U-WeAr: User Recognition on Wearable Devices through Arm Gesture

Simone Bianco, Paolo Napoletano, Alberto Raimondi, and Mirko Rima

***Abstract*—The use of wearable devices equipped with inertial sensors has become increasingly pervasive. It has been widely demonstrated in the literature that inertial signals acquired by these sensors can be used by machine learning algorithms to predict actions performed and/or to recognize the identities of the person wearing the sensors. In this article, we present a hardware/software system for arm gesture recognition, identity recognition, and verification of a person based on inertial sensors. The hardware part is a custom wristband that consists of a computing unit, a wireless communication unit, and an inertial sensor. The software part is an algorithm based on recurrent neural networks that is able to process the signals coming from the sensor and to return a prediction. To validate the system, a dataset consisting of 25 symbols drawn with the arm is collected. These symbols are performed by 33 subjects. We conduct two evaluations: 1) performance evaluation for arm gesture recognition, user recognition and verification; and 2) usability assessment of the system. The performance of the three recognition tasks indicate that this system can be reliably applied in real environments with an accuracy above 96% for gesture recognition, an accuracy of about 85% for user identification, and an equal error rate of about 13% for user verification. The outcome of the usability test proves a great satisfaction from the users in terms of high simplicity in the use of the wristband and goodness of the machine learning predictions.**

***Index Terms*—Arm gesture, arm gesture database, device, gesture recognition, inertial sensors, user identification, user verification.**

I. Introduction

NOWADAYS, inertial sensors like accelerometers and gyroscopes are present in many consumer devices, such as smartphones, smartwatches, and fitness bands. Inertial sensors can be exploited by machine learning algorithms to perform fine and coarse-grained human action recognition with a high level of accuracy [1]–[3] and to identify users performing a gesture [4]. Inertial signals, in contrast with visual signals taken with video cameras, preserve the privacy of the user thus being a very good candidate for authentication on mobile devices [3].

Over the last decade, there has been a radical change in human–machine interaction devices and technologies. Human gesture or body pose recognition algorithms are largely used in various areas, such as sign language recognition [5], robotics [6], game technology [7], immersive education [8], medicine [9], and many others. New technologies improve everyday life [10], but they also create unprecedented security issues. User identification is needed to prevent violation of restricted physical areas, mobile devices, etc [11]. Authentication processes can be performed in different ways: typing in login credentials, digital fingerprints recognition, speech recognition, face recognition, handwriting recognition, DNA recognition, recognition using biometric data, recognition using physiological signals, recognition using inertial signals, etc. Some of these ways are more robust than others to attempts at fraud [12]. For instance, the security of a facial recognition-based system can be threatened if a picture of the user is used instead of a live recording. In contrast, systems based on physiological or inertial signals are more difficult to threaten because it is more difficult to reliably emulate these signals.

Driven by the need to create high-security devices for smart interaction, in this article we present a hardware/software system for arm gesture recognition, identity recognition, and verification of a person based on inertial sensors: a custom wristband with a computing unit, a wireless communication unit, and an inertial sensor makes up the hardware; the software component is a recurrent neural networks (RNNs)-based algorithm that runs on the server and and processes sensor signals returning a prediction. A preprocessing procedure is introduced in order to make the system invariant with respect to the magnitude in time and space of the performed gesture. We evaluate three different recognition tasks: arm gesture recognition, user identification, and user verification. To evaluate the performance we use a dataset designed and collected by us. The dataset consists of 25 symbols drawn with the arm by 33 subjects. Symbols comprise numbers, letters, and actions. Performance of the three recognition tasks as well as the outcome of the usability test are very satisfactory and this indicates that this system can be reliably applied in a real environment with great satisfaction from the users.

To sum up, the main contributions of this article are as follows.

1) We propose a novel hardware/software system for arm gesture recognition, user identification, and verification using inertial signals.
2) We propose the use of a preprocessing technique to make the system invariant with respect to the magnitude in time and space of the performed gesture.

Manuscript received March 22, 2021; revised October 15, 2021 and December 29, 2021; accepted March 26, 2022. Date of publication May 13, 2022; date of current version July 14, 2022. This article was recommended by Associate Editor Giancarlo Fortino. *(Corresponding author: Paolo Napoletano.)*

The authors are with the Department of Informatics, Systems and Communication, University of Milan-Bicocca, 20126 Milano, Italy (e-mail: simone.bianco@unimib.it; paolo.napoletano@unimib.it; a.raimondi21@campus.unimib.it; m.rima@campus.unimib.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/THMS.2022.3170829.

Digital Object Identifier 10.1109/THMS.2022.3170829

3) We propose a new dataset of inertial data including recordings of 33 subjects performing 25 gestures.
4) The proposed dataset, contrary to what happens in the literature, contains the rejection class.

The rest of this article is organized as follows. Section II resumes the main works in the state of the art related to this research, Section III describes the hardware and software components of the system in detail, while Section IV presents the recognition methodologies. Section V presents the experiment results. Finally, Section VI concludes this article.

II. Related Work

The literature that has been analyzed is divided into two main topics: 1) arm gesture recognition and 2) user identification.

A. Arm Gesture Recognition Using Inertial Sensors

Recognition of human physical activity as well as recognition of multiuser activity is quite relevant in the field of body sensor network (BSN) [13], [14]. Wireless BSN is a collection of wearable sensor nodes with computational and storage capacity that communicate with a local personal device, such as a smartphone, tablet, etc. [15]. BSN are largely used for human behavior monitoring.

Due to the high development of mobile systems equipped with inertial sensors, arm gesture recognition is a topic largely investigated in the literature in recent years.

Hofmann *et al.* [16] proposed a recognition scheme based on hidden Markov models (HMM), using a discrete HMM (dHMM) to recognize dynamic gestures recorded by a triaxial accelerometer. The experiments were conducted using 500 gestures with 10 repetitions per gesture for training and 100 gestures for testing. The overall accuracy reached is 95.6%. Kallio *et al.* [17] using an HMM model to classify 8 gestures with an accuracy of 96.1%. Pylvänäinen *et al.* [18] obtained reliable results, with 96.67% of classification on a database of 20 repetitions for 10 gestures, while with a multistream HMM with a dataset of 18 gestures (10 repetitions each), the accuracy of the average recognition is was about 91.7%. In addition to HMM, Wu *et al.* [19] employed the multiclass support vector machine (SVM) for user-independent gesture recognition, demonstrating that SVMs significantly outperformed other methods including HMM. Indeed it is achieved a classification accuracy of 93%, for the recognition of seventeen complex gestures.

The advance of deep learning then led the research toward the use of approaches for the recognition of gestures through a RNN. Shin *et al.* [20] trained an RNN model for gesture recognition using the SmartWatch gesture dataset, where each sequence of gestures contained acceleration data (recorded by a 3-D accelerometer). Lefebvre [21] conducted experiments for the recognition of gestures on a dataset of 14 gestures, consisting of both accelerometer and gyroscope sensors and this led to a great improvement in performance. Furthermore, the BLSTM-based method achieved 95.57% accuracy on the database of 1540 total gestures. Yuanhao Wu *et al.* [22] had the goal of recognizing the continuous natural gestures of 12 action classes consisting of 1615 instances of gestures, recorded by 3 IMU sensors and obtained from 10 subjects. The experiment achieved an accuracy of 91.54%. Chunyu Xie *et al.* [23] obtained an accuracy rate of 97.4% to classify 20 mobiles gesture class from a database consisting of 3200 samples. Ce Li *et al.* [2] obtaining an accuracy rate of 99.15% to classify 12 class gestures of numbers and alphabet letters from a database consisting of 5547 samples.

From the analysis of the state of the art it seems quite clear that methods based on RNNs are the best, which is why the methodologies tested in this work are based on this technique.

B. User Identification and Verification

User identification and/or verification is required in many common activities, such as mobile phone and door unlooking, bank payement, etc. Different techniques are available to recognize the identity of the subjects [24]. In 2006, Clarke and Furnell [25] proposed keystroke analysis for user authentication using mobile devices. Gait is a biometric trait that can allow user authentication, it can be obtained by using inertial sensors (accelerometer, gyroscope). Nickel *et al.* [26] introduced a method for extracting gait features to be classified using k-nearest neighbor (k-NN) algorithm in mobile devices. Marsico *et al.* [27] using a DTW algorithm, k-NN and other methods to recognize different users by capturing the dynamics of the walking pattern. Dynamic time warping (DTW) algorithm achieves an EER of 20% using a dataset with 51 subjects, each with two roundtrip sessions and an EER of 5.7% on a dataset of 60 subjects with 12 gait acquisitions each, collected in two different days. Liu *et al.* [28] presented a gesture recognition technique based on an accelerometer, DTW and template adaptation to define user custom gestures. This technique was used to recognize the act of answering or placing a phone call by different users. This method has been improved by Conti *et al.* [29], which propose two variations of DTW algorithm, DTW distance (DTW-D), and DTW similarity, that are applied to accelerometer and gyroscopes data for the authentication of 10 subjects. Abate *et al.* [30] created a multibiometric system based on the observation that the instinctive gesture of responding to a phone call can be used to capture two different biometrics, namely, ear and arm gesture. The data are collected by accelerometer, gyroscope, and camera and they used various techniques of feature extraction, features matching, and data-fusion. They acquired a new dataset called "ear-arm database," including data capture from more than 100 subjects performed during different sessions. According to the best identification result the EER value for the combined ear-arm is 10% and for the single-arm gesture is 13%.

