, a novel approach to classify devices using their networking features by applying unsupervised machine learning techniques is proposed. This model helps to find common patterns of network parameters in devices to group them under a particular class. A technique shows the usefulness of k-Means clustering algorithms in automating the process of device classification. using prominent network features such as the number_of_nodes, signal strength, energy consumption, throughput along with secondary features such as constant bit rate (cbrrate), packet size, overhead, delay and hop count for device grouping. We demonstrate the usefulness of our model to derive logically semantic context that could be used for resource management.

The rest of this paper is organized as follows. Section 2 focuses on motivation; Section 3 presents related work. The proposed approach and technical details are discussed in Section 4. The experimental results are given in section 5. Finally, the paper is concluded in section 6.

2. Motivation

We are living in an era of anything, anytime, anywhere paradigm. In the last two decades, advancements in technologies brought down hardware costs drastically and allowed device manufacturers to add communication technologies in tiny devices like heartbeat monitors to large appliances like TV, fridge, washing machine. More devices are being designed with WiFi capabilities, and many heterogeneous devices are interconnected or connected in the sequel to allow resource sharing and communication with other network participating nodes. Technology coverage has shifted from traditional desktops to smart things [9]. Smart things observe, gather, and transmit data to offer personalized services. The contribution of the ubicomp system is critical to the deployment of personalized services in smart hyperspaces [9]. The IoT is about adding capabilities to these objects to connect them with the internet. An example of ubicomp is a smartwatch that alerts the user about a phone call and allows that call processing through that watch. Traditional methods of access control and security solutions may not be applicable as it is to this scenario. Historically, security requirements are considered to be relatively static because access control decisions do not change with context, nor do they account for changing environmental conditions. but, smart connectivity and context-aware computations are an important part of IoT [2] for various services to make them adaptive.

As we discussed previously, context plays a vital role in today's networks. Context-aware services respond and adapt to changes in their computing environment by designing policy rules [10]. As ubicomp comprises heterogeneous devices, context related to these devices [11] and their capabilities is important for context-aware applications . This context may affect the behavior of these applications as they not only use user interactions and their internal state information but also context information sensed during execution. The core issue of this problem is how to allocate or authorize the resources in the IoT system to accommodate the requirements imposed by applications. Due to the large scale and heterogeneity, long term context management of these devices becomes crucial. also, Resource-aware computing [12] is an approach to implement systems where the system continuously monitors the consumption of essential resources and can help the application make a decision based on resource availability now and in the future. These applications need to track existing resources, their capabilities [13], and their availability. For example, video streaming can be adjusted to the available bandwidth or signal and battery level, or the device may be asked to go to an area with better wireless local area network coverage [14].

The scope provided in the aforementioned points makes us look at device classification from the perspective of device context management. This essentially would be created through linking, monitoring, and analyzing some key parameters such as – signal strength, data packet size, energy consumption, transmission delay due to nodal hops, and the device and network-related features. Various factors affect the state of the network and device state. For Device-to-Device communication, in the context of IoT, there have been several approaches studied and validated. The Validity of such approaches and their respective feasibilities have also been tested successfully. These several approaches revolve around either Supervised or unsupervised environments or rule-based standard algorithms. After studying all these approaches, their respective strengths, and feasibility, we decided to take a novel approach of unsupervised machine learning algorithms based on clustering techniques to investigate unlabeled information of devices.

3. Related work

3.1 Device classification

Classification of each device connected to a network and participating in communication is a tedious task, and there needs an approach to classify devices based on device capabilities. Significant efforts were made in the past by researchers to classify devices based on various device parameters or device contexts. Bharat [15] discussed classification based on features extracted from network traffic. A. Sivanathan [16] worked on a multistage machine learning-based classification framework that uniquely identifies IoT devices with high accuracy. Mahalle [17] discussed Decision theory-based Object Classification, and the paper has presented the logical framework for object classification to provide contextual information by considering energy parameters. A. Sivanathan [18] presented a technique to classify IoT traffic based on network and device parameters like Sleep time, Active volume Avg. Packet size, Mean Rate, Peak / Mean rate Active time, No. of servers, No. of protocols, Unique DNS req., DNS interval NTP. Given the resulting traffic profile, the probability histogram of the sleep time attribute is studied and observed that there is a unique pattern for some IoT devices. This approach was to classify IoT and Non-IoT devices.

Y. Meidan [19] presented a machine learning approach for IoT device identification based on network traffic analysis. This work used supervised learning and trained a multistage classifier that can distinguish traffic generated by IoT and Non-IoT devices. S. Sharma [20] presented a generalized approach of incremental clustering to classify heterogeneous devices in a dynamic ubiquitous computing environment. The HiCHO technique is protocol neutral and based on attributes, and it's too general for dynamic dimensions. Kalmar [21] used the Hierarchical temporal memory (HTM) framework for context identification of objects facilitating context-aware services. Through different sets, this work proved that if the training data set was ideal and consistent; then the network could classify previously unseen context vectors related to objects quite efficiently.

