Gamboa-Montero, Juan José; Alonso-Martín, Fernando; Castillo, José Carlos; Malfaz, María; Salichs, Miguel A. (2020). Detecting, locating and recognising human touches in social robots with contact microphones, *Engineering Applications of Artificial Intelligence*, v. 92, 103670.

DOI: https://doi.org/10.1016/j.engappai.2020.103670

Detecting, Locating and Recognising Human Touches in Social Robots with Contact Microphones

Juan José Gamboa-Montero [a], Fernando Alonso-Martín [a,*], José Carlos Castillo [a], María Malfaz [a], Miguel A. Salichs [a]

[a] *Robotics Lab, Universidad Carlos III de Madrid, Av. de la Universidad 30, Leganés, Madrid 28911, Spain.*

Abstract

There are many situations in our daily life where touch gestures during natural human-human interaction take place: meeting people (shaking hands), personal relationships (caresses), moments of celebration or sadness (hugs), etc. Considering that robots are expected to form part of our daily life in the future, they should be endowed with the capacity of recognizing these touch gestures and the part of its body that has been touched since the gesture's meaning may differ. Therefore, this work presents a learning system for both purposes: detect and recognize the type of touch gesture (stroke, tickle, tap and slap) and its localization. The interpretation of the meaning of the gesture is out of the scope of this paper.

Different technologies have been applied to perceive touch by a social robot, commonly using a large number of sensors. Instead, our approach uses 3 contact microphones installed inside some parts of the robot. The audio signals generated when the user touches the robot are sensed by the contact microphones and processed using Machine Learning techniques. We acquired information from sensors installed in two social robots, Maggie and Mini (both developed by the RoboticsLab at the Carlos III University of Madrid), and a real-time version of the whole system has been deployed in the robot Mini. The system allows the robot to sense if it has been touched or not, to recognise the kind of touch gesture, and its approximate location. The main advantage of using contact microphones as touch sensors is that by using just one, it is possible to "cover" a whole solid part of the robot. Besides, the sensors are unaffected by ambient noises, such as human voice, TV, music etc. Nevertheless, the fact of using several contact microphones makes possible that a touch gesture is detected by all of them, and each may recognize a different gesture at the same time. The results show that this system is robust against this phenomenon. Moreover, the accuracy obtained for both robots is about 86%.

Key words: Acoustic Sensing; Social Robots; Touch gesture recognition; Touch localisation; Human-Robot Interaction; Machine Learning applications

1 Introduction

During human-human interaction there are many communication channels, some of them are verbal, and others not verbal, such as facial expressions, body gestures, and, in many cultures, physical interaction, that is, touch gestures (Gallace & Spence, 2010). In fact, in some situations, humans try to communicate, or even emphasize, important social messages using these non-verbal communication channels (e.g. to tap on the back trying to get a person's attention, to comfort someone giving him a hug, or when a father caresses his crying son's face) (Hertenstein et al., 2006, 2009). All those touch gestures are easily recognizable for everyone (depending on their culture) since complex languages have incorporated these touch interactions (Wilhelm et al., 2001). On the other hand, it is important to note that, depending on the part of the body that has been touched, gestures may have different meanings.

Due to the ability of touch gestures to communicate or improve the other communication channels, some studies have explored how this kind of interaction can be used to improve Human-Robot Interaction (HRI) (Schmid et al., 2007), (Altun & MacLean, 2015). In fact, considering that social robots are designed to interact with people and are expected to behave following social norms during HRI, it seems logical to think that they should be able to perceive and recognise [1] different touch gestures to behave appropriately (Kim et al., 2010; Silvera-Tawil et al., 2011; Altun & MacLean, 2015; Jung et al., 2015). As an example, Paro, the seal-robot designed to interact with elders with cognitive impairment, has proven to improve their mood by taking and stroking the robot as if it were a real animal (Sabanovic et al., 2013; Sharkey & Wood, 2014).