III. Proposed System

The proposed system for arm gesture recognition, identity recognition, and verification of a person based on inertial sensors is a hardware/software system. Each component of the system will be described in the following.

A. Hardware Components

The hardware part is composed of a custom wristband and a server.

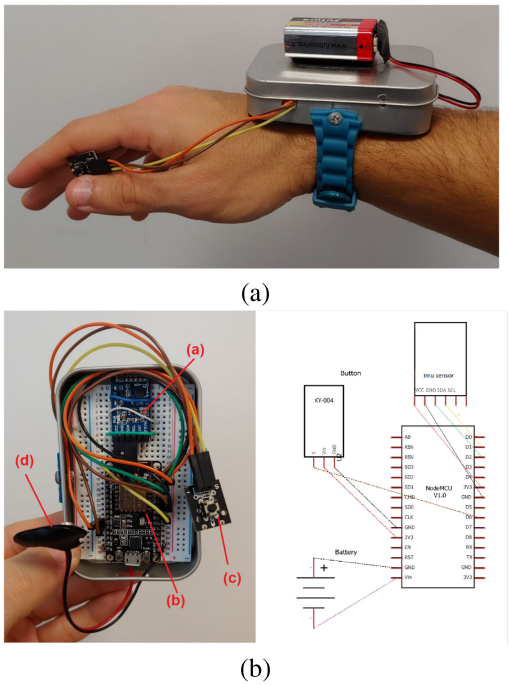


Fig. 1. (a) External and (b) internal view with circuit connections of the device.

The wristband consists of a computing unit, a wireless communication unit, and an inertial sensor. The prototype wristband is shown in Fig. 1(a) and it is composed by the following components.

1) Inertial measurement unit (IMU) sensor, including a 3-axis accelerometer and 3-axis gyroscope.
2) NodeMCU microcontroller, equipped with the ESP8266 chip.
3) Activation button (KY-004 module [31]).
4) External power source.

All components are shown in Fig. 1(b).

The sensor chosen for the prototype is the version 2 of an IMU sensor 10 degrees of freedom (DOF) produced by Waveshare [32], comprising the MPU-9250 chip (3-axes accelerometer, 3-axes gyroscope, 3-axes magnetometer) and the BMP-280 chip for the barometer. In this work only the 6 degrees of accelerometer (x, y, z) and gyroscope (x, y, z) are used. The magnetometer is not considered because we have chosen to have an initial position not limited by height, position in space or by the inclination of the arm. In order to make the system independent on the starting 3-D point of the gesture, the initial reference points of the axes are calculated each time. The server is a workstation equipped with Ubuntu 18.04. The hardware specifications of the computer are the following: 16 GB RAM, and i7-7700 CPU with a clock speed of 3.60 GHz.

The pins for connecting the various components to the microcontroller are shown in the Fig. 1(b).

B. Software Components

The software components are separated in wristband-side and server-side components.

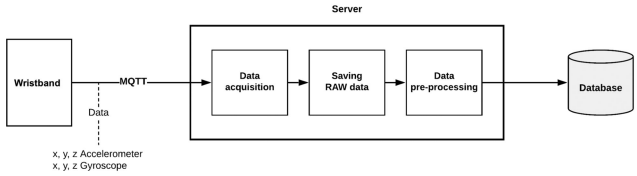


Fig. 2. Schematic representation of the Wristband and Server interaction during data collection. A wristband equipped with a IMU is used for data collecting. Data are sent through the MQTT protocol from the wristband to a server. On server side, a data buffer is used until the gesture performed by the user is finished. Then, raw data are saved and after a preprocessing step data are also saved into a database.

1) Wristband-Side: The wristband is programmed with the Arduino IDE environment. The sketch running on the Node MCU cycles three operations: a) activation button status reading; b) IMU values reading; and c) IMU values communication to the server.

To acquire and interact with the IMU we use the *bolderflight/MPU9250* library [33]. This library permits to modify and control all the parameters with a fine granularity, thus maximizing customization in the use of the sensor. The library settings for the acquisition of accelerometer and gyroscope information are the following.

i) Accelerometer full scale range: $\pm$ 2G.
ii) Gyroscope full scale range: $\pm$ 2000 deg/s.
iii) Digital low pass filter (DLPF) bandwidth: 41 Hz.
iv) Data output rate (sampling rate): 200 Hz.

with the following units of measurement.

v) Accelerometer: m/s^2 (meters per second squared).
vi) Gyroscope: deg/s (degrees per second).

After the assembly of the wristband, in order to record reliable data, we verify that the detection by the IMU sensor is adequate. The correctness of the accelerometer and gyroscope is analyzed in detail, verifying that the rotations are in line with expectations.

The IMU data acquired by the sensor is then sent to the server through the MQ telemetry transport or message queue telemetry transport (MQTT) protocol [34].

2) Server-Side: A Python-based application is running on the server. This application is listening on the MQTT topic used by the wristband to transmit the IMU sensor data to the server. Once the data from the wristband is received, then data are kept in a buffer until the user has his hand steady for a given time. The data are saved in raw format and after a preprocessing step it is saved to the final database. The interaction between the wristband and the server is reported in Fig. 2.

a) Preprocessing: The inertial signals associated to each gesture may have a different spatial and temporal size, and magnitude values. For this reason, a preprocessing procedure is applied to conform the data for each gesture to a standard length and magnitude. We also applied noise reduction to the recorded signal. Summing up, the preprocessing consists of the following three steps.

i) A low pass filter is applied to the accelerometer and gyroscope data of a single gesture for noise reduction.

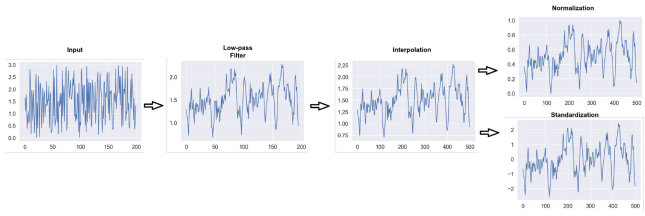


Fig. 3. Example of the signal preprocessing. From left to right: RAW data, signal after applying the low pass filter, signal after interpolation to the standard length, final signal after normalization (used for Gesture recognition), final signal after standardization (used for User identification).

ii) Recorded data are subsampled or upsampled to a standard length of 500 samples, which is equivalent to a recording of length 2.5 s considering 200 samples per second.
iii) Data is then normalized. The normalization strategy depends on the recognition task. In the case of gesture recognition, we constrained each recording gesture to a range between 0 and 1. In the case of user identification, we applied the Z-score normalization [35], [36].

The preprocessing step provides significant benefits to the models, allowing faster convergence and higher accuracy in the classification of gestures. The nonfixed time of the performed gestures gives the possibility of not having a time limit in their execution. In addition, no spatial limits are given in the gesture thanks to the normalization step. The possibility of being able to change the stride and speed of a gesture makes the model flexible, with fewer constraints. This allows an increase in the accuracy in the classification of gestures. Moreover the interpolation of the datapoints reducing each gesture to a specific length allows higher parallelization during training and simplifies the batching operation necessary for training big models. The final data consist in a 500 elements sequence for each gesture, where each element of the sequence consists in a 6-D vector encoding the information about the 6 DOF of the sensor. Fig. 3 shows the preprocesing operation on a recorded gesture.

b) Recognition module: The vector of dimensions 500×6 is the input of the recognition module. The recognition methodologies proposed in this article will be discussed in detail in Section IV.

C. Data Collection

To validate the system, a dataset of several gestures performed by several subjects is acquired. Taking inspiration from the work by Li *et al.* [2] and the work by Wu *et al.* [22] we design a novel gesture vocabulary consisting of 25 classes, which includes gestures representing numbers, letters, actions, and a rejection class. The gesture vocabulary is reported in Fig. 4. The rejection class contains examples of gestures composed by random movements, e.g., the ones generated with a dangling arm. The introduction of the rejection class permits to intercept all the movements that are not valid gestures thus avoiding the unintentional execution of the associated command.