S. V. Radhakrishnan [22] and A. J. Pinheiro [23] used network traffic analysis to classify devices with a non-machine learning approach. Earlier, M. Danieletto [24], addressed device classification based on ontology parameters while X. Feng [25] proposed a rule-based engine and P. R. J. Pego and L. Nunes [26] proposed approach-based custom communication properties. to classify devices. Ke Gao [27] focused on the classification of only AP's using wavelet analysis of network packets. The overall perspective of the various researchers to classify devices is application specific.

Table I lists a summary of the related work for device classification based on various machine learning techniques.

Table I. Summary of related work

Existing Work	**Device classification parameters**	**ML**	**ML Type**	**Algorithm**	**Issues**
Big data analytics for classification of network-enabled devices. [28]	Source IP address, destination IP address, source and destination port numbers, direction of the flow of traffic,	Yes	Supervised	KNN, NB, SVM, RF	Limited to Medical IOT devices

	protocol used, number of packets transmitted, duration for which the connection was made, and total data received in bytes				
A Comparative Study of Classification Techniques for Managing IoT Devices of Common Specifications. [29]	Software and hardware specification	Yes	Supervised	KNN, NB, SVM, RF	Physical Attributes only
A machine learning approach for IoT device identification based on network traffic analysis. [19]	Logical characteristics of the network traffic	Yes	Supervised	Multistage Classifier	Two Parameters IOT and Non-IoT Device
Characterizing and classifying IoT traffic in smart cities and campuses. [18]	Sleep time, active volume, average packet size, mean rate, peak to mean ratio, active time, number of servers, number of protocols, unique DNS requests, DNS interval, NTP interval, most frequent port number, and a label identifying the IoT device	Yes	Unsupervised	k-Means	Classification based on device types.
Identifying IoT devices and events based on packet length from encrypted traffic[30]	Device Events	Yes	Supervised	k-Nearest Neighbors (k-NN), Decision Tree, Random Forest, Support Vector Machine	Hypotheses testing based
Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. [16]	Packet-level and flow-level	Yes	Supervised	Multistage Classifier	Signal Strength Not considered
IoT devises Recognition Through Network Traffic Analysis [31]	Size of the first N packets sent and received, arrival times	Yes	Supervised	Random Forest classifier	Limited Dataset

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 January 2021 doi:10.20944/preprints202101.0273.v1

Automatic Device Classification from Network Traffic Streams of Internet of Things [7]	time when the packet is sent out or received, packet length, protocol, MAC address of source device and destination device	Yes	Deep Learning	CNN	Training

3.2 Gap analysis

Analysis of related work shows that in the early days, the Machine Learning approach was not used in the area of device classification. However, recent work employs machine learning techniques in this domain, particularly supervised. Very few attempts have been made to use unsupervised machine learning. Most of the device classification models focus on packet data, while some consider very limited parameters for classification, like, energy or software/hardware specifications, vendor information, packet size, etc. It is also observed that the data set used was limited and specific to limited types of devices. The use of signal strength of devices along with other networking parameters while classifying them is unaddressed. This is unstructured information which makes unsupervised learning a suitable solution to identify patterns to form logical groups of devices.

4. Proposed work

In ubiquitous computing environments, smart applications need methods for classifying devices based on certain criteria such as physical, logical, and networking attributes. This classification may help in applying rules and policies, providing secure access control for a group of devices, and in device management. In this section, the proposed approach and methodology to classify devices with semantic labels are discussed. Later, an algorithm to categories IoT devices using unsupervised machine learning is presented. An automatic technique to classify IoT devices into different groups based on their network event information could be useful in many applications.

4.1 Proposed approach and methodology

Figure 1 shows the main modules of the proposed approach, which includes Data acquisition, Data Preprocessing and Feature Selection, Device Clustering, and Cluster Labelling.

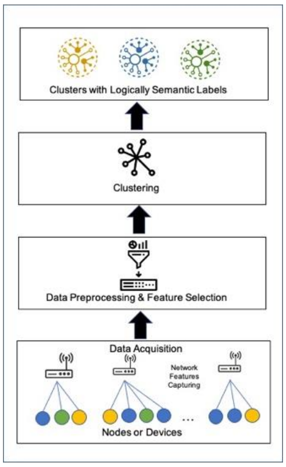

Figure 1. Proposed approach and methodology

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 January 2021 doi:10.20944/preprints202101.0273.v1

Data acquisition

When IoT devices are connected to wireless networks, the gateway monitors network event logs of these devices which contain useful information about devices such as number_of_nodes, rssi, packet_size, overhead, throughput, delay, energy_consumed, nodes in the network, etc. We propose to use this dataset to do clustering of the devices. This dataset could be represented as a collection of device event records as:

$$D = \{D1, D2, \ldots, Di, \ldots, Dn\} \qquad – (1)$$

Where Di represents network event information of various features of the i^{th} device. Each record Di contains features information that corresponds to a specific device and is given in eq. 2:

$$Di = \{n_i , rssi_i, throughput_i, energy_i, \ldots, others_i \} \qquad – (2)$$

Where n_i represents the number_of_nodes in the network when an i^{th} device is connected, $rssi_i$ represents the Received Signal Strength Indicator of the i^{th} device, $throughput_i$ is the throughput of i^{th} device, $energy_i$ represent energy_consumed of i^{th} device, and so on. All other features' information that is not considered in this work is represented as $others_i$.

