

WI-FI DEVICE IDENTIFICATION IN CROWD COUNTING USING MACHINE LEARNING METHODS

Jean Pierre Jarrier Conti^{1,2}, Tiago Buatim Nion da Silveira^{1,2}, Heitor Silvério Lopes¹

¹ Bioinformatics and Computational Intelligence Laboratory (LABIC)

Federal University of Technology - Paraná (UTFPR)

² Department of Technology – Telefonica Brazil

jean.conti@telefonica.com, hslopes@utfpr.edu.br, tiago.silveira@telefonica.com

Abstract – The increase in the availability of computational resources gave rise to new technologies to estimate the amount of people in a given area. In this context, algorithm-based solutions for crowd counting can be grouped into image-based and non-image based approaches, the latter considering any other feature that is not visual. Currently, due to the popularization of smartphones and mobile devices, several researchers have been using Wi-Fi request packets for crowd counting estimation. Assuming that, on average, each person in a given place carries a Wi-Fi device, the number of unique MAC addresses can be associated with the number of people. However, since the probe may capture all Wi-Fi traffic – which may include broadcast messages from other access points or packets from notebooks and desktop devices – some strategy must be applied in order to identify only personal mobile devices, thus improving the method accuracy. In this work, we trained classifiers to segment mobile from static devices through its Wi-Fi behavior pattern. Therefore, using data collected from different devices and in different environments, we evaluated the proposed methodology by using several machine learning algorithms. Best results were achieved with logistic regression and neural network (MLP). The results of this study suggest the feasibility of the proposed method for crowd counting in high-density Wi-Fi zones.

Keywords – Crowd counting, Machine learning, IEEE 802.11, Wi-Fi, Probe requests.

1. INTRODUCTION

Estimating the size of crowds has become a relevant topic in recent years [1–3], mostly stimulated by the increasing availability of computational power and new emerging technologies. Motivation for crowd counting goes beyond the informational purpose: for a recurring event, such as festivals or parades, it can be useful for a proper services sizing, products consumption forecasting, and adequacy of the security strategy. The same approach could be helpful for estimating the boarding time delay in airports or understanding people behavior while shopping at malls. In summary, estimating the distribution of people in a given place becomes an effective tool for social and business intelligence studies.

Traditional methods for crowd counting are mainly based on the Jacobs's method [4] which takes into account the occupancy percentage of a square footage based on a previous known crowd density distribution. More recently, applications based on computer vision and deep learning architectures have emerged for the same purpose [2]. However, it should be noticed that the first approach is time-demanding and low accurate – leading to biased results and misinterpretations [5] – and, on the other hand, the latter requires high operational and computational costs, not always available to the municipal public administration nor to the small organizations.

In the last years, due to popularization of the use of smartphones, some approaches relating the number of devices in a particular area with the number of people in there have aroused as an accurate and low-cost alternative for crowd counting, as demonstrated in [3, 6]. The main point of such approaches, therefore, is the premise that each person has with him a personal device with Wi-Fi connectivity.

Based on this, the hardware identification (MAC address) provided by the vendor can be used as a user identification (ID) [7, 8], which is then anonymized and used just for counting purpose. This process is performed capturing some packet management data, known as *probe requests*, through access points (AP) distributed in the area of interest [6]. This strategy becomes simpler when applied to outdoor areas. Although previous works have also shown satisfactory results indoor, in such situations the counting task is affected by the number of static devices in the same area (e.g. notebooks; computer desktops; and smart TVs). Since many manufacturers of mobile devices for personal use are also the same for static devices, just checking for the MAC address manufacturer is not enough for an accurate device type identification.

For the hitherto exposed, in this work we seek to explore the methodology introduced in Conti et al. [6] by applying data mining tools [9] to enhance the crowd counting strategy. Our purpose, then, is to first differentiate mobile from static devices before estimating the crowd number through the unique MAC addresses in a given environment. In the next section we will present the background knowledge for understanding the Wi-Fi MAC layer, as well as some related works. Following, in Section 3, we will detail the proposed architecture, as well as the data acquisition and classification process. Further, in Section 4, we will discuss the results we have obtained followed by some conclusions in Section 5.

2. BACKGROUND

In this section we provide the reader with details about the MAC layer of the IEEE 802.11 standard, which is the basis of any Wi-Fi service. Following it, we will present some related works also based in this same protocol.

2.1 Wi-Fi MAC Layer

The IEEE 802.11 protocol defines several types of management packets that are exchanged among APs and the Wi-Fi devices during a communication session. It is important to notice that this protocol is based on the OSI model¹. In this work we are particularly interested only on the *Media Access Control* (MAC), which is defined as a sublayer of the OSI data link layer and is responsible for managing the access to the physical air interface that is shared among all wireless devices.

In the OSI conceptualization, each data packet is made of several encapsulated frames. The MAC frame has a header, as shown in Figure 1, that among others is composed by three fields of 6 bytes each, which in turn identify a device – known as *MAC Address*. Thus the MAC address is a unique identifier assigned to a Network Interface Controller (NIC) for communications at the data link layer of any *IEEE 802.x* network [10]. Considering the crowd counting problem, if we assume each person in the area holds one Wi-Fi device, it can be associated to the person as a user ID.