The dataset includes gestures registered by 33 volunteers (21 males, 12 females) wearing an IMU sensor, placed in the aforementioned custom wristband. Each subject performed the

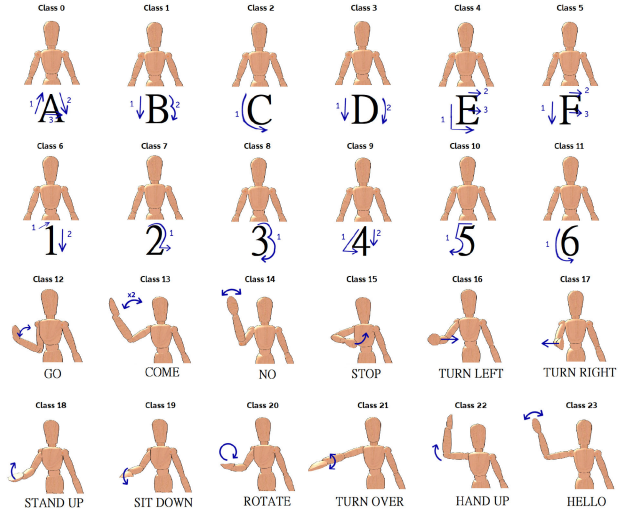


Fig. 4. Vocabulary of the 24 gestures considered. It comprises letters (from class 0 to class 5), numbers (from class 6 to class 11), actions (from class 12 to class 23), and a rejection class (class 24, not shown in the figure).

required gesture following the path as reported in Fig. 4. For instance, to draw the symbol "A," the subject first drew the segment number 1, then the segment number 2, and finally the segment number 3.

The IMU sensor records the triaxial accelerometer at a frequency of 200 Hz. No requirements in terms of temporal duration of the gesture execution are given to the subjects. This means that each gesture class may have a different duration. This is one of the novelties introduced in this work with respect to the state of art, where each subject executed gestures of the same temporal duration. Each user performs 5 recordings per class, producing 125 samples. The final dataset is composed of 4125 arm gestures.

The dataset is split in a training set and a test set used for evaluation with a split of respectively 3300 and 825 samples. In the following sections we will refer to the 500 dimensional sequence of 6-D sensor readings as a gesture when talking about the inputs to the models. The collected dataset and the splits used in this work are made freely available for research purposes.[1]

IV. Recognition Methodology

In this section we discuss the recognition methodologies we adopt for the following recognition tasks.

a) Gesture Recognition: The goal of this first task is to recognize the input gesture into one of the 25 classes of the acquired dataset.

b) User Identification: The aim of the second experiment is the recognition of the identity of the user that performed one of the gestures taken from the vocabulary. The number of users to be recognized is 33.

c) User Authentication: The third experiment is focused on the task of user authentication, meaning the ability to verify the identity of a user attempting to use the wristband. The experiment is performed in two different configurations as follows.

[1][Online]. Available: http://www.ivl.disco.unimib.it/activities/u-wear/

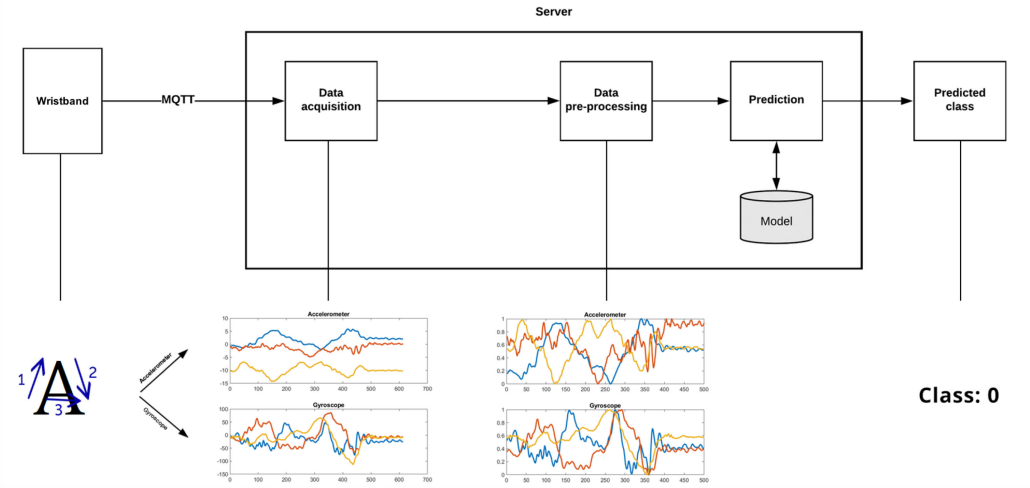


Fig. 5. Schematic representation of the pipeline used for gesture recognition, from gesture acquisition to gesture class prediction.

1) Gesture independent, where the aim is to verify the identity of a user whatever is the arm gesture performed. For example, being able to verify the identity of user *X* performing the gesture *A* as well as the gesture *B*, *C*, etc.
2) Gesture dependent, where the aim is to verify the identity of a user performing a given arm gesture. For example, being able to verify the identity of user *X* performing the gesture *A* and not *B*, *C*, etc.

The whole recognition pipeline from gesture acquisition to gesture recognition is reported in Fig. 5. Without loss of generality we just report the recognition pipeline for the gesture recognition task, since the other two can be easily generated from it.

A. Models

The technique proposed is inspired by [2] and two different RNN models are used for the task. The two models differ in the type of recurrent block used: the first model uses LSTM cells [37], while the second uses GRU cells [38]. LSTMs and GRUs are two variant of the classical RNN model that implement a forward mechanism inspired by residual connections [39] to aid the propagation of the gradient and avoid phenomenons of numerical instability during training. In standard RNNs the input and the hidden state are used at each time-step as inputs to a non linear transformation to obtain the output. This process implies a full transformation of the inputs and hampers gradient propagation. LSTMs and GRUs use different mechanisms to obtain a similar result: computing a transformation of the inputs that is then added to the original input to produce the output. This residual operation gives each time-step a stronger memory about previous events if needed or the ability to discard them depending on how useful they are considered. LSTM and GRU models use different types of gates to compute the output from the inputs. Specifically the LSTM model uses a higher number of gates and passes an additional information vector between the time-steps (the cell state), while the GRU models use only two gates operating directly on the hidden state. For this reasons the GRU models are less computationally intensive, while the LSTM models appear to be more expressive. The bidirectional variant of both the LSTM and GRU models is chosen for this task [40], motivated by its ability to consider changes both forward and backward in time, with the drawback of having to wait for the whole sequence to have concluded to be able to obtain the output. For each of the recurrent architectures considered a model is trained both with and without the addition of the Fisher criterion. The models with the Fisher criterion addition are called, respectively, Fisher criterion bidirectional long-short term memory (F-BLSTM) and Fisher criterion bidirectional gated recurrent unit (F-BGRU).

The loss function used to train the models is the cross-entropy loss with the addition of the Fisher criterion used to minimize the intraclass distance and maximize the interclass distance. The combination of the Fisher criterion with the cross entropy loss function is done through a linear combination controlled by the θ hyperparameter, as shown in (1), where $\mathcal{L}_s$ is the cross entropy loss function and $\mathcal{L}_f$ is the Fisher criterion

$$\mathcal{L} = \mathcal{L}_s + \theta \mathcal{L}_f. \tag{1}$$

The Fisher criterion is reported in (2) and is computed per batch of size m, where μ_{y_i} is the ith class mean of output vectors, δ is the discriminative factor, and O_i is the output vector produced by the model while n is the number of classes

$$\mathcal{L}_f = \frac{1}{m}\sum_{i=1}^{m} \|O_i - \mu_{y_i}\|_2^2 - \frac{\delta}{n(n-1)} \sum_{j=1,k=1}^{n} \|\mu_j - \mu_k\|_2^2 . \tag{2}$$

The model takes a batch of gestures as input and passes it to the two parts of the bidirectional RNN, the forward one and the backward one. Both parts of the RNN consist of a single layer with a dimensionality of 64 units with a ReLU activation function. In the case of the LSTM RNN the cell state also consists of 64 units to allow the gating operations. The two RNNs outputs are joined by concatenation and then averaged over the time dimension by a mean pooling layer. The softmax activation is applied to this output to obtain the final class prediction. In the

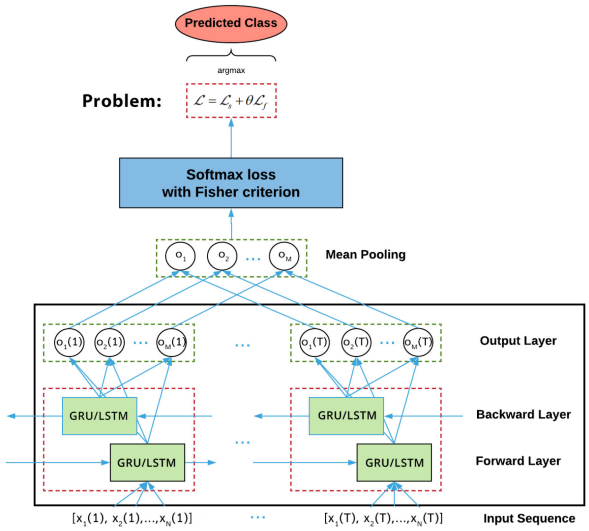


Fig. 6. Diagram of the RNN-based model used. The recurrent cells can be either LSTMs or GRUs.

experiments using the Fisher criterion the relative parameter is added during training. A diagram of the model structure is reported in Fig. 6.