Data Preprocessing and Feature Selection

In the data preprocessing module, values for the above-mentioned parameters are analyzed and normalized on a scale of 0 – 1. Normalization is done in a dataset to bring the values of these parameters to a common scale, without changing differences in their ranges of values. Further correlation between features is identified to evaluate their importance in cluster formation. Dataset D is extracted for features of interest by filtering out othersi.

Device Clustering

We aim to categorize IOT devices using selected features and classify them accordingly. For this, we need to identify certain unseen patterns amongst selected features. These patterns are to be used to form clusters of devices. Here, we propose a device classification model based on unsupervised machine learning and by using the k-Means clustering algorithm. The k-Means clustering algorithm is a simple and widely used unsupervised machine learning technique to make inferences from datasets without referring to known or labeled outcomes. The dataset of selected features is represented by eq. 1 which is an input to the k-Means clustering algorithm. Here, k refers to a finite number of clusters to be formed and the elbow method is used to obtain the optimal value of k. The output clusters refer to a group of devices that are formed due to certain similar patterns in selected features. Let, C be the set of k output clusters denoted as:

$$C = \{C1, C2, \ldots, Ck\} \qquad - (3)$$

Suppose k=2, then C = {C1, C2} i.e. 2 clusters C1 & C2 of devices will be formed and every device in dataset D is allocated to one of these 2 clusters as follows:

$$C1 = \{d \mid d \in D \text{ and } d \notin C2\} \text{ i.e. } C1 \subset D$$

$$C2 = \{d \mid d \in D \text{ and } d \notin C1\} \text{ i.e. } C2 \subset D$$

$$C1 \cap C2 = \emptyset \qquad - (4)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 January 2021 doi:10.20944/preprints202101.0273.v1

$$C1 \cup C2 = D \quad - (5)$$

If the optimal value of the number of clusters is *k*, then clusters C1, C2, ..., Ck will be formed. Accordingly, eq. 3 and eq. 4 can be extended as:

$$(C1 \cap C2 \cap \ldots \cap Ck) = \emptyset \quad - (6)$$

$$(C1 \cup C2 \cup \ldots \cup Ck) = D \quad - (7)$$

Cluster Labeling

Each cluster formed is further analyzed by considering values of selected features from the dataset to derive semantic labels for it. These labels given to clusters represent the logical type of the device indicating a specific category. This categorization characterizes the classification of IoT devices as an outcome of our approach. These labels can be considered as context information of the device belonging to that category which may serve for policy enforcement while granting resources to them.

4.2 Algorithm

The algorithm *AssignClusters()* to classify devices based on their selected features using the clustering technique is summarized in pseudo-code as depicted in Algorithm 1.

Algorithm 1: AssignClusters() // Identify clusters and assign devices to clusters

Input: D[]
Output: C[] (C1, C2, …, Ck k number of clusters of devices)
Method:

1. N[] ⟵ normalize(D[])
2. F[] ⟵ corr(N[])
3. Initialize itr ⟵ 15
4. k ⟵ FindOptimalK(N[], itr)
5. C[] ⟵ k-Means(k, N[])
6. R[] ⟵ ReduceDimentionsT-SNE(D[], C.labels)
7. Get a cluster label from R for each record in D.

Input to *AssignClusters()* algorithm is a dataset D. D is a collection of vectors of features characterizing devices. The first step of algorithm 1 normalizes the dataset D between 0 and 1. The normalized dataset is denoted as N which contains values of features on the common scale of 0 to 1. In the next step of the algorithm, the correlation between normalized features is calculated to find the features of interest. Features with maximum correlation are selected for analysis. This dataset is used as a training dataset for the clustering model. The next step is to find out the optimal number of clusters for a given training dataset. *FindOptimalK()* algorithm finds optimal value for k by plotting distortion for every iteration of *itr*. It is described in Algorithm 2. Further, *k-Means()* clustering algorithm generates k number of clusters using this optimal value of k. C represents a list of k clusters where each record in D is assigned to one of the k clusters. As D is multi-dimensional data, to visualize the clusters formed, the t-SNE (t-Distributed Stochastic Neighbor Embedding) method is used to map the output cluster labels. It is a dimensionality reduction algorithm that plots high-dimensional data to two or more dimensions suitable for visualization. t-SNE would be a better option because it can reduce dimensions with non-linear relationships. R denotes k clusters of data points with reduced dimensions. Cluster id for each record representing a specific IoT device is obtained from R.

Data of the records belonging to a specific cluster can further be analyzed to infer the semantic meaning of that cluster. As per the semantic meaning derived, clusters can be labeled logically which could be used as context for device management.