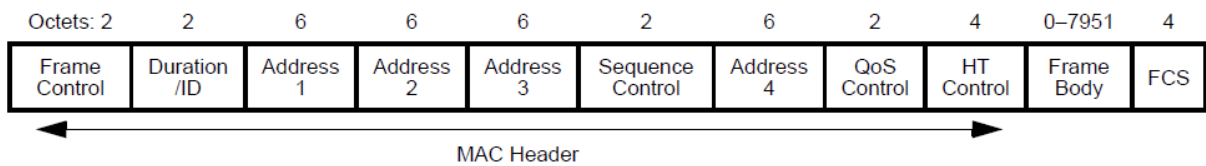


Figure 1: A general MAC frame structure for the IEEE 802.11 standard (adapted from [10]).

A MAC address consists of six sets of two hexadecimal characters. The first three character pairs are called OUI – Organizationally Unique Identifier – and uniquely identifies a vendor or manufacturer. There are around 600 suppliers officially considered as NIC manufacturers². This reference is used to identify fixed and mobile devices. However, as mentioned before, most manufacturers do not clearly distinguish how the MAC addresses are distributed among their different classes of products.

2.2 Related Works

Current algorithms for a crowd estimation can be grouped into two main categories: image-based and non-image-based methods. While the former have high computational costs, the latter offer the advantages of being non-expensive and non-intrusive alternatives [11].

Most of the image-based approaches use deep learning and other computer vision techniques to perform crowd counting [12, 13]. In this context, there are studies focusing on improvements of the neural networks architectures [12], while others propose to apply Intelligent Video Surveillance (IVS) systems by using the videos to estimate the number of people [14].

Among the non-image-based approaches, the use of Wi-Fi based strategies has gained special interest, due to the accuracy and the economic aspects when compared to the former methods. For instance, Conti et al. [6] proposed a scheme that achieved 83.3% of accuracy in estimating the number of people in a room by using management packets from a Wi-Fi network, known as *probe request* packets.

Probe request packets were also used by Yaik and Wai [3] to show the correlation between the number of new MAC addresses collected and the number of people in a given area. Using a similar approach, the Euclid Analytics Company³ has captured Wi-Fi probe request packets to estimate the audience size, the length of stay and the number of visitors. These data were stored and then applied to decision-making support systems to help business managers. In a quite similar strategy, but now considering the *Channel State Information* (CSI) instead of the probe request packets, Xi [15] used Wi-Fi radios to study the relationship between the number of moving people and the variation of CSI parameter. Adib [16] also considered the CSI parameter to detect people movement in a room, as well as estimating their number, by using a MIMO (Multiple-Input and Multiple-Output) set of antennas.

Machine learning algorithms have been also applied in such experiments. For instance, Wang et al. [17] proposed a deep learning method to improve the indoor localization using Wi-Fi. The model proposed used CSI information in a deep neural network architecture. Although the proposed experiment was performed in just two different indoor environments, the results showed that location error could be effectively reduced, when compared with other three methods. Similarly, Cheng [11] used a three-layer deep neural network architecture and proposed a people counting system for indoor areas, being able to discriminate up to nine people with 88.66% of accuracy.

¹Open Systems Interconnection model, which characterizes a communication session through seven abstraction layers: (i) physical; (ii) data link; (iii) network; (iv) transport; (v) session; (vi) presentation; and (vii) application.

²<http://standards-oui.ieee.org/oui.txt>

³<http://euclidanalytics.com/product/malls/>

The references above show contributions to the crowd density estimation by both image-based and non-image-based approaches. We can notice that machine learning can be applied in both strategies, either to improve performance on images, or in the classification of parameters such as the CSI data. In this paper, we seek to apply machine learning to classify devices into static or mobile categories. This approach can be used as an alternative to improve previous works, such as those of Yaik and Wai [3] and Conti et al. [6].

3 METHODS

The main idea of our study is to apply machine learning to classify devices into fixed or mobile ones and, doing so, improve the accuracy of algorithms for crowd counting using Wi-Fi. In order to achieve it, we propose a methodology, as depicted in the flowchart of Figure 2, which comprises two main stages: (1) data acquisition and (2) data processing.

Due to the lack of labeled data for training classifiers for this purpose, the first step (1) is to obtain such data. This is accomplished by collecting the IEEE 802.11 frames for a given time and store them all in a cloud server storage, as detailed in Section 3.1.

In the next step (2), the data must be processed in such a way to extract features from them, as well as to understand their behavior. At the end of this stage, we train a classifier that is capable of segmenting the Wi-Fi packets sources into fixed or mobile devices. This process is detailed in Section 3.2.