The input data for each gesture has a dimensionality equal to N in the spatial domain, corresponding to the six degrees of freedom (DoF) of the accelerometer (3 DoF) and gyroscope (3 DoF), and a dimensionality equal to T in the time dimension, corresponding to the 500 timesteps to which each gesture is interpolated to in the preprocessing step. During training the model parameters are optimized to minimize the loss function and in this way obtain a better classification performance. Since there is no *a priori* better model between F-BLSTM and F-BGRU, both must be tested to understand which one is better depending on the task at hand. For this reason, both models are tested to understand which is the best one for the task of gesture recognition and user identification.

For the user authentication task a different approach is used: once the model is trained we discard the output of the softmax activation and perform a feature extraction using the unnormalized log probabilities instead. This makes the model produce a 128-dimensional representation for each gesture. We hypothesize that, similarly to what happens in speaker recognition and verification [41], the extracted vector has a high discriminating capacity both with respect to the users performing the gesture and the gesture themselves. The gesture representations are computed for all the training and the testing datasets and then their pairwise distances are computed using a set of distance metrics. All the distance metrics are tested with and without a preceding step of l_2-normalization transforming the representations into unit vectors. The possible pairs of representations are 62 500, reduced to 62 250 by removing the repetitions. In the case of gesture independent authentication we define a correct authentication when for a gesture performed by a user, the closest gesture is one performed by the same user. In the case of gesture dependent authentication we restrict this constraint to be a gesture of the same class performed by the same user.

TABLE I
RESULTS OF THE GESTURE RECOGNITION EXPERIMENT FOR BLSTM, BGRU, F-BLSTM, AND F-BGRU MODELS

	Accuracy		F1-measure	
	Fivefold CV	LOSO	Fivefold CV	LOSO
BLSTM	95.75 ($\pm$ 0.6)	94.35 ($\pm$ 5.1)	95.83 ($\pm$ 0.6)	94.80 ($\pm$ 4.7)
BGRU	96.24 ($\pm$ 0.5)	95.37 ($\pm$ 4.6)	96.29 ($\pm$ 0.5)	95.63 ($\pm$ 4.8)
F-BLSTM	95.90 ($\pm$ 0.8)	95.00 ($\pm$ 4.5)	96.00 ($\pm$ 0.8)	95.38 ($\pm$ 4.3)
F-BGRU	**96.63** ($\pm$ 0.3)	**95.58** ($\pm$ 4.2)	**96.70** ($\pm$ 0.3)	**95.67** ($\pm$ 4.3)

All results are percentage values (%). For each column the global best result is reported in bold, and the results that are statistically equivalent to it at the level $\alpha = 0.05$ are underlined. The results are reported by performing a fivefold cross-validation or a leave one subject out cross validation.

B. Training

All the models are trained using stochastic gradient descent with the Adam optimizer using parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-07}$ and learning rate $\eta = 0.002$. The gestures are fed in batches of 32 samples each and each model is trained for a total of 100 epochs. The discriminative factor of the Fisher criterion δ is set to 0.01, while the scalar θ used for its linear combination with the cross entropy loss is set to 0.1.

C. Evaluation

The metrics used to evaluate the models performance are: accuracy and macro F1-measure for gesture recognition; accuracy, top-3 accuracy, top-5 accuracy, and macro F1-measure for user identification, and error rate (EER) [42] for user authentication.

Concerning the data split, for the gesture recognition experiments a fivefold cross validation and a leave one subject out cross validation (LOSO) are performed. The LOSO technique is particularly exhaustive because it uses, in rotation, the data of a subject as a validation set, while the remaining subjects are used as a training set and represents the performance that the system achieves when a new user not seen during training performs a gesture. For the user identification experiment a fivefold cross validation is performed, while for the user authentication experiment a different split is used by considering the data of 28 of the users as the training set and the data of the remaining five users as the test set.

For the user authentication step the following distance metrics are used, both with and without l_2 normalization.

1) Euclidean distance (Euc).
2) Normalized Euclidean distance (N-Euc).
3) Standardized Euclidean distance (S-euc).

V. RESULTS

In this section we report the performance of the tested models in the three experiments performed.

A. Gesture Recognition

The gesture recognition results are reported in Table I in terms of accuracy and F1-measure. It is possible to see that the use of the Fisher criterion is able to improve the performance in terms of all the metrics considered. The best model, i.e., F-BGRU, reaches an accuracy of 96.63% and an F1-measure of 96.70% for the fivefold CV, and 95.58% and 95.67%, respectively, for

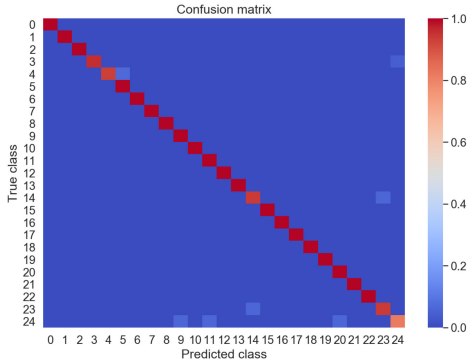


Fig. 7. Normalized confusion matrix of the F-BGRU model trained for gesture recognition.

TABLE II
RESULTS OF THE USER IDENTIFICATION EXPERIMENT FOR BLSTM, BGRU, F-BLSTM AND F-BGRU MODELS

	Top-1 acc.	Top-3 acc.	Top-5 acc.	F1-measure
BLSTM	82.61 (± 2.9)	93.48 (± 1.4)	96.32 (± 0.9)	83.00 (± 2.7)
BGRU	80.65 (± 2.2)	91.92 (± 1.2)	94.76 (± 0.7)	81.00 (± 2.1)
F-BLSTM	**85.24** (± 1.3)	**94.35** (± 0.4)	**96.68** (± 0.2)	**85.57** (± 1.3)
F-BGRU	83.00 (± 0.8)	93.23 (± 0.4)	96.00 (± 0.3)	83.42 (± 0.7)

All results are percentage values (%). For each column the global best result is reported in bold, and the results that are statistically equivalent to it at the level $\alpha = 0.05$ are underlined. The results are reported by performing a fiveold cross-validation.

the LOSO. It is also possible to notice how for this task both GRU-based models outperform their LSTM-based counterparts, with an improvement that is statistically significant at a level $\alpha = 0.05$ in the fivefold CV setup. In order to better understand the recognition performance of the F-BGRU, we report in Fig. 7 the normalized confusion matrix. From the figure it is possible to see how the gesture recognized with the lowest accuracy is the gesture 24, that corresponds to the rejection class. This is expected, due to the heterogeneous nature of this class.

B. User Identification

Table II reports the results for the user identification task in terms of top-1, top-3, top-5 accuracy, and F1-measure. From the results it is possible to see that the best performance is reached by the F-BLSTM model for all the metrics considered, followed by BLSTM that obtains statistically equivalent results at the significance level $\alpha = 0.05$ for three of the four metrics considered. In particular the best model, i.e., F-BLSTM, reaches a top-1 accuracy of 85.24% and an F1-measure of 85.57%. Considering top-3 and top-5 accuracy, the results obtained by the F-BLSTM model are equal to 94.35% and 96.68%, respectively. We can also observe that similarly to the first experiment the use of the Fisher criterion improves the results for both BLSTM and BGRU.