Algorithm 2: FindOptimalK () // Find Optimal K

Input: N[], itr
Output: k (optimal count of clusters)
Method:

1. **Initialize** k←1
2. **For** k = 1 to itr **do**
    a. Model ← k-Means(*k*, *N*[])
    b. Calculate distortion of the model for k
    c. k ← k + 1
3. **End for**
4. Return k, where the cost of the distortion drops significantly.

Algorithm 2 finds the optimal number of clusters for a given dataset. It assumes some finite value of *itr*. It iterates in a loop for values of *k* from 1 to *itr* to generate the k-Means Model and to find distortion of the same. This algorithm returns the value of k where the cost of the distortion drops significantly.

5. Results and discussion

To evaluate our approach, we collected network event data of wireless nodes through simulation. Network simulator NS2 is used to perform the simulation. In this section, we first provide an overview of the collected dataset and further discuss an experimental evaluation of the proposed methodology. Later, we also analyze the impact of different features in cluster formation.

5.1 Dataset

Network event information of more than 200 wireless nodes was collected through several simulations of various wireless communication scenarios. The simulation setup of each scenario considered 802.11 mac layer protocol representing the IoT environment. Other input parameters to the simulation include the number_of_nodes, packet_size, signal strength indicator (ssi), cbrrate Rate with varying values. The number of nodes ranging from 10 to 115 were deployed in the area of 500 * 500 meters. The initial energy of each node was set to 100 Jules. From the trace file generated through simulation, we extracted node-specific features such as total packet sent, total packets received, total packet dropped, total packet forwarded, packet delivery ratio, total hop count, overhead, throughput, delay, energy_consumed, residual energy, etc. Values of input parameters and extracted features merged to form a dataset.

5.2 Experimental evaluation

Initially, a dataset of more than 200 devices comprising values of 21 attributes is generated through simulation trace file logs. This work considers a dataset of 9 features which includes the number_of_nodes in the network, ssi, packet_size, constant bit rate (cbrrate), average hop count, overhead, throughput, delay, energy_consumed. After cleaning and normalizing this data set, the correlation between these features is calculated. Table II shows the correlation matrix of these features.

 doi:10.20944/preprints202101.0273.v1

Table II. Correlation of features of interest

Features	number_of_nodes	ssi	packet_size	cbrrate	avg_hop_count	Overhead	throughput	delay	energy_consumed
number_of_nodes	1.000	-0.145	-0.333	-0.342	0.032	0.532	-0.129	-0.063	-0.252
ssi	-0.145	1.000	0.074	0.184	-0.213	0.089	0.822	0.079	0.898
packet_size	-0.333	0.074	1.000	0.480	-0.070	-0.069	0.261	0.417	0.257
CBR	-0.342	0.184	0.480	1.000	0.059	-0.122	0.264	0.387	0.324
avg_hop_count	0.032	-0.213	-0.070	0.059	1.000	-0.097	-0.210	0.020	-0.198
Overhead	0.532	0.089	-0.069	-0.122	-0.097	1.000	-0.019	-0.028	0.080
Throughput	-0.129	0.822	0.261	0.264	-0.210	-0.019	1.000	0.068	0.833
Delay	-0.063	0.079	0.417	0.387	0.020	-0.028	0.068	1.000	0.223
Energy Consumed	-0.252	0.898	0.257	0.324	-0.198	0.080	0.833	0.223	1.000

Each cell in the table indicates the correlation coefficient value (in the range of -1 to 1) between two variables. These variables represent dataset features and are shown in the respective row and column of the cell. The extreme correlation coefficient value -1 is a negative correlation and 1 is a positive correlation i.e., a perfectly linear relationship. Diagonal values show that each variable always perfectly correlates with itself and can be ignored as we want a correlation between different pairs of variables. Values closer to 1 and -1 indicate a strong nonlinear correlation. From Table II, it is observed that ssi, throughput, and energy_consumed are strongly correlated with each other and are shown in darker cells. These features are potential candidates to form clusters. The line chart depicting the correlation between these features is shown in Figure 2 and Figure 3. However, we also investigated the impact of other features in cluster formation.