With respect to sensing technologies, social robots are commonly endowed with basic tactile sensors (e.g. capacitive, force or temperature sensors) to detect physical contact (throughout the text this will be called 'touch activity detection'). Some studies have aimed to recognise 'touch gestures' (contacts made on a surface with a certain communicative intention) using these devices (Argall & Billard, 2010). Nevertheless, most of these proposals usually require important hardware deployments that are oriented towards equipping the robot with large amounts of sensors (Stiehl et al., 2005). Moreover, dealing with such a large amount of sensors increases the chance of false positives, leading to low recognition rates.

* Corresponding author. Tel: +34 626540365
Email addresses: jgamboa@ing.uc3m.es (Juan José Gamboa-Montero), famartin@ing.uc3m.es (Fernando Alonso-Martín), jocastil@ing.uc3m.es (José Carlos Castillo), mmalfaz@ing.uc3m.es (María Malfaz), salichs@ing.uc3m.es (Miguel A. Salichs).

[1] In this paper, recognition and classification are treated as synonyms.

The objective of the work presented in this paper is to develop and implement a system to learn to recognise and localise a touch gesture made on a social robot. Besides, this system uses a small number of sensors unlike other approaches described above. The present work builds on the preliminary results presented in Alonso-Martín et al. (2017). Our contribution explores a novel application of a sensing technology, piezoelectric pickups, in Human-Robot Touch Interaction. These devices perceive the sound vibrations that are generated when a user touches the robot's surface. The perturbations originated by the contact propagate through the rigid parts of the robot (its shell and inner structure). One of the main advantages that these devices offer is that they are not affected by usual ambient sounds propagated in the air, such as the human voice, TV, music, etc.

The system is able to classify both contact location and touch gestures, working as follows: When one of multiple sensors perceive an interaction, the system commences to process their signals separately, extracting a group of features that belong to the time and frequency audio domains [2] from each of them such as Signal to Noise Ratio, Pitch or duration. When the system detects that the contact has ended, it computes the features' average values during this timespan. These values are then grouped into labelled instances that represent the contact performed. A dataset composed by a set of these instances is later used as training data for further classification of the gesture through machine learning techniques.

The main contributions of this paper are: the ability to recognise touch gestures on the different parts of the robot's surface, and to identify the zone in which the touch gesture takes place. Both, the type of gesture and its localisation are fundamental to be able to make a correct interpretation of the communicative message of the user. This paper focuses on the learning process of the recognition of the type of gesture and its localization, the interpretation of its possible meaning is out of the scope of this paper.

Moreover, our system also follows a modular design that is able to adapt to different robotic platforms. In order to prove this, data acquisition part of the system proposed has been implemented on two robotic platforms, the social robots Maggie (Gonzalez-Pacheco et al., 2011) and Mini (Salichs San Jose et al., 2016), which have been developed by the RoboticsLab at the Carlos III of Madrid University. Both robots have a similar physical structure: head, body and two arms at their sides. Finally, the real-time version of the whole system has been integrated in the robot Mini.

As will be later explained, three piezoelectric pickups (or contact microphones) will be integrated inside the rigid parts of each of the robots' shells: in

[2] The complete set of features is shown and explained in Section 4.2.

their heads and inside each arm. As already explained, this type of touch sensors perceives the sound vibrations generated when the robot's surface is touched and these perturbations propagate through the rigid parts of the robot. Therefore, we have different microphones detecting simultaneously in connected rigid bodies (head and arms). This is important because extending the touch recognition to multiple sensing devices means that touch gestures can be recognised by several sensors at the same time. This fact generates a problem considering how the nature of the sound signal affects both localisation and identification of gestures. On one hand, regarding localisation, sound propagates over the rigid parts of the robot (hard shells and inner structure) and different microphones may sense it at the same time. This behaviour complicates the task of locating where the user touched the robot. On the other hand, concerning touch gesture recognition, the propagation of the sound signal changes as it moves through the irregular surface and inner structure of the robot (note that our robots are composed by different rigid parts connected). This causes that each microphone perceives the signal in a different way depending on the distance that separates each sensor from the signal source. For instance, touching the arm of the robot, not only introduces the vibrations corresponding to the touch itself, but also the extra movement caused by the force exerted on that part.

