Published version Digital Object Identifier 10.1109/JRFID.2022.3185804

Edge Machine Learning techniques applied to RFID for device-free hand gesture recognition

Massimo Merenda, *Member, IEEE*, Giuseppe Cimino, Riccardo Carotenuto, *Senior Member, IEEE*, Francesco G. Della Corte, *Senior Member, IEEE*, and Demetrio Iero

***Abstract*—Gesture recognition is a novel technology that aims to change the way people interact with machines. Existing solutions typically recognize gestures using camera vision, wearable sensors, or specialized signals (e.g., WiFi, sound, and visible light), but they have limitations such as high-power consumption or low signal-to-noise ratio (SNR) in comparison to their surroundings, making it difficult to accurately detect finger movements. In this research, we propose a device-free gesture identification system that recognizes different hand movements by processing through Edge Machine Learning (EML) algorithms the received signal strength indication (RSSI) and phase values from backscattered signals of a collection of Radio Frequency IDentification (RFID) tags mounted on a plastic plate. The performances of three algorithms, the Random Forest Classifier, the Support Vector Machine, and the Decision Tree Classifier, were compared giving very encouraging results with accuracy up to 99.4%.**

Keywords*—*edge machine learning, hand gesture, RFID, RSSI, sensorless

I. Introduction

Hand gesture is used in the everyday lives of persons and is an undeniable element of body language in the interaction of human beings. With the development of computing devices, the research of natural interaction between humans and computing devices has become increasingly important. Hand gesture recognition is gaining more and more attention for use in different areas, such as virtual devices, virtual and augmented reality (VR and AR), safety in workplaces, especially when handling heavy machinery used in manufacturing, such as in the casting process [1].

Several works for gesture recognition have been carried out using camera or application-specific sensors [2]–[4], based on computer vision, localization systems [5], [6] or time-of-flight [7] cameras. This kind of gesture interface has attracted attentions because it does not require additional hardware from the user perspective.

A simple solution for RFID-based gesture recognition is to utilize RFID localization schemes [8] to directly locate tagged objects [9]. Generally, gesture recognition systems mainly use phase values to achieve an accurate localization [10] or a combination of RSSI and phase values [11]–[13].

Recently, the use of RFID tags on special gloves [11] or body parts [14], [15] is widespread in the literature, which identifies RFID technology as a very promising candidate for this type of task. Nevertheless, the approach used in [11]-[12], [14]-[16] limits the usability due to the need of wearing special clothes and/or a fixed positioning of the tags.

Indeed, in some other works, the tags are not attached to body parts or objects to be located but are placed in the surrounding environment, and the modified backscattered signal is analyzed [16]; the number of tags used to provide a fine-grained resolution of the gesture recognition is quite high [13], while providing the higher resolution in terms of gesture recognized. The approach that relies on the backscatter signal analysis, however, causes a reduction of the read range of the system in the order of centimeters [13], [16]. Other approaches require specific design of augmented tags with sensing capabilities [17]–[21] or protocol enhancements [22]. In addition, the use of recent machine learning (ML) models [23] extends the field of application of RFID device-free approach to gesture recognition [16]. The systems found in the literature fail to provide at the same time: i) a compact solution that can be implemented on devices with limited resources; ii) a good accuracy in gesture recognition; iii) relatively small number of tags not attached to the hands or objects performing the gesture; iv) operation without requiring a sophisticated setup or a larger number of antennas.

In this paper, we present a device-less gesture recognition technique that leverages the use of ML, based on the inference of RSSI and phase values from the backscattered signals as measured by an array of tags placed on a plastic plate, extending the results previously presented in [24]. The main additions to the previous work are the extension of the measurement campaigns for the dataset gathering, the usage of two antennas, an increased number of tags, the use of a higher number of ML models trained and compared, and finally the implementation on an edge device. The above improvements have resulted in a comprehensive analysis of the results and improved accuracy levels, which are discussed in detail in the following sections.

The proposed technique novelties are mainly associated to the use of:

1) a single channel frequency to avoid post-processing for phase dewrapping;
2) two antennas and evaluation of the performance of the system considering alternatively: both antennas, only

M. Merenda is with the Center for Digital Safety & Security, AIT Austrian Institute of Technology GmbH, Vienna, 1210, Austria (e-mail: massimo.merenda@ait.ac.at).

G. Cimino is with the DIIES Department, University Mediterranea of Reggio Calabria, Reggio Calabria, 89124, Italy (e-mail: cmngpp97e10h224t@studenti.unirc.it)

R. Carotenuto is with the DIIES Department, University Mediterranea of Reggio Calabria, Reggio Calabria, 89124, Italy (e-mail: r.carotenuto@unirc.it)

F. G. Della Corte is with the Università di Napoli Federico II, 80125 Napoli, Italy, (e-mail: francescogiuseppe.dellacorte@unina.it).

D. Iero is with the DIIES Department, University Mediterranea of Reggio Calabria, Reggio Calabria, 89124, Italy (e-mail: demetrio.iero@unirc.it)

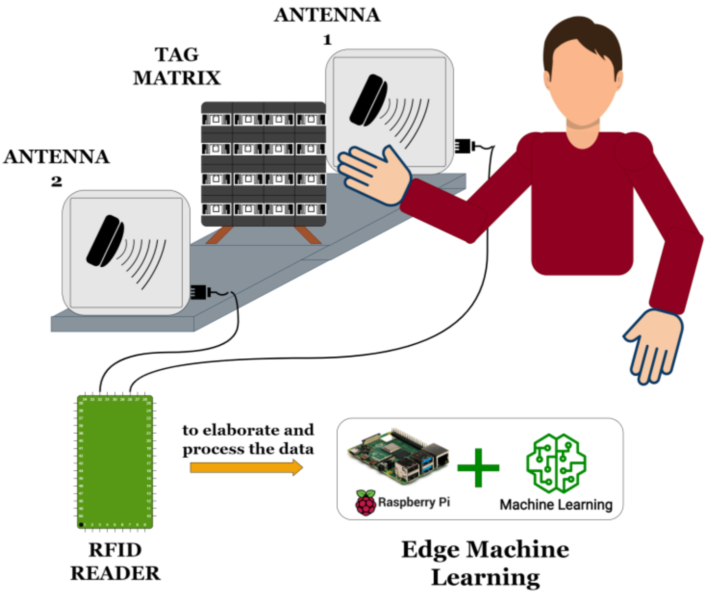

Fig. 1. Gesture recognition system setup.