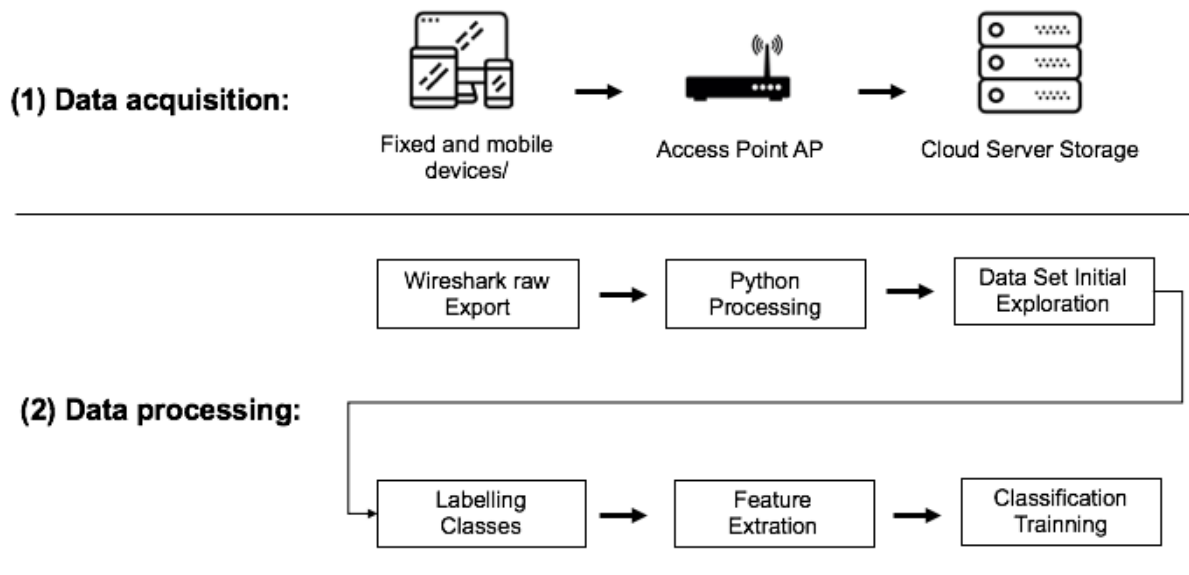


Figure 2: Flowchart detailing the proposed method in two different stages: (1) data acquisition: a probe system captures data from Wi-Fi devices and stores them in a cloud server; (2) data processing: different algorithms are used to analyze the data, extract their features, and train a classifier.

3.1 Data acquisition

In order to collect data, two different environments were set up following the same dispositions shown in Figure 3. The first one (Fig. 3a) was placed at the 5th floor of a corporate office, while the second one (Fig. 3b) was placed at the ground floor, in a food court of a large mall.

In the corporate office, there were approximately 100 people by floor. While most of them remained seated, few people were walking in the corridors of the building. Once each person had a smartphone and a personal computer with a Wi-Fi card, this local can be considered as a high-density Wi-Fi location. The data acquisition process lasted 1 hour and 15 minutes and was performed at the busiest period of the day (between 9:00 to 10:15 AM). From this environment, we collected data from 25 smartphones and 16 computers, whose MAC addresses were previously known. These data were used to build the *training dataset*.

In the second place (Fig. 3b), there were approximately 80 people in the area covered by the AP. It is worth noting that here people behaved in a different way from the previous place: while a group of people was periodically checking their smartphones while eating, another group was moving around the place while using their devices. The data acquisition process here lasted 30

minutes. All data collected in this environment came from unknown devices (i.e. there was no information if the device was a smartphone or a computer). These data were used to build the *test dataset*.

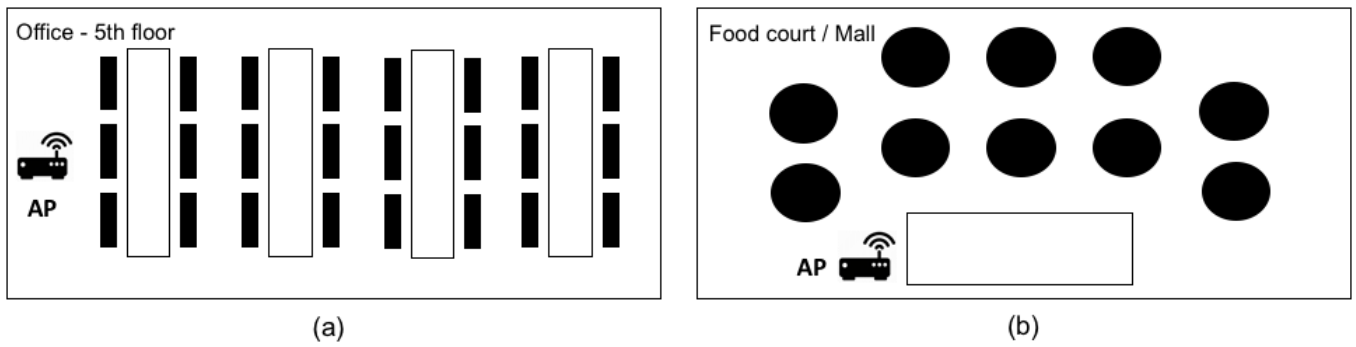


Figure 3: Layout of the environments where data were acquired for (a) training and for (b) test.

In addition to the preparation of the physical environment with the APs provisioning, the data acquisition stage also consists of logical steps for preparing the data, i.e., the data cleaning; the class assignment and the exporting to an appropriate file format before being processed. Figure 4 shows the steps applied to each of the datasets during the data clearing process, which will be detailed in the next sections.