Therefore, it is possible that each sensor recognises different touch gestures at the same time. In this work, we studied all these effects and our results show that our system is robust against these phenomena.

The rest of the paper is structured as follows. First, Section 2 reviews the literature related to touch interaction in social robotics and describes different systems that integrate contact microphones to locate and classify touch gestures on different kinds of surfaces. Section 3 details the hardware involved in this proposal: two robotic platforms with contact microphones beneath their respective shells. Then, Section 4 explains the structure and the phases of the proposed system primarily at a software level. Section 5, describes the experimental part of this paper: the set of gestures selected, the data collection process, the way to construct the dataset and the evaluation metrics used. Next, Section 6 and Section 7 present and discuss the results obtained in the classification process, respectively. Finally, Section 8 shows the conclusions that have been gathered from the results of the previous section.

2 Related Works

In recent years, touch interaction has attracted attention and has been introduced in many different areas, such as domotics, electronics, or robotics. Nowadays, devices such as smartphones, wearables, laptops or

tactile fingerprint sensors implement technology related to touch interaction (Murray-Smith et al., 2008; Robinson et al., 2011; Wang et al., 2019). In this section, we analyse the literature, while paying special attention to two trends: first, we review the solutions related to Touch Interaction in HRI, deepening in the sensors implemented on several social robots; and second, we focus on the evolution of touch interaction based on acoustic sensing and applied to different contexts or scenarios, extending robotics.

2.1 Touch-sensing technologies in social robots

Several sensing technologies traditionally have been applied to touch detection and touch gesture recognition in robotics. Nicholls & Lee (1989) reviewed different proposals based on endowing robots with skills related to touch recognition. These authors found that touch interaction in robotics implemented mostly capacitive, resistive, mechanical and optical sensors, due to their robustness and durability. The authors also addressed some disadvantages of this kind of sensor—mostly their susceptibility to noise and heat, and that their capacitance decreases as the surface size increases, limiting its spatial resolution. The sensing technologies that were analysed in this survey are still popular.

More recently, Argall & Billard (2010) studied how social robots that are made of different materials and shapes integrate tactile technologies in their designs. This paper classified robots by taking into account the consistency of their shell—distinguishing between soft or hard-skinned robots. According to this survey, robots such as WENDY (Morita et al., 1999) belonged to the hard-skinned group. This kind of hard-skin robots usually integrate force/torque, force sensitive resistors (FSR), accelerometers, capacitive sensors, and deformation sensors to detect touch. They are able to detect physical contact, but they cannot locate its source nor categorize the kind of contact performed. Soft-skinned robots such as Paro (Sabanovic et al., 2013; Sharkey & Wood, 2014) and CB2 (Minato et al., 2007) tend to be equipped with the following sensors to detect the physical contact: piezoelectric, FSR, capacitive sensors, potentiometers that provide kinesthetic information, temperature sensors (thermistors), electric field sensors and photo reflectors. PROBO, which is a soft-skinned robot shaped like a huggable animal-like creature (Goris et al., 2011), has more than 1000 force sensors that are arranged in a grid (which are able to detect the amount of pressure exerted), 400 temperature sensors and nine electric field sensors. This system senses contact and recognises some kinds of touch gestures.

In their research, Silvera-Tawil et al. (2011, 2014) focused on advanced touch interaction systems, designed to identify gestures. They carried out several

experiments with an artificial arm covered by a sensible skin layer. Their prototype recognised six different kinds of contact and a LogitBoost algorithm performed classification, reaching an accuracy of 74% in cross-validation with a dataset composed of 1050 instances obtained from 35 users.