antenna 1 (placed in front of the tags) and only antenna 2 (placed behind the tags);
3) 16 tags;
4) a single dataset obtained in our laboratories with the hand in a static position, performed by 5 people, in two different locations and varying the relative distance from antennas to tags (station 1: 30 cm and 40 cm; station 2: only 30 cm);
5) the Edge Machine Learning (EML) capabilities for the inference of the gesture recognition on a Raspberry PI 3 model b device.

Throughout the measurement campaign detailed in the following sections, we trained three different models, namely Random Forest Classifier (RFC), Support Vector Machine (SVM), and Decision Tree Classifier (DTC), with the data coming from the complete dataset obtained by using both antennas, and from two datasets derived using only data from Antenna 1 and Antenna 2, separately.

As we will see in the following, all three ML models showed high performance on all the considered metrics, obtaining accuracy values of over 90%, with RFC achieving the best results.

In Section II, RFID technology and its low-level features (RSSI and phase), used to discriminate the gestures, are introduced. In Section III, the architecture and set-up of the gesture recognition system are described. In Section IV, the datasets and the ML models employed are described together with the EML implementation. Section V reports the performance results of the ML models trained on the provided datasets, as per the evaluation metrics employed. Lastly, the conclusions are drawn in Section VI.

II. RFID Working Principle

Passive Radio Frequency Identification (Passive RFID) is the technology chosen for the development of the gesture recognition system. RFID is a technology engineered to both identify and track objects using RFID tags. Passive RFID tags have no batteries and work by harvesting energy from the RF signal produced by the reader.

Passive RFID systems working in the UHF (Ultra High Frequency) band gather the data stored in the tags by means of the backscattering mechanism. The reader emits an RF signal, and the tag is energized allowing it to activate its microchip. The absorption of the impinging energy is modulated based on the data stored in the chip and a modulated reflected wave is sent back to the reader. Then, the reader intercepts the backscattered signal and evaluates the contained information.

In our design, the low-level information, made available by the RFID technology standard, and used to characterize the different gestures are RSSI and phase. RSSI is the power level of the RF signal received by the RFID reader. The Phase is a measure of the displacement angle between the RF carrier transmitted by the reader and the return signal from the tag. The RFID reader performs frequency hopping from one channel to another and, as a result, the actual phase values are dependent also on the switching of the channel frequency.

III. Gesture Recognition System

The system exploited for gesture recognition consists of sixteen passive Impinj Monza 5 RFID tags [25], an RFID Reader ThingMagic M6e Micro UHF RAIN [26], and two antennas Laird S8658PLJ (LHCP) [27] as shown in the setup reported in Fig. 1.

The sixteen tags are laid out on a plastic plate supported by a wooden support, to prevent metal surfaces from interfering with the data acquisition. In fact, it is proven that the presence of metallic materials impacts the strength of the signal received and thus the RSSI value [28]. Each RFID tag was programmed with a unique EPC (Electronic Product Code) so that each individual tag could be easily identified across acquisitions. The antennas used were positioned at the same distance from the tag array, one in front (Antenna 1) of the tag array and the other behind (Antenna 2). For the acquisitions, the hand was inserted only between the Antenna 1 and the plate that supports the matrix of tags. For this study, five hand gestures were analyzed, each one associated with a numeric label ranging from 1 to 5.

Furthermore, the case of absence of the hand between the tags and the Antenna 1, called gesture 0, was considered. In addition, we also considered case of no hand between the antenna and the tags' plate, and it was referred to as gesture 0. The same setup has been used to perform the hand gestures and acquire data that compose the training dataset.

For the reading operations, we developed a C# application that exploits the ThingMagic Mercury API library, to setup the reader and to handle the operations. The same software library allows for gathering data from the tags detected by the reader. In addition, the Universal Reader Assistant (URA) software was used, which provides a complete and thorough interface for the ThingMagic RFID readers.

By default, the RFID reader has been set to operate at a fixed frequency of 865.7 MHz. In normal working mode, the reader is set by default to work in frequency-hopping mode, which consists of the reader changing the working frequency, at every reading, in a given frequency range following to the operating region of the reader. However, this behavior leads to a discontinuity in the saved phase values over time, which makes them unsuitable for the given purposes, unless a phase correction formula is applied [11].

Moreover, the reader is configured to use the EPC Gen2 protocol and an output power of 30 dBm. The reader is polling every 300 ms (150 ms Antenna 1 + 150 ms Antenna 2), for every tag in the range, the RSSI, the phase, and the EPC unique code of the backscattered signal is acquired. This timing has been determined as a trade-off between the ability to properly read the appropriate information from all tags and the speed at which the acquisition is performed.

The tags are placed on the far-field zone border. We positioned them on the border since the tags we employ have a technological limitation: following several preliminary tests, we discovered that the reader struggled to receive data from all 16 tags beyond 40 cm, hence we set the upper limit at such distance. Since going to lower distance would violate the near field region, which is not a goal in our study, the lower limit was set at 30 cm.

Moreover, the distances must be sufficient to allow for easy positioning of the hand between tags and antenna, as well as accurate data collecting at rapid rates, which is why distances of 30-40 cm are more than sufficient. To switch from one gesture to another without colliding into the antenna and/or tags, a spacing of 30-40 centimeters is sufficient.