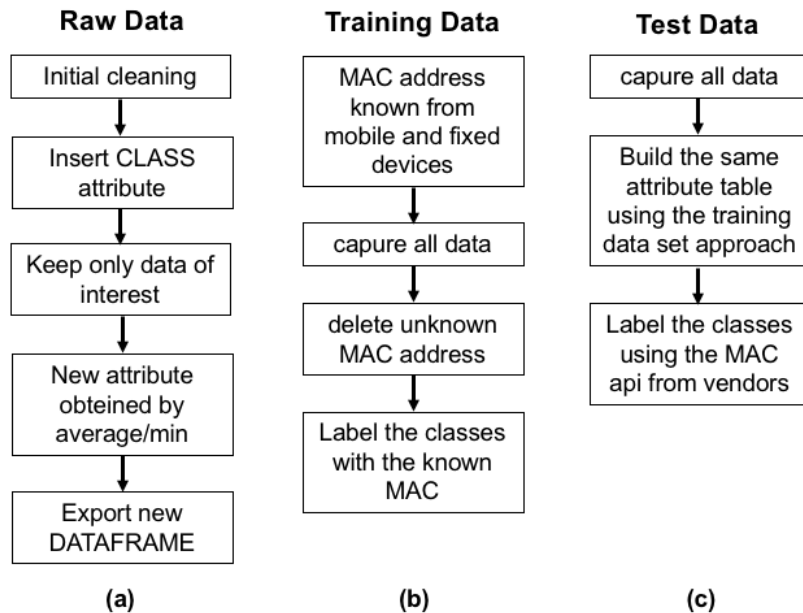


Figure 4: Three flowcharts detailing the difference between raw, training and test datasets. (a) Raw dataset with the fields without formatting. (b) Training dataset with the classes included. (c) Test dataset using the classes from vendor API.

3.1.1 Raw data

The same architecture created to collect the data was used to build both the training and test data. The system was set up to capture all the traffic of the radio interface from an Access Point (AP) and record them in a database. After examining the raw captured data, they were converted into information on the formatted datasets. The system is composed by the AP and servers connected to the public Internet. Besides capturing all wireless traffic, the AP is also in charge of executing scripts and send the data to servers.

The AP model chosen for the experiments was the TP-Link TL-WDR3600⁴ running OpenWRT – an embedded Linux distribution⁵ designed for wireless routers and compatible with many manufacturers. This was done to circumvent the limitations

⁴http://www.tp-link.com.br/Resources/document/TL-WDR3600_V1_datasheet.zip

⁵<https://openwrt.org/>

imposed by some operating systems or firmwares, since they do not allow the in-depth exploration of the wireless communication attributes. On the other hand, OpenWrt distribution allows several customization, enabling the possibility to keep the radio interface in monitor mode. This mode allows capturing layer-2 traffic using a non-intrusive approach, therefore getting data from the 802.11 protocols wireless transmitted in a given area. Table 1 defines the attributes collected from the packet header and which compose the initial dataset.

Table 1: Attributes extracted from 802.11 packet headers which compose the initial dataset.

Attribute	Definition
ArrivalTime	Frame arrival
SourceAddress	Device address
RSSI	Received signal
Length	Frame size
Type/Subtype	Frame type
DataRate	Data Rate
Channel	Wi-Fi channel
FrameControlField	Frame information
Duration	Frame duration
PHY Type	Defined modulation

3.1.2 Training dataset

The data originally acquired in frames were converted into a time series. Next, the instances with empty or null values were removed. In order to be able to perform a supervised classification, a target column was added to the dataset, defining whether the MAC address corresponds to a computer or smartphone.

One of the most important steps for building the training dataset was to segregate the “Type/Subtype” attribute into a new attribute by averaging data in a 1-minute time rate. This attribute stores different type of control message from the 802.11 control protocol. The following control parameters were selected to be used in this work: *802.11 Block Ack*, *802.11 Block Ack Req*, *Association Request*, *Authentication*, *Data*, *Deauthentication*, *Null function*, *Probe Request*, *QoS Data*, *QoS Null function*, *Reassociation Request*, and *Request-to-send*.

In order to create a distribution of the information across time, the dataset was enriched with new attributes related to the above-mentioned parameters. If such attribute appeared in a given moment for the same instance, a marker was assigned to its corresponding field. Although in some studies (see, for instance, [18] and [19]) the standard deviation had a behavior more stable than the average, the dataset in this work was created in such way that the attributes reflected the average per minute found in the incoming data.

A total of 16 attributes and 2178 samples were included in the training dataset. From these samples, 801 belonged to the *computer* class, while 1377 belonged to the *smartphone* class. Since the amount of samples in each class was different, the dataset can be considered unbalanced. Table 2 shows the final attributes of the training dataset.

3.1.3 Test dataset

The test dataset was obtained using the same strategy applied before to build the training set, both built using a Python script running on the Jupyter⁶ framework. The main difference between the training and test datasets, in addition to the location they were obtained, is that in the latter the target class was defined using information from API vendors⁷, as depicted in Figure 4.

Since the vendor name was associated with the MAC address captured from the device, we defined a criterion to choose which vendors should be in the dataset: only those that had the word *mobile* in the vendor information field. Following this heuristic, only smartphones were guaranteed to be included in this dataset. Therefore, 2886 samples were included in the one-class test dataset.

⁶<http://jupyter.org/>

⁷<https://macvendors.com/>

Table 2: Attributes of the training dataset constructed over a time series.

#	Attribute	Definition
1	length/min	Frame size per minute
2	datarate/min	Average rate per minute
3	channel/min	Channel per minute
4	duration/min	Duration per minute
5	blockack/min	ACK per minute
6	blockackreq/min	ACK per minute
7	assocreq/min	Association per minute
8	authen/min	Autentication per minute
9	deauthen/min	De-Autentication per minute
10	null/min	Frame power per minute
11	probereq/min	Probe requests per minute
12	qos/min	QoS information per minute
13	qosnull/min	Null QoS information per minute
14	rerequest/min	Retransmission per minute
15	resend/min	Resend per minute
	class	[computer, smartphone]

3.2 Data processing

3.2.1 Exploratory Data Analysis

In the Knowledge Discovery in Databases (KDD) process, the Exploratory Data Analysis (EDA) seeks to unveil possible interactions among the variables collected during the data acquisition process (Section 3.1). In order to obtain a visual exploration of the training data, scatterplots, boxplots and violinplots were done using Python with the Seaborn library ⁸.