Albawi et al. (2018) followed a similar hardware setup as Silvera et al.: an artificial arm covered by a sensible skin layer that is able to register the pressure applied to it. This information is processed with a Convolutional Neural Network that achieves an accuracy of 63.7% using cross-validation. In this work, the authors propose a set of 14 touch gestures. Müller & Gross (2018) proposed a combination of capacitive and pressure sensors. These were mounted on an assistive robot, achieving an accuracy of 74% when recognising between four possible gestures. These authors integrated a probabilistic classifier, particularly the Gaussian Mixture Models, and cross-validation. Hughes et al. (2017) also proposed the use of deep learning to deal with touch recognition on social robots. This system achieves an accuracy of 61.35% with a combination of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Zhou & Du (2016) proposed an evolution of these techniques using a 3D CNN achieving the accuracy of 76.1% using the same database that Hughes.

Cooney et al. (2012) continued along the same line, presenting a survey that tried to differentiate between 20 affective contacts on a humanoid robot (kisses, hugs, strokes in the cheek, handshakes, etc.). Their approach implemented a combination of artificial vision techniques and Kinotex tactile sensors [3]. In this case, the machine algorithm selected was an SVM classifier, and they claimed that they were able to get 90.5% accuracy using cross-validation over a dataset formed by 340 instances gathered from 17 users.

Most of these works tend to agree that the traditional touch sensors used in HRI have some shortcomings, including short-range, tendency to false positives, susceptibility to noise, inability to recognise touch gestures, poor scalability and, in some cases, high complexity.

2.2 Acoustic-sensing technologies applied to touch interaction

Traditional touch-sensing technologies (e.g. capacitive sensors) suffer from drawbacks such as hardware complexity, high manufacturing cost, and high power consumption. These technologies can also affect the platform where they are present. For example, they can reduce the optical performance and transparency in touch screens or introduce cross-talks with other electronics

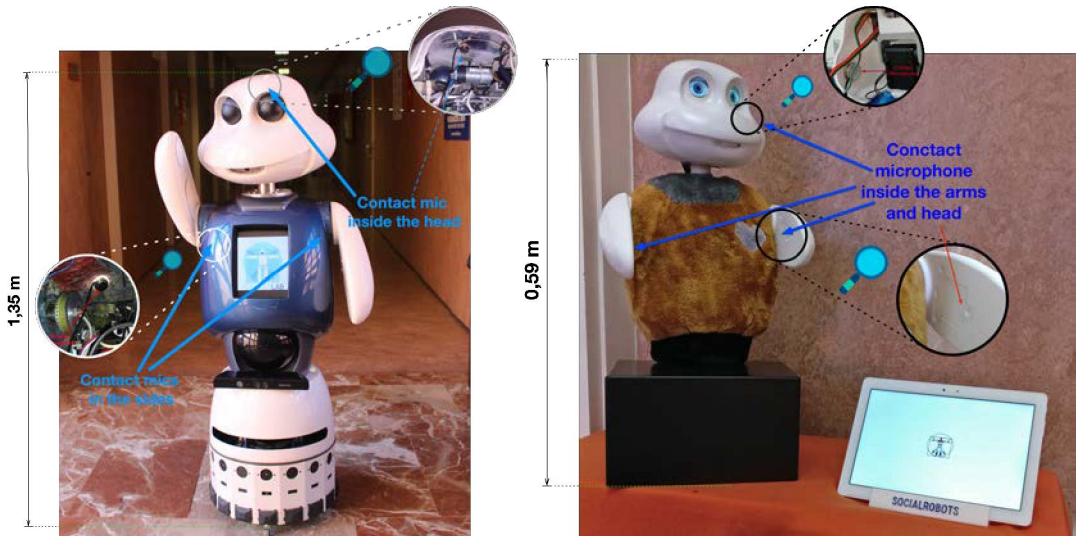
[3] Details about the Kinotex sensor: `http://www.esa-tec.eu/workspace/assets/files/1203588444_1246-51ba009775868.pdf`

in the device (Walker, 2012). Therefore, using acoustic devices may pose an attractive alternative for touch interaction.