IV. Machine Learning

A. Datasets

For the dataset acquisition, it was considered to explore the system functioning at different distances and in different positions of the antennas:

- Position 1: both antennas placed first at 30 cm from the tags, and then at 40 cm;
- Position 2: both antennas placed at 30 cm from the tags

Five people participated in the creation of the dataset, composed of approximately 450 acquisitions.

For each gesture, EPC, RSSI, and phase of every single tag in the matrix have been acquired. The EPC value is only used to filter and sort the data read from the various tags, while the acquired RSSI and phase values are saved in the

TABLE I.
Total lost tags vs. gesture & antenna

	Gesture					
	0	**1**	**2**	**3**	**4**	**5**
Antenna 1	1	952	1184	1166	913	1655
Antenna 2	1	0	0	2	0	3
Total records	2560	2805	2735	2552	2527	2452

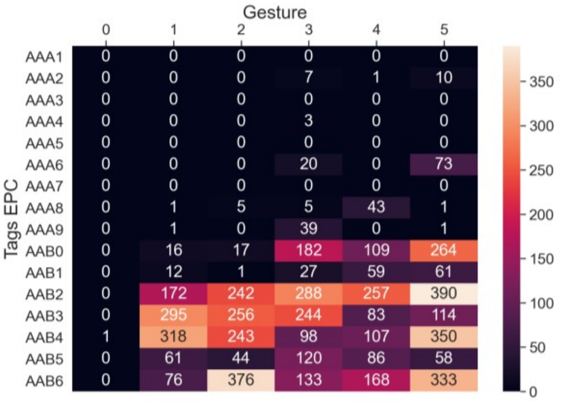

Fig. 2. Lost tags by Antenna 1 vs. gesture.

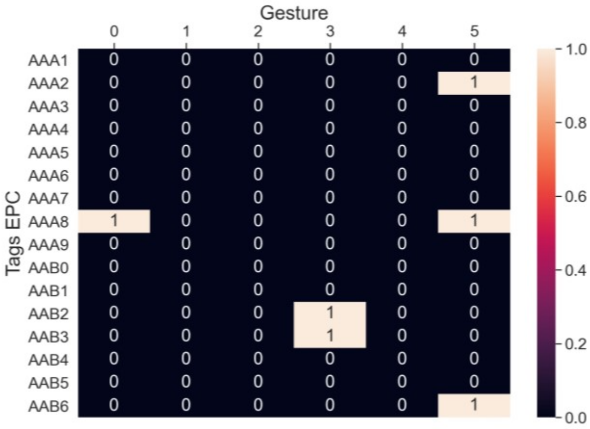

Fig. 3. Lost tags by Antenna 2 vs. gesture.

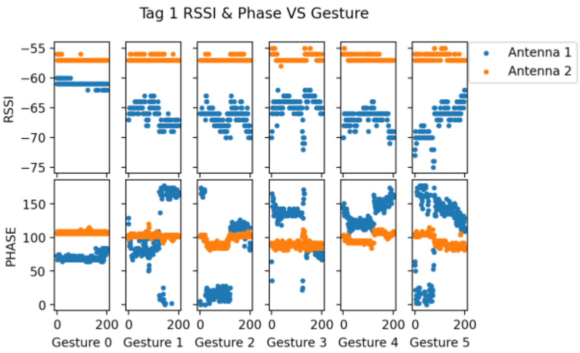

Fig. 4. Tag 1 RSSI and phase vs. gesture.

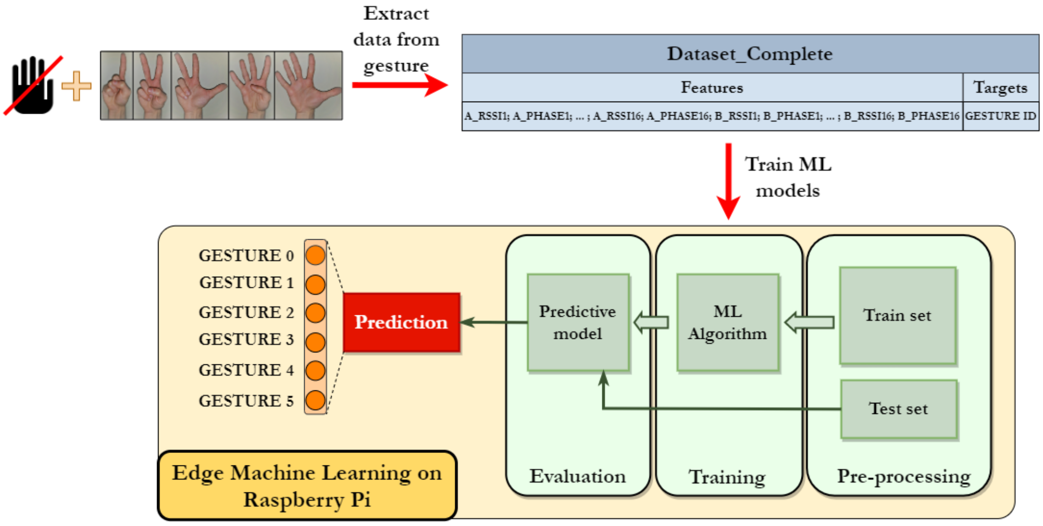

Fig. 5. Block scheme of the experiments.

dataset, along with a numeric label representing the gesture number of each record.

The acquired data results in a dataset that consist of 65 features, including RSSI and phase for each of the 16 tags detected by the two antennas (a total of 64 elements), plus one gesture numerical label. About 2500 records were acquired for each gesture, resulting in a dataset populated by approximately 15000 samples (6 gestures × 2500 records = 15000 samples).