3.2.2 Feature selection

Using the training and test datasets, previously described, classifiers were trained and their performance assessed, as explained in Section 3.3.

However, before applying classification algorithms, an important issue has to be addressed: the dimensionality reduction, and this can be achieved by using feature selection methods. This procedure reduces the amount of data by eliminating noisy and irrelevant attributes.

Feature selection methods are usually grouped into two main categories: filter, and wrapper (although a third category – embedded methods – is sometimes referred in the literature) [20, 21]. The former are based only the statistical properties of the data, such that features are ranked based on their individual ability to separate data into the correct classes. On the other hand, wrapper methods are fundamentally based on a search in the space of possible combinations of features. A subset of features is later used to train a classifier, and its performance regarding actual classes of the original data is used to rank the subset of features, instead of individual features.

In this work we used three filter methods (Information Gain, Chi-Squared test, and Symmetrical Uncertainty), and two wrapper methods (Cfs+BestFirst and GA+JRip) [22]. The Cfs+BestFirst methods was proposed by [23] and searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility. Then, each feature subset is evaluated considering the individual predictive ability of each feature along with the degree of redundancy between them. The GA+JRip method uses a Genetic Algorithm [24] to perform a search in the space of possible combinations of the input features, and JRip (Repeated Incremental Pruning to Produce Error Reduction classifier [25]) to evaluate results. The Genetic Algorithm running parameters were: population size=20, mutation probability=0.033, crossover probability=0.6 and maximum number of generations=20. Both wrapper methods used a 10-fold cross validation procedure for evaluating each feature subset.

All the above-mentioned feature selection methods either give a merit value for each attribute (filter methods) or give a subset of attributes (wrapper methods). To combine all of them, a normalized average merit table was generated from the individual results. Next, a consensus merit was computed taking into account the normalized contribution of each feature selection method.

⁸<https://seaborn.pydata.org/>

The top-9, top-7 and top-4 best ranked attributes are shown in Figure 5. Also, Table 3 summarizes the features selected by the above-mentioned procedure and the wrapper methods.

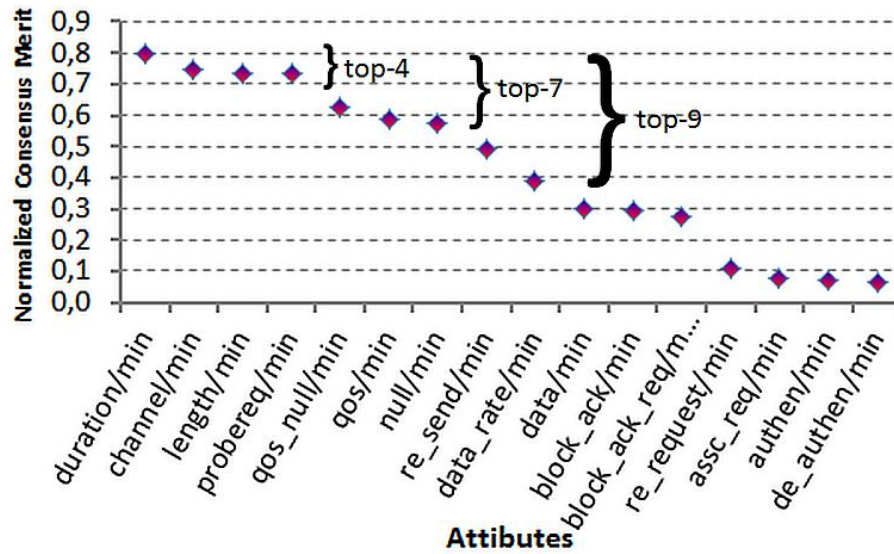


Figure 5: Normalized consensus merit for each attribute.

Table 3: Attributes selected by different feature selection methods.

#	Attribute	Consensus all methods			Wrapper methods	
		top-9	top-7	top-4	Cfs+BestFirst	AG+Jrip
1	length/min	X	X	X	X	X
2	data_rate/min	X				
3	channel/min	X	X	X	X	X
4	duration/min	X	X	X	X	
5	block_ack/min					X
6	block_ack_req/min					X
7	assc_req/min					
8	authen/min				X	
9	data/min				X	X
10	de_authen/min					
11	null/min	X	X		X	X
12	probereq/min	X	X	X		X
13	qos/min	X	X			
14	qos_null/min	X	X			X
15	re_request/min					
16	re_send/min	X				

3.3 Classification

All classification experiments were performed using the Weka⁹ [26] data mining platform, which has an easy-to-use interface, as well as many methods implemented. The following machine learning methods were used: OneR (standing for *One Rule*, which uses the attribute with the highest Information Gain to construct a single classification rule for all classes), LR (Logistic

⁹<https://www.cs.waikato.ac.nz/ml/weka/>

Regression, also known as logit model or maximum entropy classifier), J48 (C4.5-based rule induction algorithm [27]), JRip, and MLP (Multilayer Perceptron neural network [28]). The first method establishes the lower bound for evaluating the performance of classifiers, i.e., a method based on a very simple heuristic that yields a baseline performance metrics by which the remaining advanced methods are further compared. Similar to the feature selection phase, all classification process experiments were done by using a 10-fold cross-validation scheme. Using the training set, the classification methods were run with (i) all features, (ii) top- k features (with $k = [9, 7, 4]$), (iii) features selected by the wrapper methods.