Paradiso & Checka (2002) presented a system to locate and classify touch interactions upon an interactive tap window. They placed four contact microphones (also known as piezoelectric pickups) on each corner of a square glass screen. Touch localisation was based on Time Difference of Arrival (TDOA) analysis (Cho et al., 2015), which was evaluated using a cross-correlation technique. They reported a high performance, with an accuracy of 2 to 4 cm on a surface of 2.24 square meters. The system was also able to differentiate between touch gestures (i.e. knock, tap or bang). Later, Lopes et al. (2011) presented a system that extended traditional multi-touch systems by mixing two technologies: capacitive sensors to detect the position of the touch, and acoustic sensing devices to recognise different kinds of touch gestures. The user established contact with the a glass surface of 1.12 square meters, using different parts of the hand and expressing gestures such as finger taps, a knock, a slap, and a punch.

Nikolovski (2013) developed a 10 mm thick and 1 square meter screen panel that was capable of locating low-energy contacts generated by fingernail taps. The system implemented Lamb wave absorption as localisation principle, a technique derived from the TDOA algorithm (Nikolovski, 2003). The system had a spatial resolution of 2 mm positioning the contacts. Also following the Lamb wave principle, Firouzi et al. (2016) presented an ultrasonic touch screen system that detected multiple touch contacts simultaneously with high contact sensitivity. This system had a resolution of 0.5 cm^2, taking into account the size of the cells composing the surface. Even though all these works achieved good accuracy results, these systems are focused on locating the contact's source and they were not designed to recognise the kind of touch gesture performed.

Acoustic technologies have traditionally obtained good performance on flat, glass-like surfaces. Therefore, we believe that this sensing technology could shed good results when implemented in social robots with rigid surfaces; however, in this case we must consider the challenge of recognising touch gestures in rounded and irregular surfaces. Besides, in this contribution we align with the proposals studied regarding the set of touch gestures detected and also take inspiration from the work of Yohanan & MacLean (2012), resulting in the four touch gestures, tap, slap, stroke and tickle, described in Section 5.1.



(a) The social robot Maggie (b) The social robot Mini

Fig. 1. Robotic platforms, showing the location of the acoustic sensors.

3 Hardware components of the system

The system proposed in this work integrates piezoelectric pickups in two robotic platforms as touch sensors. This section provides some insights into the hardware platforms, detailing how the sensors have been installed on the rigid surfaces of the platforms.

3.1 Robotic platforms

The touch sensing system was first integrated in the robot Maggie (see Fig. 1a). This robot was developed by the RoboticsLab at the Carlos III University (Spain) as a research platform on HRI (Gonzalez-Pacheco et al., 2011; Salichs et al., 2006). Since Maggie is a social robot, its external appearance has to be friendly. This means that it has, for example, an array of LEDs to represent its mouth, which will light up as the robot speaks, or two animated eyelids, that endow it with a life-like blinking. Maggie is 1.35 meters tall and has a wheeled base to allow it to move around. Originally, the robot had some capacitive sensors installed on its body, which allowed it to sense contact in some specific portions of its head and body. In this work, the touch sensing capabilities of the robot are extended by integrating the acoustic devices.

The touch system proposed in this work was also integrated in the social robot Mini (see Fig. 1b). Mini was developed by the RoboticsLab and is a small desktop robot (0.59m tall) with an external appearance based on its predecessor Maggie. It was designed to assist and entertain the elderly who

have cognitive impairment through improving their social skills (Salichs San Jose et al., 2016). Some of these abilities include telling stories, expressing emotions, showing pictures or videos on its tablet or playing games that belong to a cognitive stimulation exercise program. Mini has LEDs to represent its cheeks and heart, a VUmeter as its mouth, and two OLED screens serving as eyes. Mini also has servomotors in its arms, base, and neck to allow simple but natural movements to convey liveliness. In its first version,the robot also integrated capacitive sensors on its body, which were made of foam, specifically on both its shoulders and belly.