During the process of dataset creation there was a problem of non-acquisition of data from some tags, i.e., when the hand was present between the tag matrix and Antenna 1 the system was unable to identify some tags and consequently to read the related information.

To overcome this problem, we proceeded in the following way:

1) during the creation of the dataset, not a number (NaN) values have been entered in correspondence of the cells relating to the RSSI and phase values of the missing tag information to keep the size of the dataset fixed (64 columns) and paying attention to the ordering of the columns (from tag 1 to tag 16) which, as already mentioned, is based on the tag EPC;
2) subsequently, using the MATLAB code, the NaNs were replaced with the arithmetic mean of RSSI and phase carried out on the column values (for each tag) and in relation to each gesture.

Before replacing the NaN values with the arithmetic mean of the correct readings, we estimated the number of NaN values present in the dataset to estimate how many tags were lost during the acquisitions.

Table I shows the number of tags lost for each gesture and each antenna. The numbers in the last row (Total records) are the readings acquired for each gesture, so for example for Gesture 1 on 2805 acquisitions were found 952 NaN values. Note that the loss of acquisition is relative to the tag, so since for each tag there are two data (RSSI and phase), for each lost tag there are two NaN values, but in Table I we considered only one NaN value for each lost tag since we are analyzing the number of tags lost during the acquisitions. Fig. 2 and 3 give a better look at this analysis. They show the number of readings lost for each tag (number of NaN values), in correspondence with each gesture, and for both antennas. For example, for Gesture 1 out of 2805 acquisitions the NaN values found are: 1 in the AAA8 tag, 1 in the AAA9 tag, 16 in the AAB0 tag, 12 in the AAB1 tag, 172 in the AAB2 tag, 295 in the AAB3 tag, 318 in the AAB4 tag, 61 in the AAB5 tag and 76 in the AAB6 tag; for a total of 952 NaN values. As can be seen, the number of failed tag readings during the polling time is significantly higher for Antenna 1, and this could be due to the fact that the hand between the tag and antenna reduces the backscattered signal amount under the sensitivity threshold, resulting in a missing acquisition from the reader.

Fig. 4 shows the trend of the RSSI and phase values of tag 1 (EPC: AAA1) over 200 readings for each gesture and for both antennas. The figure highlights how data variability is larger in the case of Antenna 1, and this is motivated by the presence of the hand between the tag and Antenna 1.

B. Models

The study in this paper evaluates three different ML models to assess their performance for this specific application. Since the challenge we are solving can be considered as a supervised learning problem, having labeled the acquired samples (the model is trained on the input data knowing the associated output class), and since the model has to categorize a set of input data into different classes (5 hand gestures + 1), what we are facing is, in particular, a multi-class classification problem. To this end, the following models were selected and tested:

1) Random Forest Classifier (RFC): ensemble classifier obtained from the aggregation by bagging of decision trees. The implemented causal forest is made up of 100 decision trees with a maximum depth equal to 14.

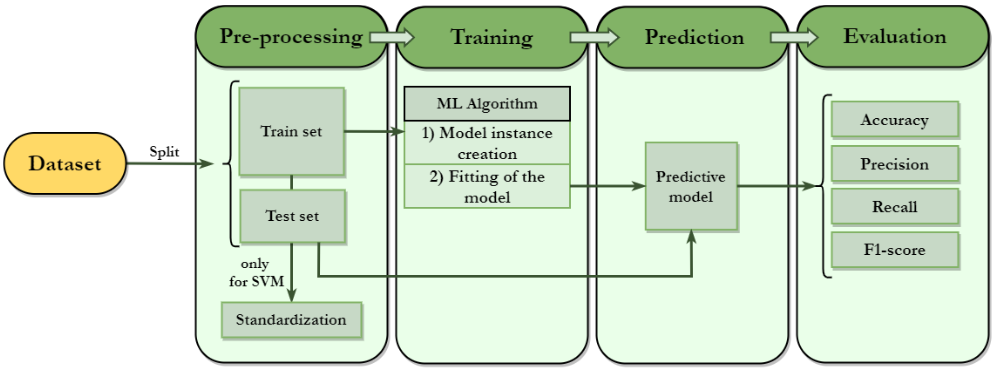

Fig. 6. Block scheme of the Machine Learning (ML) procedure.

2) Support Vector Machine (SVM): has the objective of identifying the hyperplane (decision boundary) that best divides the samples into classes, trying to maximize the margin. The SVM was implemented in a one vs. one mode to perform multiclass classification, using both a polynomial kernel and an RBF (Radial Basis Function) kernel. The kernel method is used to tackle non-linear classification problems. The idea behind kernel methods is to define linear combinations of the original characteristics to project them into a larger space through a mapping function, called the kernel function, where such data becomes linearly separable.

3) Decision Tree Classifier: ML model that uses the tree data structure. The model evaluates the characteristics of the training dataset and learns a series of questions to determine the class labels of the samples. The decision-making algorithm starts from the root of the tree and divides the samples according to the characteristic that produces the greatest information gain (IG), whose goodness is evaluated through a metric called impurity. This subdivision process takes place iteratively on each child node obtained until the leaf nodes (pure nodes) are reached. The implemented decision tree has a maximum depth of 17 and uses the Gini Criterion for calculating impurity.

The choice of these three algorithms was based on several motivations.

A first motivation lies in the nature of the dataset; in fact, having a dataset with a high number of samples (about 15000) and a low number of features (64, low compared to the number of samples) we narrowed the choice to algorithms with low bias and high variance, so that they learn best without underfitting. Algorithms like Decision Tree, Random Forest, Kernel SVM fall into this category.