After the training and test steps were concluded, we used the J48 classification method again to extract some high-level knowledge from the rules it generates. In some cases the classification rules can offer human comprehensible knowledge that gives insights about the data and the problem. In this work, from the decision tree generated by the J48 algorithm, we observed that the attributes *duration/min* and *null/min* play an important role in this problem, as shown in Figure 7, and discussed in the next section.

4 RESULTS AND DISCUSSION

In this section, firstly, we present and discuss some results of the EDA (Section 3.2.1). Next, we will present the classification results and bring some considerations about the performance of the classifiers presented in the Section 3.3.

During the EDA, several plots were visually analyzed. Figure 6 shows some relevant patterns unveiled regarding the following attributes: *null/min*, *datarate/min*, *channel/min*, and *qos/min*.

In Figure 6(a), it is observed significant different patterns of the attributes for the training classes. Particularly, the *null data frames* occurrence and variance are considerably larger for smartphones than for computers. A possible explanation for this fact is that *null data frames* are considered by the IEEE 802.11 standard as a kind of special frame that is widely used for energy control, conduit scanning and for keeping-alive association [29]. Such data frames are also used by low-power devices to signalize they entered in hibernation for some milliseconds, a procedure that would allow a better energy management and power saving, while the AP buffers some packets waiting for a next *null frame*, when the device comes back from its hibernation.

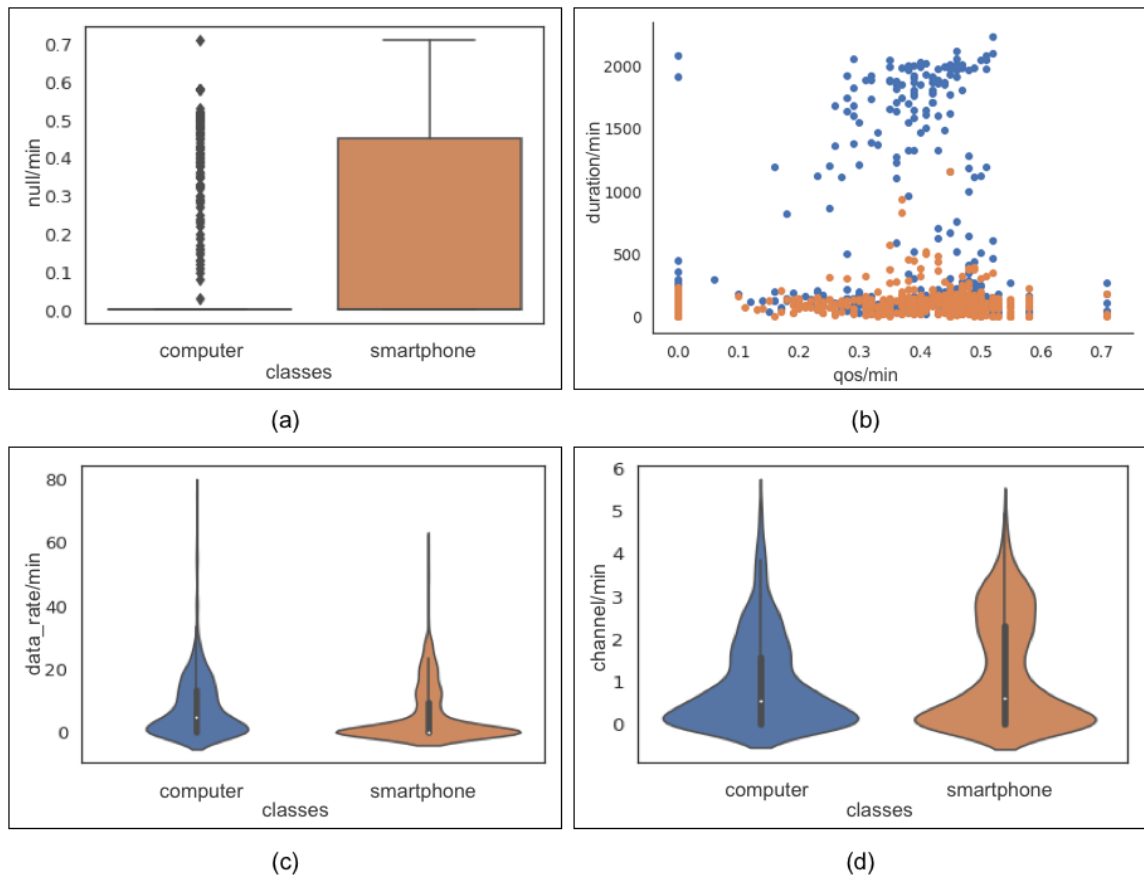


Figure 6: Exploratory Data Analysis: (a) Distribution of the *null/min* attribute among computer and smartphones. (b) Correlation among *datarate/min* and *qos/min* attributes. (c) Distribution of the *datarate/min* attribute. (d) Distribution of the *channel/min* attribute.

It is worth noting that while the *null frames* were considered irrelevant so far, they are unveiled now as a critical feature in this classification process.