3.2 Touch sensors: contact microphones

Section 2 analysed the feasibility of using contact microphones as touch sensors. With these devices, it is possible to extract features from the characteristic vibration patterns generated by touch gestures and, therefore, they enable accurate touch classification. The reach of this kind of sensor also allows us to cover the entire shell of the robot using only a few microphones. The external noise does not affect the performance of the system because the sensors only receive the vibrations transmitted by the solid material in direct contact with the pickups. This property is also one of this device's main drawback because their performance heavily relies on the material that they are attached to (these sensors do not work on non-rigid materials) and also on the uniformity of the contact. In the case of the robot Maggie, this was not a problem because Maggie has a hard shell that covers all of its body. However, in the case of Mini, its trunk is made of foam, whereas the head and arms are made of a rigid material. Figure 1 shows the sensor placement in both robotic platforms.

The system integrates an *Oyster Schaller 723* [4] contact microphone. This sensor consists of a polished and chromed oyster-shaped piezoelectric pickup, with a chrome silver cover pre-wired to a 1/4" standard instrumental cable. This device provides several advantages, such as not requiring active circuitry or pre-amplification.

3.3 System setup

The integration of the microphones followed a similar process in both robotic platforms. In both robots, we integrated three receivers beneath each robot's shell to ensure that they would not hinder interaction. In the robot Maggie, one contact microphone was placed on the internal side of the robot's head shell,

[4] https://schaller.info/en/search?sSearch=Oyster+723

and the other two pickups are inside its left and right shoulders. Maximising the contact area and achieving a uniform fitting was crucial to guarantee good sound acquisition. Consequently, it was necessary to use modelling clay to create a smooth and homogeneous layer between the microphones and the inner parts of the shell, mostly because the latter was concave and rough (see Fig.1a).

In the case of the robot Mini, we placed the pickups on the rigid surfaces on its head and arms. The main reason to discard Mini's shoulders (mimicking Maggie's setup) was that Mini's torso is made of foam. Sound-waves propagate with difficulty through foam as a solid mainly due to its lack of rigidity. The rest of Mini is made from ABS, which is one of the most popular materials in 3D printing, including the head, arms, and the internal structure of the robot. Inside the head, the microphone was placed on the left cheek, while two microphones were placed inside compartments in the arms that were made for this purpose; as shown in Figure 1b.

4 Software components of the system

This section describes the different phases of our touch gesture recognition system, namely: Sound Acquisition (SA), Feature Extraction and Touch Activity Detection (FED), Instance Creation (IC), and finally, Touch Classification and Localisation (TCL). Figure 2 shows a summarized view of the operation flow where each contact microphone implies a pipeline of audio analysis consisting in the SA and FED phases. The touches performed on the robot's shell are received by all of the contact microphones built into the robot, what implies a parallel analysis of the sound collected by each pickup. Finally, all the features obtained in each pipeline are merged together in instances in the IC phase, which produces a dataset that can then be used to train different families of classifiers to recognize and locate touches [5] in the TCL phase (see Section 4.4).

4.1 Sound acquisition

When the user touches the robot, an acoustic vibration propagates through the robot's shell. The contact microphones integrated into the inner side of the shell of the robot collect this perturbation. According to the propagation of the sound waves (Thomson, 1950), it would be expected then that those receivers

[5] A brief documentation of our system can be found at: `https://trekirk.github.io/acoustic_touch_recognition`