A second motivation is to concurrently obtain good values of computational speed and accuracy. Decision Tree is a simple algorithm that requires little data preparation (no standardization) and is fast in training (computational cost logarithmic with data size), but with the flaw of being inaccurate, as overly complex (high depth) decision trees do not generalize the data well, which results in overfitting. Another shortcoming is instability, because small changes in the data could generate a completely different tree. These problems are mitigated by the use of Decision Tree within an ensemble (one of the reasons we use RFC). Random Forest and Kernel SVM are more complex algorithms, reason why they provide better performance but longer training times. Random Forest overcomes the problems of maximum depth (irrelevant for RTC) and sensitivity to data variations of Decision Tree since, being an ensemble algorithm, it employs multiple decision trees trained on subsets of the starting dataset and obtains the output (class label) based on a majority vote. The only element of concern is the high number of decision trees it generates, because this, increasing the complexity of the model, makes it slower, albeit more accurate. SVM adapts well to problems with a relative high number of features (as in our case: 64) and, since we have to deal with a nonlinear classification problem, we employ two different kernels: polynomial and RBF, which are among the best known and performing ones. Finally, we included Kernel SVM to evaluate how the performance of the model changes if the number of features is reduced. As shown below, the performance of the two SVM worsen more than the other two models when we employ only 32 features instead of 64. The three considered classifiers fulfill the same function, accepting as input unidimensional vectors whose entries are the RSSI and phase read from the reader, and producing a prediction of the six class labels, corresponding to gestures from 0 to 5, as output.

Fig. 5 shows the block scheme that represents the conducted experiments.

C. Edge Machine Learning

EML Learning is a technique by which IoT devices exploit ML or Deep Learning algorithms to perform data processing locally (using local or at device level processing resources). The goal of this approach is to reduce dependence on Cloud infrastructure both for reasons related to the limitations of the Cloud (high latency, intermittent connectivity, use of IoT constrained devices, etc.) and to satisfy the need for many IoT applications that require processing operations, data transmission and receipt of the result in a very short time (for example applications in the medical and vehicular fields).

For the implemented system, a Raspberry Pi 3 Model B running Xubuntu operating system was used. Python was used as the programming language to implement the ML models presented above.

ML models were imported from the open-source library Scikit-learn [29] and the data within the dataset have been manipulated using NumPy [30] and Pandas libraries [31]. All three model implementations follow the same procedure detailed in Fig. 6:

1) *Preprocessing*

a) Data scaling: we performed a standardization of the data only in the SVM. Through this approach, the feature values were transformed into a zero-mean standard normal distribution with unity standard deviation. This operation was necessary for this model to avoid long training times and low-quality results.

b) Subdivision between training and test dataset: it consists in shuffling the records of the dataset and dividing them into two datasets: one training and one test dataset, respectively. The training dataset is the set of samples from which the model learns, while the test dataset is used in the evaluation phase to assess whether the trained model is actually capable of predicting outputs.

2) *Training*

a) Model instance creation: we imported the model from the Scikit-learn library and created the model instance. Each model has parameters that define it and determine how it will learn during the training process.

b) Fitting of the model: once the instance of the model has been created, it is trained by feeding the training samples set. This is the part of the process that often takes the longest to perform.

c) Evaluation and prediction: after having trained the model, the forecast quality of the data involved is estimated. Different metrics are used to perform this estimation, as presented below.

V. Results

Different metrics have been considered to reliably evaluate the system performance. The performance metrics used to evaluate the models are accuracy, precision, recall, and f1-score. Fig. 7 shows the confusion matrix that summarizes the prediction results of the classification model; in the confusion matrix, each row represents the number of instances in a certain class while each column represents the number of instances expected in a certain class. The matrix provides the results with the adoption of the RFC algorithm with whole dataset.

For the training of the models, we considered three configurations of the dataset:

- complete dataset: considering all the data collected, both from Antenna 1 and Antenna 2;
- Antenna 1 dataset: considering only the data read by Antenna 1;
- Antenna 2 dataset: considering only the data read by Antenna 2.

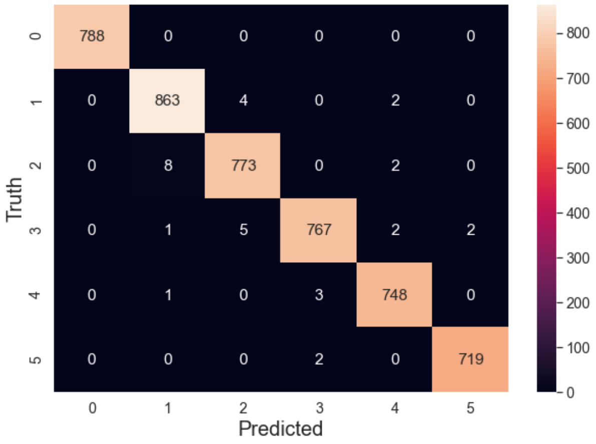

Fig. 7. RFC confusion matrix.

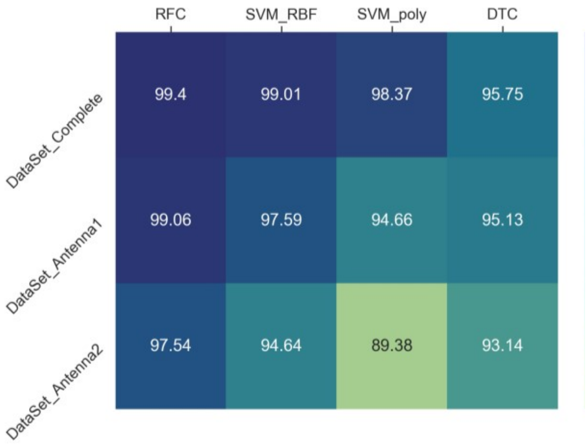

Fig. 8. Accuracy of the different models vs datasets.

TABLE II.