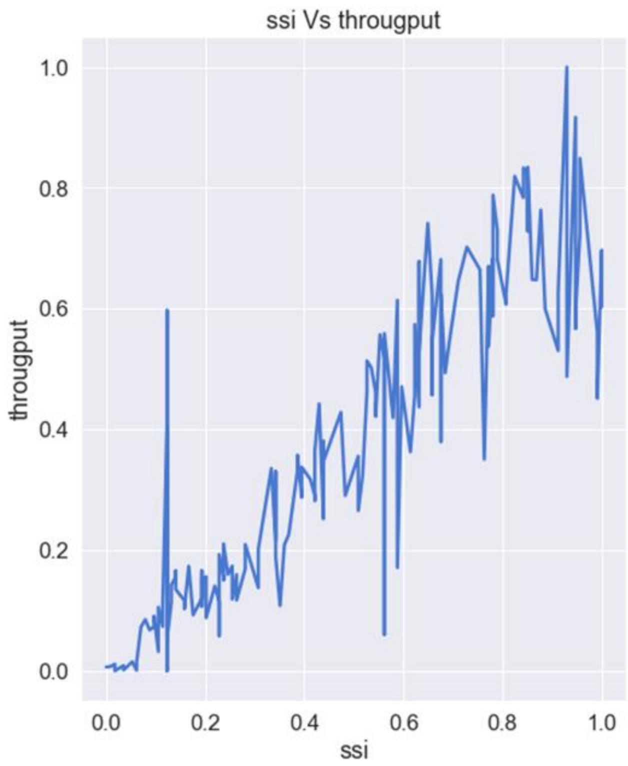

Figure 2. Correlation ssi Vs throughput

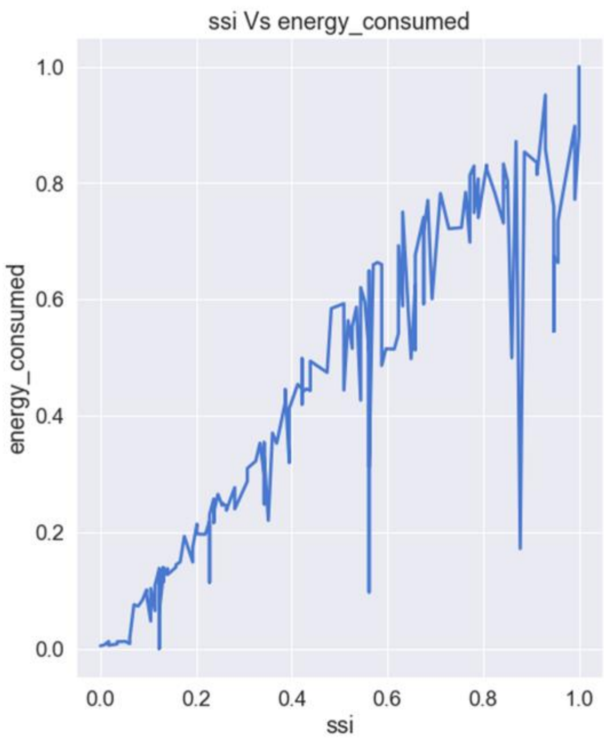

Figure 3. Correlation ssi Vs Energy Consumed

We used the elbow technique [32] to get an appropriate value of k which represents the optimal number of clusters. This included plotting the calculated distortions (average of the squared distances from centers of the respective clusters) to the number of clusters produced by different values of k. The value of k at which improvement in distortion declines the most is called the elbow point(curve). At this point dividing the data into further clusters should stop.

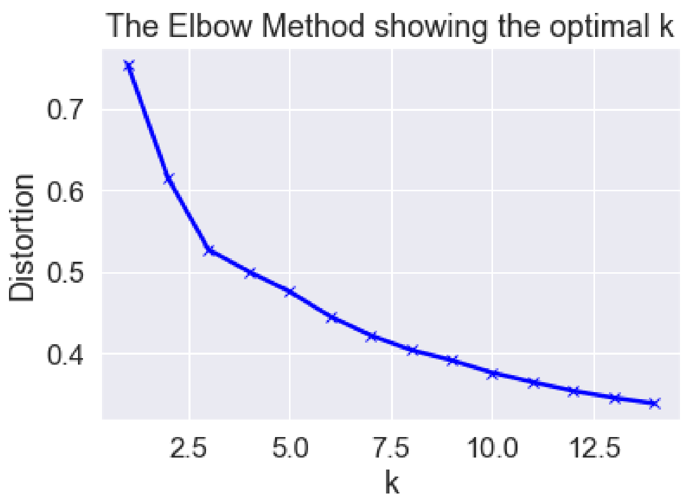

Figure 4. Optimal k using Elbow method

From Figure 4, a curve is observed at a value of k=3. As a result, we conclude that the optimal number of clusters is 3 for the considered dataset. At this cluster number, it can be seen from the Figure 4 the decrease in distortion starts to level off.

5.3 Cluster visualization

Figure 5 depicts the output of the k-Means clustering algorithm implemented using the python toolkit. It shows three clusters generated and data points assigned to these clusters. As the dataset is multidimensional, the t-SNE algorithm is used to visualize clusters in two dimensions. Data records of 221 devices are distributed as 71 in cluster '0', 65 in cluster '1', and 85 in cluster '2'.

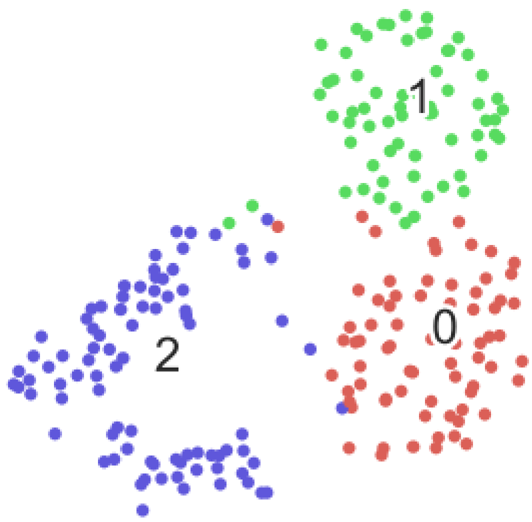

Figure 5. Cluster visualization

5.4 Cluster analysis

Internal evaluation of resultant clusters based on data itself suggests that clusters are well-formed. Figure 6 shows multivariate analysis in the form of a matrix of scatter plots depicting the impact of selected features in cluster formation along with resultant clusters. It displays a distribution of values associated with specific features in resultant clusters. This involves checking out distributions as well as potential relationships, patterns, and correlations amongst these features. Each data point is color-coded by the cluster to which it was assigned on the scale of normalized value. This helps in identifying which features give separation in the clusters and to observe each variable separately.