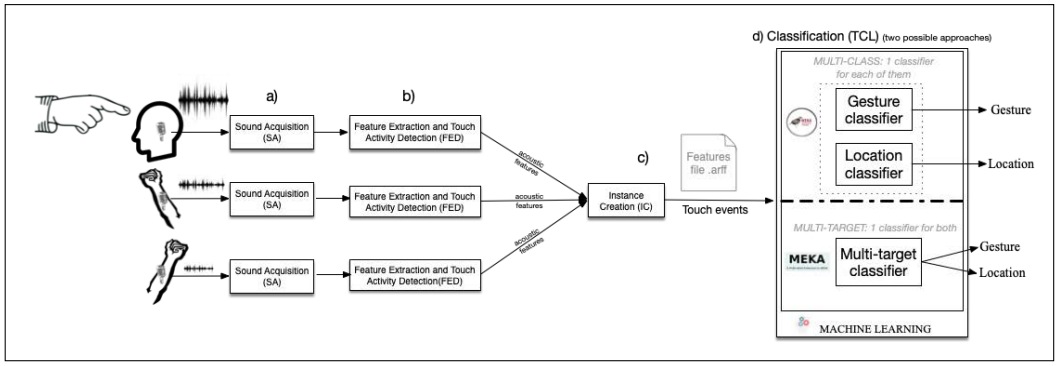


Fig. 2. Data flow scheme: a) the touch is performed by the user and the sound vibration is transmitted through the robot's shell and is collected by each contact microphone built-into the robot; b) the system perceives a touch gesture and computes the significant values of the audio features (max, min, average) until the gesture ends; c) these values from all the audio receivers are stored in a single instance, and subsequently saved in a dataset; d) the classifiers considered take as input the instances in the dataset and outcome the kind of touch and its approximate location.

located closest to the point of touch contact acquire a stronger signal than those that are located farther away. But in practice, this behaviour does not always appear. In fact,, there are cases in which strong touches are registered by different sensors with similar intensities. In other cases, softer touches are only perceived by the closest sensor. So, to improve the detection of any possible sounds on the robot while minimising the number of sensors installed, it is essential to adjust the position of each microphone and their input volume, accordingly.

The sound propagation phenomenon is illustrated in Figure 3. This figure shows a graphical representation of relevant features extracted from two instances in our dataset, although this situation repeats in the data collected. Each chart shows how the detection of some relevant features may not correspond to the intensities expected when slapping the robot in different places. That is, when slapping [6] the robot on the head (see Fig. 3 a)), the microphone in the head detects the highest values for all of the features. This would be the expected situation: when a user touches a body part of the robot with a microphone, that sensor detects the higher intensity. In contrast, this case does not seem to be that clear, as shown in Figure 3 b), where we can see how a slap on the left arm does not provide the highest intensity in the microphone located in the arm but in the head instead. This inconsistency may lead to misclassification, both in terms of localisation and gesture recognition.

[6] According to Yohanan & MacLean (2012), the term 'slap' can be defined as: *quickly and sharply strike the Haptic Creature with your open hand.* Section 5.1 elaborates on this aspect.

Therefore, the aim of this work is to study how machine learning can help to mitigate this problem.

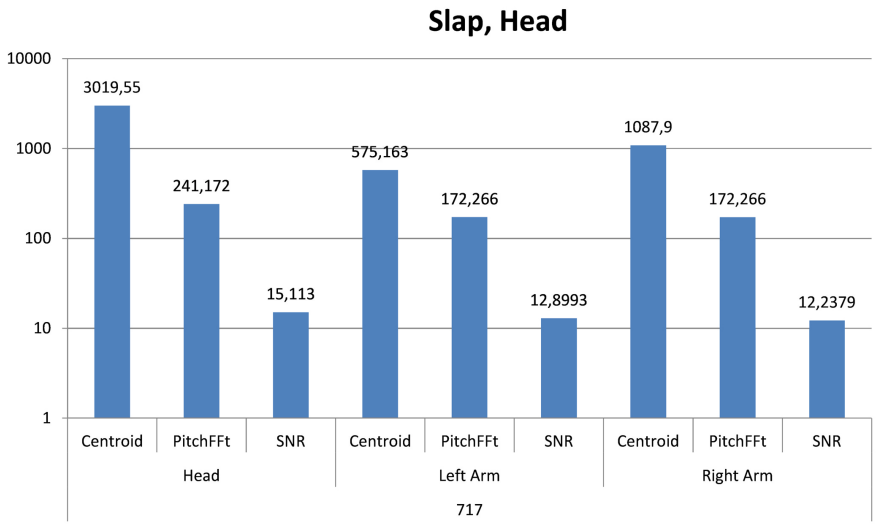
Note that the propagation time of sound waves through the robot's body does not affect the microphones response for two reasons: first, the system does not detect instantaneous samples separately. Instead, it uses a window-like scheme to group detections along time intervals as described in the next section. Additionally, given the sound propagation speed and the short distances sound travels within the robot, the three receivers acquire the sounds almost at the same time.