SYSTEM PERFORMANCE WITH WHOLE DATASET

	RFC			SVM Polynomial			SVM RBF			DTC		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Gesture 0	1	1	1	1	1	1	1	1	1	1	1	1
Gesture 1	1	1	1	0.97	0.99	0.98	0.98	0.99	0.98	0.93	0.97	0.95
Gesture 2	0.99	0.99	0.99	0.97	0.97	0.97	0.99	0.98	0.98	0.96	0.92	0.94
Gesture 3	0.99	0.99	0.99	0.98	0.98	0.98	0.99	0.99	0.99	0.95	0.95	0.95
Gesture 4	0.99	1	0.99	0.99	0.98	0.98	0.99	0.99	0.99	0.95	0.95	0.95
Gesture 5	1	0.99	0.99	0.99	0.99	0.99	1	0.99	1	0.96	0.96	0.96
Weighted Avg	**0.99**	**0.99**	**0.99**	**0.98**	**0.98**	**0.98**	**0.99**	**0.99**	**1**	**0.96**	**0.96**	**0.96**
Accuracy	**99.40 %**			**98.37 %**			**99.01 %**			**95.75 %**		

TABLE III.
SYSTEM PERFORMANCE WITH ANTENNA 1 DATASET

	RFC			SVM Polynomial			SVM RBF			DTC		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Gesture 0	1	1	1	1	1	1	1	1	1	1	1	1
Gesture 1	0.99	0.98	0.99	0.93	0.93	0.93	0.97	0.97	0.97	0.94	0.95	0.95
Gesture 2	0.98	0.99	0.98	0.90	0.94	0.92	0.95	0.96	0.96	0.93	0.95	0.94
Gesture 3	1	0.99	0.99	0.94	0.93	0.93	0.97	0.96	0.96	0.94	0.93	0.94
Gesture 4	0.98	0.99	0.99	0.95	0.93	0.94	0.99	0.97	0.98	0.95	0.93	0.94
Gesture 5	0.99	1	0.99	0.96	0.95	0.96	0.98	0.99	0.98	0.95	0.95	0.95
Weighted Avg	**0.99**	**0.99**	**0.99**	**0.95**	**0.95**	**0.95**	**0.98**	**0.98**	**0.98**	**0.95**	**0.95**	**0.95**
Accuracy	**99.06%**			**94.66 %**			**97.59 %**			**95.13 %**		

TABLE IV.
SYSTEM PERFORMANCE WITH ANTENNA 2 DATASET

	RFC			SVM Polynomial			SVM RBF			DTC		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Gesture 0	1	1	1	0.99	1	1	1	1	1	1	1	1
Gesture 1	0.97	0.99	0.98	0.88	0.96	0.92	0.96	0.97	0.96	0.95	0.95	0.95
Gesture 2	0.96	0.96	0.96	0.85	0.84	0.85	0.91	0.94	0.92	0.93	0.92	0.92
Gesture 3	0.98	0.94	0.96	0.88	0.86	0.87	0.94	0.90	0.92	0.91	0.89	0.90
Gesture 4	0.98	0.97	0.97	0.86	0.84	0.85	0.91	0.93	0.92	0.89	0.92	0.91
Gesture 5	0.97	0.98	0.98	0.92	0.86	0.89	0.96	0.94	0.95	0.92	0.91	0.92
Weighted Avg	**0.98**	**0.98**	**0.98**	**0.89**	**0.89**	**0.89**	**0.95**	**0.95**	**0.95**	**0.93**	**0.93**	**0.93**
Accuracy	**97.54 %**			**89.38 %**			**94.64 %**			**93.14 %**		

Table II outlines the performance of the hand gesture recognition system for each of the classifiers trained on the entire dataset, according to the aforementioned metrics. Table III reports the performance on the dataset related to Antenna 1, while Table IV reports the performance of the system trained on the dataset related to Antenna 2.

Fig. 8 reports the accuracy values of the different models trained on the three datasets considered.

Discussion

RFC performs best on all metrics, with an accuracy that reaches 99.40%. SVM has very similar performances to RFC although its accuracy notably drops when it considers only the dataset with a single antenna, especially with the Antenna 2 dataset. Furthermore, an RBF kernel seems to be more suitable than a polynomial one for the classification problem addressed.

Decision Tree Classifier achieves the worst performance albeit the accuracy is still higher than 93%. However, it must be taken into account that DTC is the basis of the RFC; the fact that a single tree shows high performance explains how RFC, which uses multiple decision trees, has the best performance. As it can be seen, the metric values drop when the three models are trained on a single antenna dataset and this can be explained by the fact that part of the context information is missing, compared to the complete dataset, which is therefore much richer in information.

Furthermore, between the two single-antenna datasets, the one relating to Antenna 1 offers the best performance since the data it contains has greater variability, as highlighted in Fig. 2. However, the results with Antenna 2 are still positive, and give an accuracy of 97.54 % with the RFC algorithm. The use of Antenna 2 only, in the configuration with the tags positioned between the antenna and the hand, can potentially give a great advantage from the point of view of ergonomics and practical applications of the device. In fact, with this arrangement, the user could simply place the hand in front of the surface containing the tags, and not in the space between the antenna and the tags.

TABLE V.
SYSTEM COMPARISON WITH SoTA

	Accuracy	Way of use	Range / Covered area	Number of gestures	Number of tags	Platform	Acquisition rate
[11]	98%	Tags attached on glove – 1 antenna	80 cm	5	5	Desktop	500 ms
[12]	NA	Tags attached to each object. 8 antennas and a large number of various sensors and effectors.	9 m^2	13	2 - 4	Desktop	20 ms
[13]	99.9% (static gestures) 94.8% (dynamic gestures)	Tags not attached, 49 tags, 1 antenna	10 cm	26 (static gestures) +10 (dynamic gestures)	49	Desktop	NA
[14]	NA	Tags attached to the object	120/60 cm	NA	From 2 to 6 tags for movement gesture and 1 tag for touch gesture	Desktop	NA
[15]	96.2%	One tag attached to each hand	Antenna fixed at the ceiling of a room (2.2m high)	8	2 tags (one per hand)	Desktop	NA
[16]	100 %	One antenna for each tag	Not more than 3-4cm	5	4	Desktop	200 ms
This work	99%	Tags not attached, 16 tags, 1-2 antenna(s)	30 – 40 cm	6	16	Edge, Machine Learning	300 ms

Antenna 2 could also be placed in a position close to the tags, making the system much more compact.