As a further comparison we report in Fig. 8 a plot representing the ratio of subjects having a top-1 accuracy above a given threshold. The plot is drawn for all the four models compared (i.e., BLSTM, BGRU, F-BLSTM, and F-BGRU) considering an accuracy in the range [0.3; 1]. Ideally, a perfect model will have a curve that occupies the top right corner of the plot, meaning a very high top-1 accuracy for all users. From the plot is possible

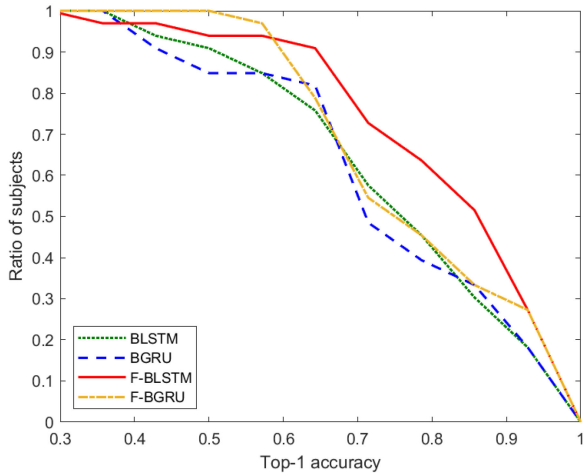


Fig. 8. Curves representing the ratio of subjects having a top-1 accuracy above a given threshold in the range [0.3; 1] for the methods compared in Table II (user identification task).

TABLE III
SPACE AND MEMORY PERFORMANCE FOR THE BEST MODEL FOR GESTURE RECOGNITION (F-BGRU) AND THE BEST MODEL FOR USER IDENTIFICATION (F-BLSTM)

	Gesture recognition	User identification
Total number of parameters	≈ 78 K	≈ 104 K
Model size (bytes of variables)	≈ 313 KB	≈ 416 KB
Model size (on disk)	≈ 31 MB	≈ 26 MB
Memory during model training	≈ 105 MB	≈ 211 MB
Time for single model training	≈ 1 h	≈ 1 h
Time for prediction (average)	23 ms	30 ms

The prediction time is computed as the average time of 10 K predictions.

to notice how the curve of the F-BLSTM shows a higher top-1 accuracy for a larger ratio of users, having a ratio of users less than 5% with a top-1 accuracy lower than 50%.

Comparing the performance of the best models for the gesture recognition task performed in Section V-A and the user identification task performed in this section, it is possible to notice how the level of accuracy reached in the former experiment is higher than that reached in the latter, reflecting the higher difficulty of the user identification task.

The best models for gesture recognition and user identification, i.e., F-BGRU and F-BLSTM, respectively, are compared in Table III also in terms of memory size, training time, and prediction time.

C. User Authentication

This section reports the results for the user authentication task. For this task, the best model identified in the user identification experiment is used (i.e., F-BLSTM). First the model is retrained on the data partitions created to allow the subsequent evaluation of the user authentication task: all the data from 28 randomly selected users is used for training the F-BLSTM model for user identification, while the data from the remaining five users is used for testing. Once the F-BLSTM model has been trained for user identification, the softmax activation is discarded, and feature extraction using the unnormalized log probabilities is performed, thus producing a 128-dimensional representation

TABLE IV
EER RESULTS OF THE USER AUTHENTICATION MODELS FOR GESTURE INDEPENDENT AND DEPENDENT FEATURES

	Gesture independent	Gesture dependent
Euc	38.38 ($\pm$ 0.77)	15.17 ($\pm$ 0.74)
Euc-l_2	36.24 ($\pm$ 0.87)	14.37 ($\pm$ 0.45)
N-Euc	36.62 ($\pm$ 0.89)	14.27 ($\pm$ 0.74)
N-Euc-l_2	36.25 ($\pm$ 0.85)	14.33 ($\pm$ 0.45)
S-Euc	37.69 ($\pm$ 0.55)	13.46 ($\pm$ 0.53)
S-Euc-l_2	**35.62** ($\pm$ 0.78)	**12.94** ($\pm$ 0.29)

For each column the global best result is reported in bold, and the results that are statistically equivalent to it at the level $\alpha = 0.05$ are underlined. All results are percentage values (%). **Euc:** Euclidean distance, **Euc-l_2:** Euclidean distance with l_2 normalization, **N-Euc:** Normalized Euclidean distance, **N-Euc-l_2:** Normalized Euclidean distance with l_2 normalization, **S-Euc:** Standardized Euclidean distance, **S-Euc-l_2:** Standardized Euclidean distance with l_2 normalization.

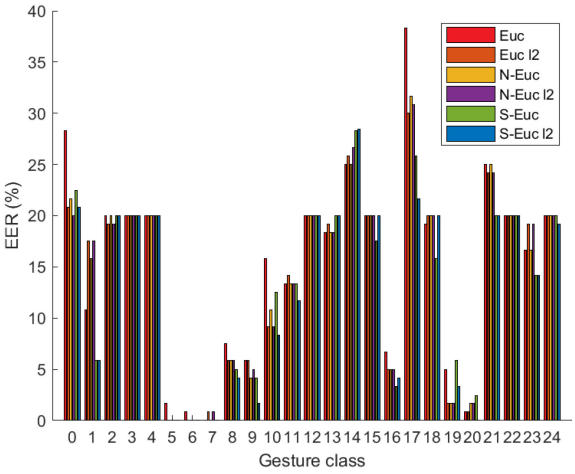


Fig. 9. Per-class EER values in the gesture dependent setup. All the six different Euclidean distance variants considered are reported.

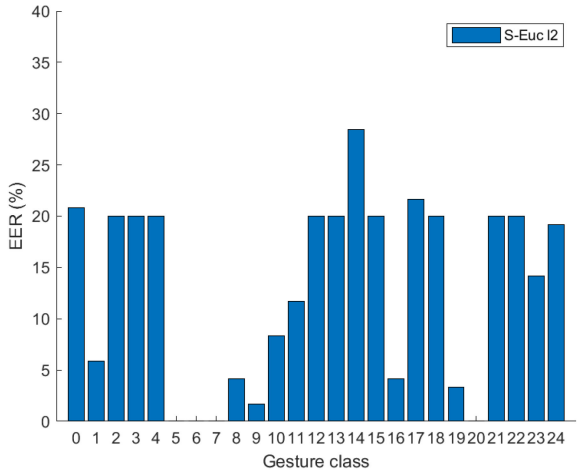


Fig. 10. Detailed view of the per-class EER for the best distance variant (i.e., S-Euc-l_2) in the gesture dependent setup.

TABLE V
EER VALUES FOR THE GESTURE AUTHENTICATION EXPERIMENT

	Gesture dependent robustness
Euc	19.60 ($\pm$ 0.45)
Euc l2	18.66 ($\pm$ 0.12)
N-Euc	18.96 ($\pm$ 0.05)
N-Euc l2	18.85 ($\pm$ 0.11)
S-Euc	18.43 ($\pm$ 0.29)
S-Euc l2	**17.37** ($\pm$ 0.35)

All results are percentage values (%). The best value is reported in bold, and the results that are statistically equivalent to it at the level $\alpha = 0.05$ are underlined.

for each gesture. After the features have been extracted for all the training and test datasets the pairwise distances are computed using the three distance metrics described in the previous section. For each metric considered, both with and without l_2 normalization to unit vectors, user authentication performance are measured by computing the EER.

The numerical results in terms of EER are reported in Table IV for both the gesture independent and gesture dependent setups.

From the results it is possible to see that the standardized Euclidean distance with l_2 normalization (S-Euc-l_2) is the variant of Euclidean distance that achieves the best results in both the gesture independent and gesture independent setups. In particular, the model evaluated in the gesture independent setup reaches an EER of 35.62%, while in the gesture dependent setup the model reaches an EER of 12.94%. The improvement is not statistically significant at the significance level $\alpha = 0.05$ with respect to the results obtained with the other l_2-normalized versions of Euclidean distance in the gesture independent setup, while the improvement is statistically significant in the gesture dependent setup.

In order to better understand the behavior of the model in the gesture dependent setup, we report in Fig. 9 the EER value per each class, using all the different Euclidean distance variants considered. Fig. 10 reports a detailed view of the best result (i.e., S-Euc-l_2) alone.

From the per-class results reported in Fig. 9 we can notice that considering the standardized Euclidean distance with l_2 normalization (i.e., S-Euc-l_2), the user authentication performs noticeably better for some classes of gestures: the recognition of users performing gestures belonging to class 5, 6, 7, and 20 has a zero EER; while five other classes of gestures (i.e., class 1, 8, 9, 16, and 19) have a lower EER compared to the other classes.

Since there is a large gap in performance between gesture dependent and gesture independent user authentication, with the latter having an error rate that is almost three times the former, we perform a further test to understand the performance of the user authentication model in terms of what we could call "gesture authentication": we define a correct authentication when for a gesture performed by a user, the closest gesture belongs to the same class independently from the user that performed it. The results of this experiment are reported in Table V.

From the results reported in Table V it is possible to see that also for this experiment the best type of Euclidean distance is still standardized Euclidean distance with l_2 normalization (S-Euc-l_2), with an improvement that is statistically significant at the significance level $\alpha = 0.05$ with respect to all the other metrics. The comparison with the results in Table IV highlights the importance of gesture recognition for a more accurate user authentication.