It can be observed that values of ssi, throughput, and energy_consumed has a somewhat similar distribution in all three clusters. Values of these features in cluster 1 are high, in cluster 0 low, and cluster 2 are medium. Nodes with smaller values of packet_size, delay, and cbrrate has a strong impact in cluster 2. Cluster 0 and cluster 1 span the average number_of_nodes in the network with moderate delay. Extreme values of the number_of_nodes can be seen in cluster 2 with low delay. Overhead is uniformly distributed across all the clusters with higher values in cluster 2. Cluster 0 and 1 represents nodes with average high and average medium cbrrate. Higher to medium-sized packet nodes are distributed in cluster 0 and cluster 1. The summary of the observations is given in Table III.

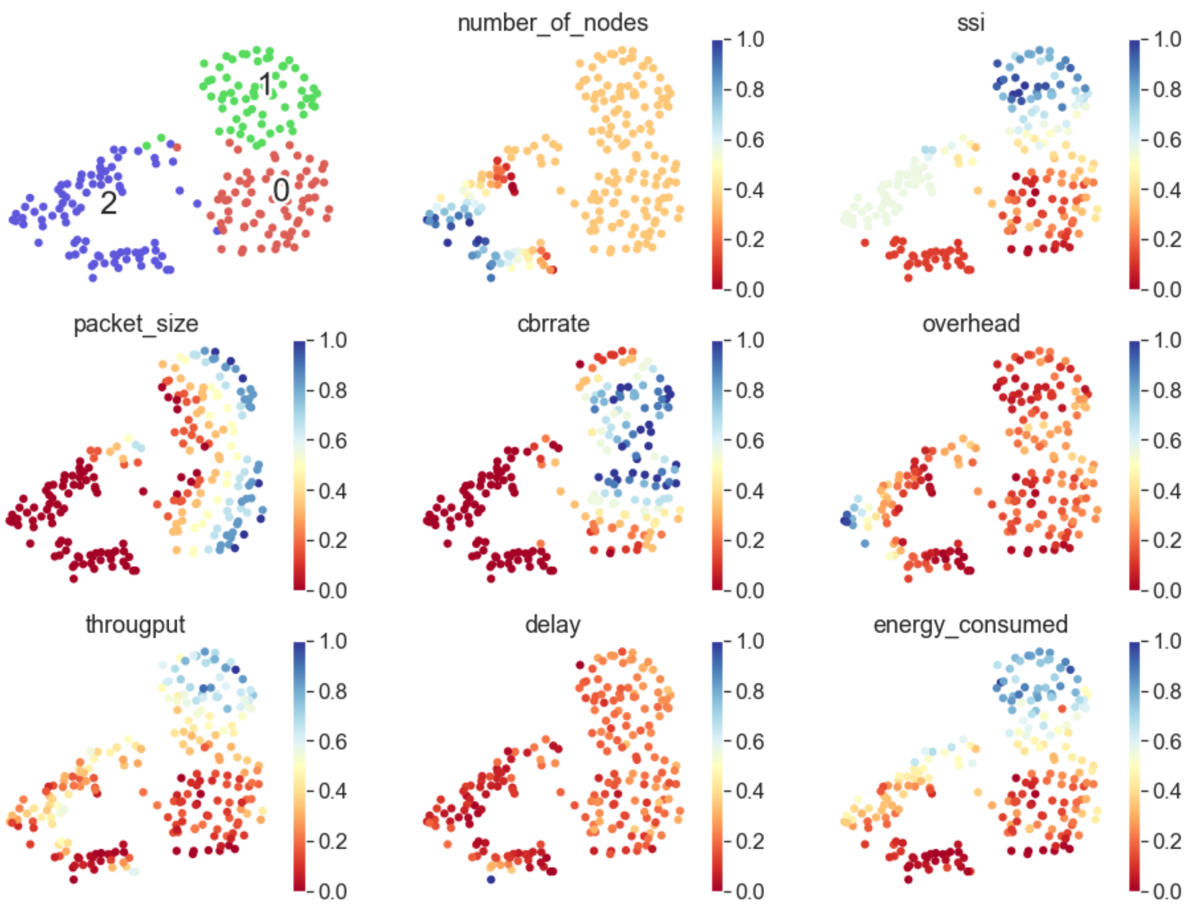

Figure 6. Scatter plots for features distribution in clusters

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 January 2021 doi:10.20944/preprints202101.0273.v1