Table V provides a comparison with similar systems found in the literature. As per our knowledge, this is the first time that a compact solution able to perform gesture recognition using such a low number of tags with the provided accuracy and a resource-constrained inference engine is presented. In [2] a vision-based hand gesture interface is introduced, and the average accuracy of recognition in the experiment is 0.938. However, the results show the performance depends on the SNR, worsening when the recognition is performed under cluttered background vs. simple background. Regarding the power consumption, the rated power of most cameras varies from 2 watt to 15 watt. Similarly, applications that use Time-of-Flight cameras [7] show a power consumption of about 2.5 W. In the proposed solution, the active part in the process of hand-gesture recognition is the RFID reader, an RFID ThingMagic M6e Micro UHF RAIN which exhibit a nominal DC Power consumption of 5.5 W in active mode (RF on, +30 dBm), 0.325 W when it is not transmitting and 0.06 W in standby mode. In the case the system acquires every second within which 300 ms are used for the single reading (RF on) and the remaining 700 ms remains in "no transmitting" mode, then the average power consumption within that second is 1.87 W @ 1 Hz. Duty-cycled operations allow to reduce the power consumption up to 0.241 W with a reading rate of 0.1 Hz.

The change in luminance or brightness of the environment does not affect the SNR, envisioning the usage of the system in not-controlled environment also under direct sunlight, which is an important pitfall of the vision-based solutions. Real-life applications of the system can be envisioned in the industrial context to improve the safety and privacy of machinery operators; in fact, the current technologies in the industrial sector for monitoring activities are based on the use of cameras that can violate the privacy of the operators. Furthermore, the sensorless nature of the system allows operators not to have to wear suitable clothing or tools, therefore facilitating the work.

It could also be thought of using the system to carry out touch-less authorization and control operations, for example entering a PIN code to access the bank account in an ATM (the system would replace the classic numeric keypads), with the advantage of not having to touch objects used by multiple users. Contactless operations can also cover gaming and entertainment, such as a simple remote control system for electrical equipment.

Another envisioned application could be a text-to-voice system for communication with deaf people.

VI. CONCLUSIONS

This work proposes a novel hand gesture recognition system that allows distinguishing five different gestures making use of ML models, deployed to an edge device. Unlike existing approaches, the designed solution does not involve the use of special signals or wearable sensors. The system consists of 16 passive RFID tags, and an RFID reader with two antennas, in front and behind the tags matrix. Placing a hand between Antenna 1 and the tag cluster causes an interference with the RF signal radiated from both antennas, resulting in variations in RSSI and phase values. A greater variability was observed in the values read by Antenna 1 compared to those read by Antenna 2. Detecting the phase and RSSI values, the system is able to distinguish one gesture from another.

Exploiting the presented system, a training dataset was built for training ML models to recognize five hand gestures with the addition of a gesture 0 related to the case of absence of the hand. During the creation of the training dataset, it was observed that the use of a relatively large number of tags can lead, in the presence of the hand between the antenna and the tag, to difficulties in reading some of them. Such difficulties lead to the consequent need to fill the gap left empty. It has been found that the *a posteriori* calculation of the arithmetic mean of RSSI and phase carried out on the column (for each tag) and in relation to each gesture can validly replace the missing data, thus not affecting the accuracy of the models trained.

EML techniques have been used for training and running the considered models (RFC, SVM with polynomial and RBF kernel, and DTC) on a Raspberry Pi3 model B The performance of these different classification algorithms has been evaluated using different metrics providing an accuracy of gesture recognition up to 99.4 %. When only the dataset with a single antenna is considered, the accuracy is slightly reduced, especially with the Antenna 2 dataset. However, this configuration, with only an antenna put behind the tags, still allows for more than 97% accuracy and is an attractive option that can eliminate the need for the user to put the hand in the space region between the antenna and the tag array, paving the way for a sort of virtual touch-less input device.

References

[1] M. Di Foggia and D. M. D'Addona, "Identification of critical key parameters and their impact to zero-defect manufacturing in the investment casting process," *Procedia CIRP*, vol. 12, pp. 264–269, 2013, doi: 10.1016/j.procir.2013.09.046.

[2] F. Yikai, W. Kongqiao, C. Jian, and L. Hanqing, "A real-time hand gesture recognition method," in *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME 2007*, 2007, pp. 995–998. doi: 10.1109/icme.2007.4284820.

[3] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015, doi: 10.1007/s10462-012-9356-9.

[4] O. Patsadu, C. Nukoolkit, and B. Watanapa, "Human gesture recognition using Kinect camera," *JCSSE 2012 - 9th International Joint Conference on Computer Science and Software Engineering*, pp. 28–32, 2012, doi: 10.1109/JCSSE.2012.6261920.

[5] R. Carotenuto, M. Merenda, D. Iero, and F. G. Della Corte, "An Indoor Ultrasonic System for Autonomous 3-D Positioning," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 7, 2019, doi: 10.1109/TIM.2018.2866358.

[6] R. Carotenuto, M. Merenda, D. Iero, and F. G. Della Corte, "Mobile Synchronization Recovery for Ultrasonic Indoor Positioning," *Sensors*, Jan. 2020, doi: 10.3390/s20030702.