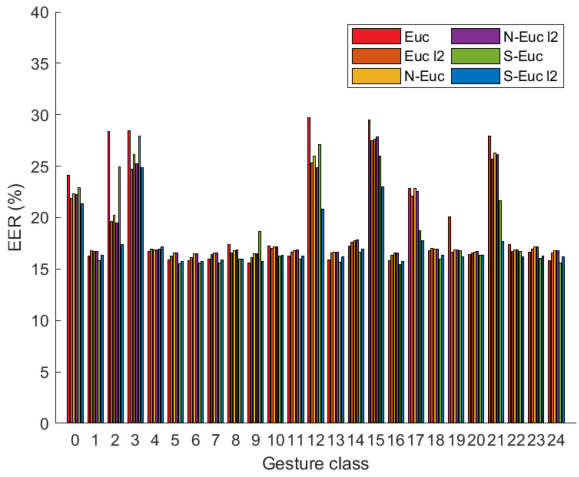


Fig. 11. Per-class EER for the gesture authentication experiment. All the six different Euclidean distance variants considered are reported.

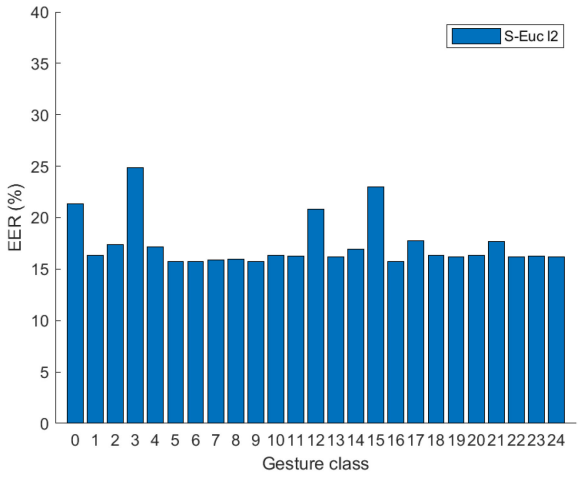


Fig. 12. Detailed view of the per-class EER for the best distance variant (i.e., S-Euc-l_2) in the gesture authentication experiment.

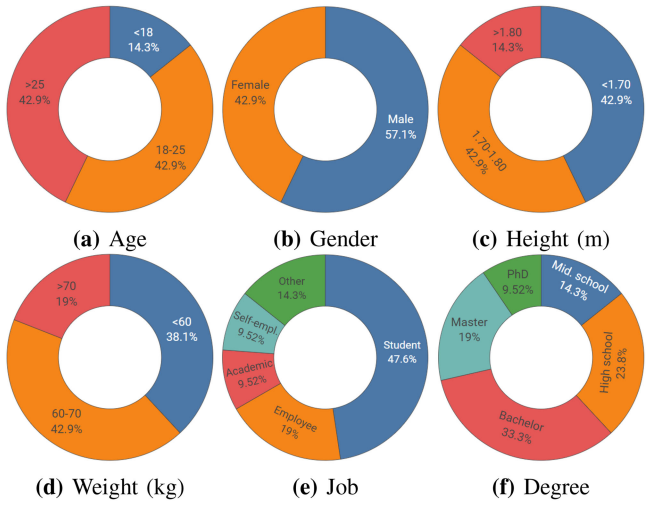


Fig. 13. Details about the subjects that participated to the usability test: age (a), gender (b), height in meters (c), weight in kilograms (d), job (e), and degree (f).

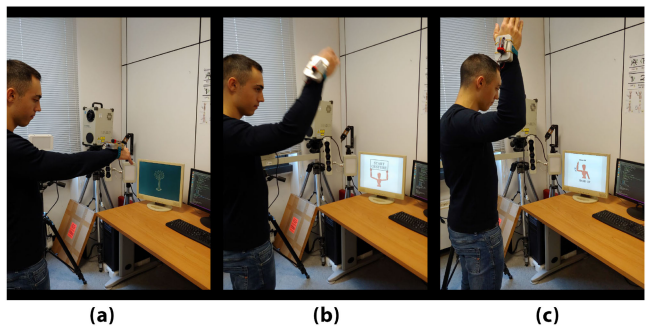


Fig. 14. Images of the setup used for the usability test, where the subjects were standing in front of a computer monitor displaying the output of the gesture recognition algorithm.

As a further analysis, in Fig. 11 we report the EER values for each class for all the variants of Euclidean distances considered, in the gesture authentication experiment. Fig. 12 reports a detailed view of the best result (i.e., S-Euc-l_2) alone. From the results reported it is possible to see how for most of the gesture classes the EER is very close, while there are some gestures (class 0, 3, 12, and 15) for which the EER is higher.

VI. Usability Test

Twenty one subjects tested the system by performing a number of actions chosen from the alphabet of 25 symbols with the aim of assessing the usability of the wristband in terms of effectiveness, efficiency, and user satisfaction. User have judged usability of the system only for gesture recognition. Details about the subjects that participated to the usability test, such as age, gender, height, weight, job, and degree are reported in Fig. 13.

After having trained the users in the use of the wristband, we asked them to perform a gesture randomly chosen from the alphabet. We asked them to repeat this task 20 times by paying attention to select every time a different gesture from the alphabet. The subject was standing in front of a computer monitor displaying the output of the gesture recognition algorithm (see Fig. 14). In fact, after a gesture was performed, the predicted class was showed on the display.

After each test session, we asked subjects to respond to an usability questionnaire of 15 statements we suitably setup with the aim to evaluate the effectiveness, efficiency, and user satisfaction of the system. The first ten statements of the questionnaire are taken from the system usability scale (SUS) questionnaire developed by John Brooke [43], while the last five statements are specifically designed to evaluate the different aspects of the experience and the functionalities of the proposed system. The list of statements composing the questionnaire is reported in Table VI. All the subjects rated the 15 statements on a five-point scale of strength of agreement from "strongly disagree" to "strongly agree." The five-point scale is converted to scores from 1 (strongly disagree) to 5 (strongly agree). Table VI also summarizes the results of the SUS in terms of average and standard deviation across the different subjects.

TABLE VI
15 STATEMENTS OF THE USABILITY TEST QUESTIONNAIRE

#	Statement	Mean	Dev.Std
1	I think that I would like to use this system frequently	3.9	0.8
2	I found the system unnecessarily complex	1.7	1.0
3	I thought the system was easy to use	3.9	1.0
4	I think that I would need the support of a technical person to be able to use this system	2.4	1.3
5	I found the various functions in this system were well integrated	3.9	0.7
6	I thought there was too much inconsistency in this system	1.4	0.6
7	I would imagine that most people would learn to use this system very quickly	4.2	1.0
8	I found the system very cumbersome to use	2.2	1.3
9	I felt very confident using the system	3.8	1.1
10	I needed to learn a lot of things before I could get going with this system	1.8	1.0
11	I think that the recognition of gestures is very fast	4.3	0.9
12	I think that the gestures predicted by the system are accurate	4.6	0.5
13	I found that performing a gesture is very cumbersome	2.1	1.2
14	I noticed that the system recognizes gestures of different size	4.0	1.0
15	I would like to be able to customize the gesture alphabet	4.2	1.3

For each statement the average and standard deviation of the scores provided by the 21 subjects that tested the system are also provided. Scores range from 1 (Strongly Disagree) to 5 (Strongly Agree).

The SUS results show that, with respect to the usability, the system has been rated positively for all the ten statements with a high concordance between users (average of the standard deviation is about 1.0). In particular, the system has been judged easy to use with an average rate of 3.9 and standard deviation of 1.0. With respect to the system functionalities, users rated the speed of the gesture recognition module very high (with an average rate of 4.3) and its accuracy very high (with an average rate of 4.6). The majority of the users (with an average rate of 4.0) noticed that the recognition of the gestures is invariant with respect to the spatial size of the gesture and time used to perform the gesture. All the subject would like to add more gestures in the alphabet (with an average rate of 4.2).

VII. CONCLUSION

In this article, we proposed a prototype system for arm gesture recognition, user identity, and user verification based on inertial signals. The system is composed of hardware and software components. The hardware components are a custom wristband and a server. The software components are the communication libraries, a server application, and the recognition methodologies based on RNN. To evaluate the system we have collected a database of 25 symbols made of letters, numbers, actions and a rejection class. The use of a rejection class is a novelty with respect to the state of the art. Each symbol has been performed by 33 volunteers. The dataset is made public to the community. To make the system robust in real environments, we proposed the use a novel preprocessing approach based on sample interpolation. Thanks to this, input gesture is allowed to be of arbitrary length and enables more input freedom for the end user. Four different RNNs have been considered: BLSTM, BGRU, F-BLSTM, and F-BGRU to find the best model for the three recognition problems at hand. Overall, recognition performance is very promising thus suggesting that the proposed system can be employed in real environments. Gesture recognition achieved above 96% of accuracy, user identification reached about 85% of accuracy, while user verification achieved about 13% of equal error rate. User satisfaction has been assessed with a custom usability test which involved 21 subjects. Outcomes of the test suggested that the user satisfaction is quite high: on average 4.0 points out of 5.0 (with 5.0 being the maximum level of satisfaction).