The *duration* attribute (Table 1) is related to the frame transmission time and, for the classification process (Table 2) it is averaged per minute. However, on the other hand, the value of the *QoS* field is set by the device according to its application requirements (e.g. voice applications usually require higher *QoS* – Quality of Service – than data applications), which becomes a considered attributed when averaged by minute. The correlation among these both attributes is shown in Figure 6(b). Although our initial hypothesis was that computers could handle large packets and, in turn, would outlast those packets from smartphones, we can notice that there are many computer devices with duration rates over 500 frames/min. Consequently, devices cannot be clearly identifiable for *duration* values below that threshold.

It is important to notice that the findings from Figure 6(a)(b) are consistent with the rules later generated by the J48 algorithm, shown in Figure 7. Such rules could be easily implemented in embedded devices, making it easier to implement crowd counting algorithms.

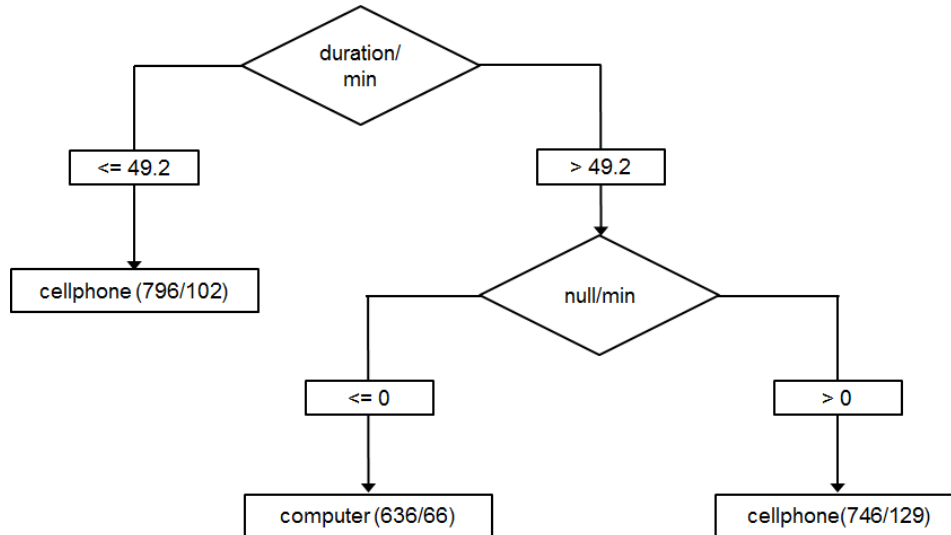


Figure 7: Final decision tree from the J48 algorithm. The *duration/min* and *null/min* outstands as the main attributes for this classification task.

Regarding the *data_rate/min* and *channel/min* attributes, the violin plots shown in Figure 6(c-d) add to the summary statistics the attributes distribution. In this context, from Figure 6(c) we can notice that there is a high concentration of smartphones in the low *data_rate/min* region, while computers stand in the higher ones. Regarding the channel variance, computers tend to be more stable than smartphones, as observed in Figure 6(d). A possible interpretation for this result is that once the computer are kept at the same place, i.e. they are static, they present a better data rate and an optimized use of Wi-Fi channels than smartphones. On the other hand, smartphones are moving frequently, and then the modulation process between their antennas and the AP needs to adapt the data rate along the connection time.

All the classifiers were trained with different sets of attributes, as mentioned in Section 3.3. The overall best performing set of attributes were those selected by the wrapper methods AG+JRip. Table 4 compares the results obtained with the several classifiers for both, training and test datasets. We evaluated the area under the Receiver Operating Characteristics curve (ROC Area) and the F-Measure (harmonic average between the precision and the recall measures).

During the training process, the attributes selected by the Wrapper AG+JRip method achieved a better performance than those selected by a consensus merit table. Therefore, the experimental results shown in Table 4 were performed using attributes 1, 3, 5, 6, 9, 11, 12, and 14 selected by the wrapper AG+JRip method (see Table 3).

From the classification point of view, the OneR was taken as a baseline in this work for comparing the classification methods only in the training step. In general, the classification methods (J48, JRip, MLP and LR) performed from 13–51% better than the baseline, considering the ROC Area, and 23–39%, considering the F-Measure. Considering that the Training dataset had two unbalanced classes (more smartphones than computers), all the classifiers were able to achieve a good balance between classes.

However, comparing results on the Test dataset, J48 and JRip had their performance decreased regarding to the Training dataset. This fact suggests some overfitting in those models. On the other hand, MLP and LR showed more robustness for classifying unseen data. The overall best performance was achieved by the LR algorithm, along with the MLP algorithm.

Table 4: Comparative performance of the classifiers on the training and test datasets.

Dataset	Classifiers	ROC Area	F-Measure
Training	OneR	0.62	0.66
	J48	0.94	0.92
	JRip	0.90	0.90
	MLP	0.89	0.87
	LR	0.83	0.81
Test	J48	0.89	0.81
	JRip	0.86	0.80
	MLP	0.96	0.85
	LR	0.99	0.93

5 CONCLUSIONS

Due to the popularization of smartphones, estimating the number of people in crowded areas using Wi-Fi has become more relevant. Assuming that, on average, each person has a (single) Wi-Fi device, we can estimate the number of people by counting the number of devices in a given area. However, different Wi-Fi devices may be in the same area and, in this case, some methods fail when trying to count smartphones.