[7] T. Kapuściński, M. Oszust, and M. Wysocki, "Hand gesture recognition using time-of-flight camera and viewpoint feature histogram," *Advances in Intelligent Systems and Computing*, vol. 230, pp. 403–414, 2014, doi: 10.1007/978-3-642-39881-0_34.

[8] R. Carotenuto, M. Merenda, D. Iero, and F. G. Della Corte, "Ranging RFID tags with ultrasound," *IEEE Sensors Journal*, pp. 1–1, 2018, doi: 10.1109/JSEN.2018.2806564.

[9] J. Wang and D. Katabi, "Dude, where's my card? RFID positioning that works with multipath and non-line of sight," *Computer Communication Review*, vol. 43, no. 4, pp. 51–62, 2013, doi: 10.1145/2534169.2486029.

[10] W. Ruan, L. Yao, Q. Z. Sheng, N. J. G. Falkner, and X. Li, "TagTrack: Device-free localization and tracking using passive RFID tags," *MobiQuitous 2014 - 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pp. 80–89, 2014, doi: 10.4108/icst.mobiquitous.2014.258004.

[11] S. N. R. Kantareddy, Y. Sun, R. Bhattacharyya, and S. E. Sarma, "Learning gestures using a passive data-glove with RFID tags," in *2019 IEEE International Conference on RFID Technology and Applications, RFID-TA 2019*, 2019, pp. 327–332. doi: 10.1109/RFID-TA.2019.8892224.

[12] K. Bouchard, S. Giroux, B. Bouchard, and A. Bouzouane, "Regression analysis for gesture recognition using passive RFID technology in smart home environments," *International Journal of Smart Home*, vol. 8, no. 5, pp. 245–260, 2014, doi: 10.14257/ijsh.2014.8.5.22.

[13] H. Ding *et al.*, "RFnet: Automatic Gesture Recognition and Human Identification Using Time Series RFID Signals," *Mobile Networks and Applications*, vol. 25, no. 6, pp. 2240–2253, Dec. 2020, doi: 10.1007/s11036-020-01659-4.

[14] Y. Bu *et al.*, "RF-Dial: Rigid Motion Tracking and Touch Gesture Detection for Interaction via RFID Tags," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020, doi: 10.1109/tmc.2020.3017721.

[15] B. Chen, Q. Zhang, R. Zhao, D. Li, and D. Wang, "SGRS: A sequential gesture recognition system using COTS RFID," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2018, vol. 2018-April, pp. 1–6. doi: 10.1109/WCNC.2018.8376998.

[16] R. Parada, K. Nur, J. Melia-Segui, and R. Pous, "Smart surface: RFID-based gesture recognition using k-means algorithm," in *Proceedings - 12th International Conference on Intelligent Environments, IE 2016*, 2016, pp. 111–118. doi: 10.1109/IE.2016.25.

[17] D. De Donno, L. Catarinucci, and L. Tarricone, "RAMSES: RFID augmented module for smart environmental sensing," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 7, pp. 1701–1708, 2014, doi: 10.1109/TIM.2014.2298692.

[18] M. Merenda *et al.*, "Performance assessment of an enhanced RFID sensor tag for long-run sensing applications," in *Proceedings of IEEE Sensors*, 2014, vol. 2014-December, no. December, pp. 738–741. doi: 10.1109/ICSENS.2014.6985105.

[19] M. S. Khan, M. S. Islam, and H. Deng, "Design of a reconfigurable RFID sensing tag as a generic sensing platform toward the future Internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 300–310, 2014, doi: 10.1109/JIOT.2014.2329189.

[20] M. Merenda, D. Iero, and F. G. D. Corte, "Cmos rf transmitters with on-chip antenna for passive RFID and iot nodes," *Electronics (Switzerland)*, vol. 8, no. 12, 2019, doi: 10.3390/electronics8121448.

[21] M. Merenda, C. Felini, and F. G. Della Corte, "A monolithic multisensor microchip with complete on-chip RF front-end," *Sensors (Switzerland)*, vol. 18, no. 1, 2018, doi: 10.3390/s18010110.

[22] I. Farris, S. Pizzi, M. Merenda, A. Molinaro, R. Carotenuto, and A. Iera, "6lo-RFID: A framework for full integration of smart UHF RFID tags into the internet of things," *IEEE Network*, vol. 31, no. 5, 2017, doi: 10.1109/MNET.2017.1600269.

[23] M. Merenda, C. Porcaro, and D. Iero, "Edge Machine Learning for AI-enabled IoT devices: a review," *Sensors (Switzerland)*, 2020.

[24] M. Merenda, G. Cimino, R. Carotenuto, F. G. Della Corte, and D. Iero, "Device-free hand gesture recognition exploiting machine learning applied to RFID," *2021 6th International Conference on Smart and Sustainable Technologies, SpliTech 2021*, 2021, doi: 10.23919/SpliTech52315.2021.9566385.

[25] "Impinji Monza 5 chip datasheet." https://support.impinj.com/hc/article_attachments/203268870/Monza%205%20Tag%20Chip%20Datasheet%20R3%2020160823.pdf (accessed Jan. 31, 2022).

[26] Jadak, "ThingMagic UHF RAIN RFID Module Series," 2015.

[27] LAIRD, "S865 Series RFID Panel Antenna."

[28] G. Çaliş, B. Becerik-Gerber, A. B. Göktepe, S. Li, and N. Li, "Analysis of the variability of RSSI values for active RFID-based indoor applications," *Turkish Journal of Engineering and Environmental Sciences*, vol. 37, no. 2, pp. 186–210, 2013, doi: 10.3906/muh-1208-3.

[29] "scikit-learn." https://scikit-learn.org/stable/about.html (accessed Jan. 31, 2022).

[30] "Numply Library." https://numpy.org/citing-numpy/ (accessed Jan. 31, 2022).

[31] "Pandas libraries." https://pandas.pydata.org/about/citing.html (accessed Jan. 31, 2022).