Table III. Fuzzy analysis of node distribution in clusters

	number_of_nodes	ssi	packet_size	cbrrate	overhead	throughput	delay	energy_consumed
Cluster 0	M	L	M, H	M	M	L	M	L
Cluster 1	M	H	M, H	H	M	H	M	H
Cluster 2	L, H	M	L	L	M	M	L	M

L: Low, M: Medium, H: High

This paper uses a fuzzy[33] approach for analyzing clusters. In an uncertain environment like IoT, resource owners cannot have crisp values of the assets. Therefore, fuzzy is more appropriate. Table 3 shows a Fuzzy analysis of node distribution in the clusters formed. Cluster inference by considering impact features is given below in the form of Mamdani fuzzy rules[34]:

Rule 1:
IF ssi is L **AND** throughput is L **AND** energy_consumed is L **AND** cbrrate is M **AND** delay is M **THEN** Cluster_label is CONSTRAINED

Rule 2:
IF ssi is H **AND** throughput is H **AND** energy_consumed is H **AND** cbrrate is H **AND** delay is M **THEN** Cluster_label is POWERFUL

Rule 3:
IF ssi is M **AND** throughput is M **AND** energy_consumed is M **AND** cbrrate is L **AND** delay is L **AND** packet_size is L **THEN** Cluster_label is SEMI-POWERFUL

Thus, Cluster 0 is labeled as CONSTRAINED, Cluster 1 is labeled as POWERFUL, and Cluster 2 is labeled as SEMI-POWERFUL. SEMI-POWERFUL cluster is characterized by smaller packet size and cbrrate.

6. Conclusion and future work

In this paper, we propose a model to use networking information of IoT devices to classify them proactively using unsupervised machine learning techniques. The k-Means clustering algorithm is used to identify groups of IoT devices using simulation data in wireless environments. Our work effectively shows the possibility to automatically classify IoT devices and derive semantic context using networking parameters. As these parameters can be readily available within an organization, apparently our model can be used to facilitate a more intelligent IoT network. Results show that the use of clustering unsupervised machine learning model provides a viable solution in automating device classification. It will help organizations to design adaptive policies framework based on contextual information extracted from network

 doi:10.20944/preprints202101.0273.v1

parameters. In the future, we plan to explore the utilization of our device classification method in the area of adaptive access control and resource management. This model can further be extended for a dataset of new devices.

References

[1] D. Sudharshan, "Internet of Everything (IoE)," in *Marketing in Customer Technology Environments*, Emerald Publishing Limited, 2020, pp. 161–208.

[2] "IoT Ecosystem - Internet of Things Forecasts & Business Opportunities - Business Insider." https://www.businessinsider.com/iot-ecosystem-internet-of-things-forecasts-and-business-opportunities-2016-4-28?IR=T (accessed Jan. 14, 2020).

[3] "Managing the Explosion of Internet of Things (IoT) Data." https://www.nelsonhilliard.com/explosion-of-internet-of-things-iot-data/ (accessed Jan. 15, 2020).

[4] "Internet of Things - number of connected devices worldwide 2015-2025." https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[5] S. Ornes, "The internet of things and the explosion of interconnectivity," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 113, no. 40, pp. 11059–11060, 2016, doi: 10.1073/pnas.1613921113.

[6] Www.fortinet.com, "Understanding the Iot Explosion and Its impact on enterprise security." [Online]. Available: https://www.fortinet.com/content/dam/fortinet/assets/white-papers/WP-Understanding-The-IoT-Explosion-And-Its-Impact-On-Enterprise-Security.pdf.

[7] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Z. Yang, "Automatic Device Classification from Network Traffic Streams of Internet of Things," *Proc. - Conf. Local Comput. Networks, LCN*, vol. 2018-Octob, pp. 597–605, 2019, doi: 10.1109/LCN.2018.8638232.

[8] A. WG03, "AIOTI ALLIANCE FOR INTERNET OF THINGS INNOVATION Identifiers in Internet of Things (IoT)," 2018.

[9] J. Ma, L. T. Yang, B. O. Apduhan, R. Huang, L. Barolli, and M. Takizawa, "Towards a smart world and ubiquitous intelligence: A walkthrough from smart things to smart hyperspaces and UbicKids," *Int. J. Pervasive Comput. Commun.*, vol. 1, no. 1, pp. 53–68, 2005, doi: 10.1108/17427370580000113.

[10] M. A. El Khaddar, M. Chraibi, H. Harroud, M. Boulmalf, M. Elkoutbi, and A. Maach, "A policy-based middleware for context-aware pervasive computing," *Int. J. Pervasive Comput. Commun.*, vol. 11, no. 1, pp. 43–68, Apr. 2015, doi: 10.1108/IJPCC-07-2014-0039.

[11] R. van Eijk, A. Salden, J. de Heer, A. Peddemors, P. Määttä, and V. Haataja, "Handling heterogeneity in context aware services," *Int. J. Pervasive Comput. Commun.*, vol. 1, no. 1, pp. 25–30, Feb. 2005, doi: 10.1108/17427370580000110.