As future work we would like to make the system more robust in the user identification and verification tasks. To this aim, the possible directions are two: increase the number of users in the database, and prove a more robust neural network solution that can be able to better model the subjectivity of the gestures performed by the user. Furthermore, we would like to make the system more dynamic and provide the ability to add users to the database and symbols to the vocabulary incrementally without the need for batch operations.

Another interesting future development is the deployment of the recognition system in a realspace environment in order to increase the technology-readiness level of the system. We have imagined three possible real-space environments.

The first one considers off-the-shelf wristbands connected via Bluetooth to a smartphone equipped with a neural processing unit (NPU). The NPU is specially designed to enable AI-based functionalities on commercial smartphones, see the survey by Ignatov *et al.* [44] for further details. The second environment is similar to the first one with the difference that the smartphone is used to call server-side services instead of providing neural computations. To make the first two scenarios working properly we need to adapt the IMU signals acquired by the off-the-shelf wristbands to the IMU signals acquired by our system in terms of unit of measure (g or m/s^2), sampling rate etc. The recognition system based on RNNs runs properly on NPU as well as on the server-side.

The third environment considers the use of a development board, such as the Arduino Nano 33 BLE Sense,[2] with the aim of creating a custom wristband with a computational capacity suitable for running tiny machine learning algorithms. [45]. In this case a less computational demanding machine learning algorithm should be developed in order to better fit computational capacity of the development board.

APPENDIX A
STATISTICAL ANALYSIS

This section reports the statistical analysis of all the experimental results of the article. The statistical test used is the two-sample equal-variance t-test. For all the metrics the upper one-sided version is considered, except for EER for which the lower one-sided version is considered. The test is repeated for all the combinations of the entries of each column of each table of experiments. For each test the $p-$value is reported.

Table VII reports the statistical analysis of the results of the gesture recognition experiment for BLSTM, BGRU, F-BLSTM, and F-BGRU models reported in Table I.

Table VIII reports the statistical analysis of the results of the user identification experiment for BLSTM, BGRU, F-BLSTM and F-BGRU models reported in Table II.

Table IX reports the statistical analysis of the EER results of the user authentication models for gesture independent and dependent features reported in Table IV.

Table X reports the statistical analysis of the EER results for the gesture authentication experiment reported in Table V.

[2][Online]. Available: https://store- USA.arduino.cc/products/arduino-nano-33-ble-sense

TABLE VII
STATISTICAL ANALYSIS OF THE RESULTS OF THE GESTURE RECOGNITION EXPERIMENT FOR BLSTM, BGRU, F-BLSTM AND F-BGRU MODELS REPORTED IN TABLE I

BLSTM	-	0.901	0.627	0.991	-	0.802	0.708	0.856	-	0.888	0.643	0.990	-	0.760	0.699	0.782
BGRU	0.099	-	0.222	0.913	0.198	-	0.371	0.576	0.112	-	0.256	0.923	0.240	-	0.412	0.514
F-BLSTM	0.373	0.778	-	0.954	0.292	0.629	-	0.705	0.357	0.744	-	0.948	0.301	0.588	-	0.607
F-BGRU	**0.009**	0.087	**0.046**	-	0.144	0.424	0.295	-	**0.010**	0.077	0.052	-	0.218	0.486	0.393	-

$p-$values lower than the significance level $\alpha = 0.05$ are reported in bold. Left to right: Accuracy in the fivefold CV setup, accuracy in the LOSO setup, F1-measure in the fivefold CV setup, and F1-measure in the LOSO setup.

TABLE VIII
STATISTICAL ANALYSIS OF THE RESULTS OF THE USER IDENTIFICATION EXPERIMENT FOR BLSTM, BGRU, F-BLSTM, AND F-BGRU MODELS REPORTED IN TABLE II

BLSTM	-	0.131	0.949	0.610	**-**	**0.048**	0.891	0.356	**-**	**0.008**	0.796	0.236	-	0.114	0.954	0.627
BGRU	0.869	-	0.998	0.973	0.952	-	0.999	0.975	0.992	-	1.000	0.997	0.886	-	0.998	0.980
F-BLSTM	0.051	**0.002**	**-**	**0.006**	0.109	**0.001**	**-**	**0.001**	0.204	**0.000**	**-**	**0.001**	**0.046**	**0.002**	**-**	**0.006**
F-BGRU	0.390	**0.028**	0.994	-	0.644	**0.025**	0.999	-	0.764	**0.003**	0.999	-	0.373	**0.020**	0.994	-

$p-$values lower than the significance level $\alpha = 0.05$ are reported in bold. Left to right: Top-1 accuracy, Top-3 accuracy, Top-5 accuracy, and F1-measure.

TABLE IX
STATISTICAL ANALYSIS OF THE EER RESULTS OF THE USER AUTHENTICATION MODELS FOR GESTURE INDEPENDENT AND DEPENDENT FEATURES REPORTED IN TABLE IV

Euc	-	0.998	0.995	0.998	0.929	1.000
Euc$-l_2$	**0.002**	-	0.257	0.493	**0.007**	0.865
N-Euc	**0.005**	0.743	-	0.740	**0.026**	0.952
N-Euc$-l_2$	**0.002**	0.507	0.260	-	**0.006**	0.872
S-Euc	0.071	0.993	0.974	0.994	-	0.999
S-Euc$-l_2$	**0.000**	0.135	**0.048**	0.128	**0.001**	**-**

Euc	-	0.964	0.955	0.969	0.999	1.000
Euc$-l_2$	**0.036**	-	0.599	0.554	0.990	1.000
N-Euc	**0.045**	0.401	-	0.440	0.959	0.997
N-Euc$-l_2$	**0.031**	0.446	0.560	-	0.988	1.000
S-Euc	**0.001**	**0.010**	**0.041**	**0.012**	-	0.955
S-Euc$-l_2$	**0.000**	**0.000**	**0.003**	**0.000**	**0.045**	**-**

$p-$values lower than the significance level $\alpha = 0.05$ are reported in bold. Top: Gesture independent; Bottom: Gesture dependent.

TABLE X
STATISTICAL ANALYSIS OF THE EER RESULTS FOR THE GESTURE AUTHENTICATION EXPERIMENT REPORTED IN TABLE V

Euc	-	0.999	0.993	0.997	0.999	1.000
Euc$-l_2$	**0.001**	**-**	**0.000**	**0.016**	0.930	1.000
N-Euc	**0.007**	1.000	-	0.962	0.998	1.000
N-Euc$-l_2$	**0.003**	0.984	**0.038**	-	0.992	1.000
S-Euc	**0.001**	0.070	**0.002**	**0.008**	-	1.000
S-Euc$-l_2$	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**-**

$p-$values lower than the significance level $\alpha = 0.05$ are reported in bold.

REFERENCES

[1] B. Hartmann and N. Link, "Gesture recognition with inertial sensors and optimized DTW prototypes," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2010, pp. 2102–2109.

[2] C. Li, C. Xie, B. Zhang, C. Chen, and J. Han, "Deep Fisher discriminant learning for mobile hand gesture recognition," *Pattern Recognit.*, vol. 77, pp. 276–288, 2018.

[3] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, "Human activity recognition using inertial, physiological and environmental sensors: a comprehensive survey," *IEEE Access*, vol. 8, pp. 210816–210836, 2020.

[4] M. A. S. Mondol, I. A. Emi, S. M. Preum, and J. A. Stankovic, "User authentication using wrist mounted inertial sensors," in *Proc. ACM/IEEE 16th Int. Conf. Inf. Process. Sensor Netw.*, 2017, pp. 309–310.

[5] J. Wu, L. Sun, and R. Jafari, "A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors," *IEEE J. Biomed. Health Inform.*, vol. 20, no. 5, pp. 1281–1290, Sep. 2016.

[6] "Inertial sensors facilitate autonomous operation in mobile robots| Analog Devices," [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/inertial-sensors-and-autonomous-operation-in-robots.html

[7] H. Liu, X. Wei, J. Chai, I. Ha, and T. Rhee, "Realtime human motion control with a small number of inertial sensors," in *Proc. Symp. Interactive 3D Graph. Games*, 2011, pp. 133–140.

[8] A. Pezeshk, "Design and implementation of a 3D computer game controller using inertial MEMS sensors," Master's thesis, Dept. Elect. Comput. Eng., Michigan Technological Univ., Houghton, MI, USA, 2004.