In this study, we trained classifiers to differentiate mobile from static devices using behavioral patterns from the traffic data at the MAC layer of the IEEE 802.11 standard. In our proposed solution, we built two distinct datasets, one for training and other for testing. The best-performing classifier was Logistic Regression classifier, achieving 0.99 (*ROC Area*). Based on these results, this study showed the feasibility of the method to classify smartphones and computers in two groups, in areas with high-density of unknown devices. These promising results foster future work towards massive real-time crowd counting using Wi-Fi.

REFERENCES

- [1] D. Ryan, S. Denman, C. Fookes and S. Sridharan. “Scene invariant multi camera crowd counting”. *Pattern Recognition Letters*, vol. 44, pp. 98 – 112, 2014.
- [2] Y. Hu, H. Chang, F. Nian, Y. Wang and T. Li. “Dense crowd counting from still images with convolutional neural networks”. *Journal of Visual Communication and Image Representation*, vol. 38, pp. 530–539, 2016.
- [3] O. B. Yaik, K. Z. Wai, I. K. Tan and O. B. Sheng. “Measuring the accuracy of crowd counting using Wi-Fi probe-request-frame counting technique”. *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 8, no. 2, pp. 79–81, 2016.
- [4] H. Jacobs. “To count a crowd”. *Columbia Journalism Review*, vol. 6, no. 1, pp. 37, 1967.
- [5] C. McPhail and J. McCarthy. “Who counts and how: estimating the size of protests”. *Contexts*, vol. 3, no. 3, pp. 12–18, 2004.
- [6] J. P. Conti, T. B. N. da Silveira and D. P. Araujo. “Dynamic crowd counting via 802.11 MAC layer”. In *Proc. IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 13–14, 2016.
- [7] Wifi Alliance. “Annual Report 2012”. Technical report.
- [8] IANA. *IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters*, 2008.
- [9] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand and D. Steinberg. “Top 10 Algorithms in Data Mining”. *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2007.
- [10] IEEE. *IEEE 802.11-2016 – IEEE Standard for Information technology – Telecommunications and information exchange between systems Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*.

- [11] Y.-K. Cheng and R. Y. Chang. “Device-Free Indoor People Counting Using Wi-Fi Channel State Information for Internet of Things”. In *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2017.
- [12] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye and C. Zhu. “Fast crowd density estimation with convolutional neural networks”. *Engineering Applications of Artificial Intelligence*, vol. 43, pp. 81–88, 2015.
- [13] K. Han, W. Wan, H. Yao and L. Hou. “Image Crowd Counting Using Convolutional Neural Network and Markov Random Field”. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 21, no. 4, pp. 632–638, 2017.
- [14] Z. Zhang, W. Yin and P. L. Venetianer. “Fast crowd density estimation in surveillance videos without training”. In *Proc. IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pp. 452–457, 2012.
- [15] W. Xi, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu and Z. Jiang. “Electronic frog eye: Counting crowd using wifi”. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, pp. 361–369, 2014.
- [16] D. Adib, F. ; Katabi. “See through walls with WiFi”. *Computer Communication Review*, vol. 43, no. 4, pp. 75–86, 2013.
- [17] X. Wang, L. Gao, S. Mao and S. Pandey. “CSI-based fingerprinting for indoor localization: A deep learning approach”. *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2017.
- [18] Y. Gu, L. Quan and F. Ren. “WiFi-Assisted Human Activity Recognition”. In *Proc. IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 60–65, 2014.
- [19] Q. L. Gu, Y. and F. Ren. “AAH: accurate activity recognition of human beings using WiFi signals”. *Concurrency and Computation: Practice and Experience*, vol. 28, pp. 3910–3926, 2016.
- [20] I. Inza, P. L. naga, R. Blanco and A. J. Cerrolaza. “Filter versus wrapper gene selection approaches in DNA microarray domains”. *Artificial Intelligence in Medicine*, vol. 31, pp. 91–103, 2004.
- [21] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection*. Chapman & Hall/CRC, Boca Raton, FL, USA, 2008.
- [22] R. Kohavi and G. H. John. “Wrappers for feature subset selection”. *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.
- [23] M. A. Hall. “Correlation-based Feature Subset Selection for Machine Learning”. PhD Thesis, Department of Computer Science, The University of Waikato, Hamilton, New Zealand, 1999.
- [24] A. F. de Pinho, J. A. B. Montevechi, F. A. S. Marins and R. de Carvalho Miranda. “Algoritmos Genéticos: Fundamentos e Aplicações”. In *Meta-Heurísticas em Pesquisa Operacional*, edited by H. S. Lopes, L. C. de Abreu Rodrigues and M. T. A. Steiner, chapter 2, pp. 21–32. Omnipax, Curitiba, PR, 2013.
- [25] W. W. Cohen. “Fast Effective Rule Induction”. In *Twelfth International Conference on Machine Learning*, pp. 115–123. Morgan Kaufmann, 1995.
- [26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten. “The WEKA data mining software: an update”. *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.
- [27] J. R. Quinlan. *Programs for Machine Learning*. Morgan Kaufmann Publishers, San Francisco, CA, 1993.
- [28] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Delhi, India, second edition, 1998.
- [29] W. Gu, Z. Yang, D. Xuan, W. Jia and C. Que. “Null Data Frame: A Double-Edged Sword in IEEE 802.11 WLANs”. *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 897–910, July 2010.