[12] K. John, "Ubiquitous Computing Fundamentals," *Ergonomics*, vol. 53, no. 5, pp. 724–725, May 2010, doi: 10.1080/00140131003769050.

[13] B. Anggorojati, P. N. Mahalle, N. R. Prasad, and R. Prasad, "Capability-based access control delegation model on the federated IoT network," *Int. Symp. Wirel. Pers. Multimed. Commun. WPMC*, pp. 604–608, 2012.

[14] D. Garlan, D. P. Siewiorek, A. Smailagic, and P. Steenkiste, "Project Aura: toward distraction-

free pervasive computing," *IEEE Pervasive Comput.*, vol. 1, no. 2, pp. 22–31, Apr. 2002, doi: 10.1109/MPRV.2002.1012334.

[15] B. A. Desai, D. M. Divakaran, I. Nevat, G. W. Peter, and M. Gurusamy, "A feature-ranking framework for IoT device classification," in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, Jan. 2019, pp. 64–71, doi: 10.1109/COMSNETS.2019.8711210.

[16] A. Sivanathan *et al.*, "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," *IEEE Trans. Mob. Comput.*, vol. 18, no. 8, pp. 1745–1759, 2019, doi: 10.1109/TMC.2018.2866249.

[17] P. N. Mahalle, N. Rashmi Prasad, and R. Prasad, "Object Classification based Context Management for Identity Management in Internet of Things," *Int. J. Comput. Appl.*, vol. 63, no. 12, pp. 1–6, Feb. 2013, doi: 10.5120/10515-5486.

[18] A. Sivanathan *et al.*, "Characterizing and classifying IoT traffic in smart cities and campuses," *2017 IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPS 2017*, pp. 559–564, 2017, doi: 10.1109/INFCOMW.2017.8116438.

[19] Y. Meidan *et al.*, "ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis," 2017, doi: 10.1145/3019612.3019878.

[20] S. Sharma, S. Kapoor, B. R. Srinivasan, and M. S. Narula, "HiCHO : Attributes based Classification of Ubiquitous Devices," no. i, pp. 113–125, 2012.

[21] A. Kalmar and R. Vida, "Towards context-aware mobile services through the use of Hierarchical Temporal Memory," in *2013 21st International Conference on Software, Telecommunications and Computer Networks - (SoftCOM 2013)*, Sep. 2013, pp. 1–5, doi: 10.1109/SoftCOM.2013.6671905.

[22] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, "GTID: A Technique for Physical Device and Device Type Fingerprinting," *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 5, pp. 519–532, Sep. 2015, doi: 10.1109/TDSC.2014.2369033.

[23] A. Aksoy and M. H. Gunes, "Automated IoT Device Identification using Network Traffic," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–7, doi: 10.1109/ICC.2019.8761559.

[24] M. Danieletto, N. Bui, and M. Zorzi, "An ontology-based framework for autonomic classification in the Internet of things," *IEEE Int. Conf. Commun.*, no. July, 2011, doi: 10.1109/iccw.2011.5963599.

[25] X. Feng, Q. Li, H. Wang, and L. Sun, "Acquisitional Rule-based Engine for Discovering Internet-of-Things Devices," *Usenix'18*, 2018.

[26] P. R. J. Pego and L. Nunes, "Automatic discovery and classifications of IoT devices," *Iber. Conf. Inf. Syst. Technol. Cist.*, 2017, doi: 10.23919/CISTI.2017.7975691.

[27] Ke Gao, C. Corbett, and R. Beyah, "A passive approach to wireless device fingerprinting," in *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, Jun. 2010, pp. 383–392, doi: 10.1109/DSN.2010.5544294.

[28] D. Arora, K. F. Li, and A. Loffler, "Big data analytics for classification of network enabled devices," *Proc. - IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2016*, pp. 708–713,

2016, doi: 10.1109/WAINA.2016.131.

[29] A. Mavrogiorgou, A. Kiourtis, and D. Kyriazis, “A Comparative Study of Classification Techniques for Managing IoT Devices of Common Specifications,” vol. 3, no. January 2019, 2017, pp. 67–77.

[30] A. J. Pinheiro, J. de M. Bezerra, C. A. P. Burgardt, and D. R. Campelo, “Identifying IoT devices and events based on packet length from encrypted traffic,” *Comput. Commun.*, 2019, doi: 10.1016/j.comcom.2019.05.012.

[31] M. Shahid *et al.*, “IoT Devices Recognition Through Network Traffic Analysis,” *2018 IEEE Int. Conf. Big Data (Big Data)*, no. 978-1-5386-5035–6, 2019, doi: 10.1109/BigData.2018.8622243.

[32] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, “Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 336, no. 1, p. 012017, Apr. 2018, doi: 10.1088/1757-899X/336/1/012017.

[33] L. A. Zadeh, “Fuzzy sets,” *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: 10.1016/S0019-9958(65)90241-X.

[34] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *Int. J. Man. Mach. Stud.*, vol. 7, no. 1, pp. 1–13, 1975, doi: 10.1016/S0020-7373(75)80002-2.