[9] T. Tamura, "Wearable inertial sensors and their applications," in *Wearable Sensors*, E. Sazonov and M. R. Neuman, Eds. Oxford: Academic Press, Jan. 2014, ch.2.2, pp. 85–104.

[10] G. Grossi, R. Lanzarotti, P. Napoletano, N. Noceti, and F. Odone, "Positive technology for elderly well-being: A review," *Pattern Recognit. Lett.*, vol. 137, pp. 61–70, 2020.

[11] I. Ali, S. Sabir, and Z. Ullah, "Internet of things security, device authentication and access control: A review," *Int. J. Comput. Sci. Inf. Secur.*, vol. 14, no. 8, 2016.

[12] S. Bianco and P. Napoletano, "Biometric recognition using multimodal physiological signals," *IEEE Access*, vol. 7, pp. 83 581–83 588, 2019.

[13] Q. Li, R. Gravina, Y. Li, S. H. Alsamhi, F. Sun, and G. Fortino, "Multi-user activity recognition: Challenges and opportunities," *Inf. Fusion*, vol. 63, pp. 121–135, 2020.

[14] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, and R. Jafari, "Enabling effective programming and flexible management of efficient body sensor network applications," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 1, pp. 115–133, Jan. 2012.

[15] G. Fortino, S. Galzarano, R. Gravina, and W. Li, "A framework for collaborative computing and multi-sensor data fusion in body sensor networks," *Inf. Fusion*, vol. 22, pp. 50–70, 2015.

[16] F. G. Hofmann, P. Heyer, and G. Hommel, "Velocity profile based recognition of dynamic gestures with discrete hidden Markov models," in *Proc. Int. Gesture Workshop*, 1997, pp. 81–95.

[17] S. Kallio, J. Kela, and J. Mantyjarvi, "Online gesture recognition system for mobile interaction," in *Proc. IEEE Int. Conf. Syst., Man Cybern. Conf. Theme-Syst. Secur. Assurance*, 2003, pp. 2070–2076.

[18] T. Pylvänäinen, "Accelerometer based gesture recognition using continuous HMMS," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, 2005, pp. 639–646.

[19] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-D accelerometer," in *Proc. Int. Conf. Ubiquitous Intell. Comput.*, 2009, pp. 25–38.

[20] S. Shin and W. Sung, "Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2016, pp. 2274–2277.

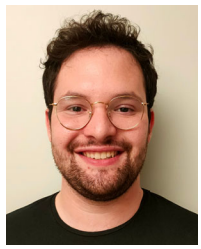
[21] G. Lefebvre, S. Berlemont, F. Mamalet, and C. Garcia, "*BLSTM-RNN based 3D gesture classification*," in *Proc. Int. Conf. Artif. Neural Netw.*, 2013, pp. 381–388.
[22] Y. Wu, Z. Wu, and C. Fu, "Continuous arm gesture recognition based on natural features and logistic regression," *IEEE Sensors J.*, vol. 18, no. 19, pp. 8143–8153, Aug. 2018.
[23] C. Xie, S. Luan, H. Wang, and B. Zhang, "Gesture recognition benchmark based on mobile phone," in *Proc. Chin. Conf. Biometric Recognit.*, 2016, pp. 432–440.
[24] Z. Rui and Z. Yan, "A survey on biometric authentication: Toward secure and privacy-preserving identification," *IEEE Access*, vol. 7, pp. 5994–6009, 2018.
[25] N. L. Clarke and S. Furnell, "Advanced user authentication for mobile devices," *Comput. Secur.*, vol. 26, no. 2, pp. 109–119, 2007.
[26] C. Nickel, T. Wirtl, and C. Busch, "Authentication of smartphone users based on the way they walk using k-NN algorithm," in *Proc. 8th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, 2012, pp. 16–20.
[27] M. De Marsico and A. Mecca, "A survey on gait recognition via wearable sensors," *ACM Comput. Surveys*, vol. 52, no. 4, 2019, Art. no. 86.
[28] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive Mobile Comput.*, vol. 5, no. 6, pp. 657–675, 2009.
[29] M. Conti, I. Zachia-Zlatea, and B. Crispo, "Mind how you answer me! Transparently authenticating the user of a smartphone when answering or placing a call," in *Proc. 6th ACM Symp. Inf.*, 2011, pp. 249–259.
[30] A. F. Abate, M. Nappi, and S. Ricciardi, "I-am: Implicitly authenticate me-person authentication on mobile devices through ear shape and arm gesture," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 3, pp. 469–481, May 2017.
[31] (2020) Key switch KY-004. 2020. [Online]. Available: https://arduinomodules.info/ky-004-key-switch-module/
[32] (2020) Imu 10 DOF waveshare v2. 2020. [Online]. Available: https://www.waveshare.com/product/10-DOF-IMU-Sensor-C.htm
[33] (2020) Mpu9250 library. 2020. [Online]. Available: https://github.com/bolderflight/MPU9250
[34] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "Mqtt-S-A publish/subscribe protocol for wireless sensor networks," in *Proc. 3rd Int. Conf. Commun. Syst. Softw. Middleware Workshops*, 2008, pp. 791–798.
[35] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognit.*, vol. 38, no. 12, pp. 2270–2285, 2005.
[36] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," 2015, *arXiv:1503.06462*.
[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[38] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. 2014 Conf. Empirical Methods Natural Lang. Process.*, Oct. 2014, pp. 1724–1734.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[40] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 6645–6649.
[41] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.
[42] E. Conrad, S. Misenar, and J. Feldman, "Chapter 5 - Domain 5: Identity and access management (controlling access and managing identity)," in *Eleventh Hour CISSP*, third edition ed., E. Conrad, S. Misenar, and J. Feldman, Eds. Syngress, 2017, pp. 117–134.
[43] J. Brooke, *Sus: A "quick and dirty" usability scale*, vol. 189, no. 3, London, U.K.: CRC Press, 1996
[44] A. Ignatov *et al.*, "Ai benchmark: All about deep learning on smartphones in 2019," in *Proc. IEEE/CVF Int. Conf. Comput. Vision Workshop*. 2019, pp. 3617–3635.
[45] C. R. Banbury *et al.*, "Benchmarking tinyML systems: Challenges and direction," 2020, *arXiv:2003.04821*.

Simone Bianco received the B.Sc. and M.Sc. degrees in mathematics and the Ph.D. degree in computer science from the University of Milano-Bicocca, Milan, Italy, in 2003, 2006, and 2010, respectively.

He is currently an Associate Professor of Computer Science with the Department of Informatics, Systems, and Communication, University of Milano-Bicocca. He is an R&D Manager of the University of Milano Bicocca spin off "Imaging and Vision Solutions", and a member of the European Laboratory for Learning and Intelligent Systems (ELLIS). He is on Stanford University's World Ranking Scientists List for his achievements in Artificial Intelligence and Image Processing. His research interests include computer vision, artificial intelligence, machine learning, optimization algorithms applied in multimodal, and multimedia applications.

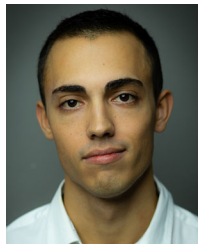
Paolo Napoletano received the master's degree in telecommunications engineering from the University of Naples Federico II, Naples, Italy, and the Ph.D. degree in computer science from the University of Salerno, Salerno, Italy, in 2003, 2007, respectively.

He is currently an Associate Professor of computer science at the Department of Informatics, Systems, and Communication, University of Milano-Bicocca, Milan, Italy. He is Chief of Software Architect with the University of Milano Bicocca spin off "Imaging and Vision Solutions." He is on Stanford University's World Ranking Scientists List for his achievements in Artificial Intelligence and Image Processing. His research interests focus on signal, image and video analysis and understanding, multimedia information processing and management and machine learning for multi-modal data classification and understanding.



Alberto Raimondi received the master's degree in data science from the University of Milano-Bicocca, Milan, Italy, in 2019.

From 2018 to 2021 he worked with the Imaging and Vision Lab, University of Milano - Bicocca, focusing on signal optimization and image detection using deep learning models. He is currently a Machine Learning Engineer with Helixa.ai working on social graph analysis and demographic prediction. His research interests include knowledge representation, architecture optimization, and graph deep learning.



Mirko Rima received the bachelor's degree in computer science, in 2016, and the master's degree in computer science from the University of Milano-Bicocca, Milan, Italy, in 2020.

After graduation he collaborated with a research grant with the Department of Informatics, Systems and Communication, University of Milano-Bicocca. He currently works as a Data Scientist with an artificial intelligence company, Vedrai Spa, Italy. His research interests include gesture recognition with a focus on deep learning methods.