4.2 Feature extraction and touch activity detection

Before implementing our audio processing system, responsible for extracting and computing the sound features of our interest, we were looking for alternatives having some basic requirements in mind. In the first place, we wanted an open-source tool compatible with *ROS* [7] (the middle-ware present in our robotic platforms and a standard the facto in robotics) and adaptable enough to adjust to our changing robot internal architecture. Second, we wanted a tool able to work in real-time having as less delay as possible. Finally, the third requirement was that the audio processing task needed to work in two audio domains (time, and frequency using Fast Fourier Transform). In our previous works we already explored the performance of systems that extract features from these audio domains in the context of voice recognition (Alonso-Martin et al., 2013a,b, 2014).

Software like *Praat* [8] or *CSound* [9] met some of these requirements, but in the end, not fulfilling one of these requirements would compromise the performance of the final system. For example, being able to process sound in real-time is a mandatory requirement, and Praat does not have this functionality. Real-time alternatives may include *Matlab* or *Octave*, but in both cases, connecting an application from these environments to ROS involves an extra delay in real-time processing. For all these reasons, we finally chose *ChucK* [10], a versatile audio processing programming language traditionally used by musicians and digital artists that allows real-time sound processing with high performance.

[7] ROS (acronym for *Robot Operating System*) is a framework widely used in robotics to communicate the different system's components. Homepage: `www.ros.org`
[8] `http://www.fon.hum.uva.nl/praat`
[9] `https://csound.com`
[10] `http://chuck.cs.princeton.edu`



(a) Bar chart representing the instance in the line 717 of the ARFF file, a slap in the robot's head

(b) Bar chart representing the instance in the line 1084 of the ARFF file, a slap in the robot's left arm

Fig. 3. Different instances represented in bar charts in logarithmic scale. The most significant features were selected to show the variations depending on the area of contact. The *pitchFFT* and *centroid* features are measured in *Hertz*, and the *Signal to Noise Ratio* (SNR) is dimensionless.

This phase runs independently for each receiver and constitutes the core of our system. The software consist of a series of scripts developed through ChucK. All of the executables are managed using ROS, which allows them to communicate with the rest of the system. For each microphone, the scripts perform three tasks. First, a set of acoustic features is gathered (showed in Table 1) to detect the beginning of the touch gesture. Second, once the system detects touch activity, the program computes the most relevant values

Table 1
The set of audio features computed.

Feature	Description	Domain
Pitch	Frequency perceived by human ear.	Time, Frequency
Flux	Feature computed as the sum across one analysis window of the squared difference between the magnitude spectra corresponding to successive signal frames. In other words, it refers to the variation of the magnitude of the signal.	Frequency
RollOff-95	Frequency that contains 95% of the signal energy.	Frequency
Centroid	Represents the median of the signal spectrum in the frequency domain. That is, the frequency at which the signal approaches the most. It is frequently used to calculate the tone of a sound or timbre.	Frequency
Zero Crossing Rate (ZCR)	Indicates the number of times the signal cross the abscissa.	Time
Root Mean Square (RMS)	Amplitude of the signal volume.	Time
Signal to Noise Ratio (SNR)	Relates the touch signal with the noise signal.	Time
Duration	Duration of the contact in time (seconds).	Time
Number of contacts per minute	A touch gesture may consist of several touches, this feature reports the number of contacts.	Time

of the features (average, maximum, minimum) according to the duration of the gesture. Finally, each script sends this information to the next phase.

The scripts running in this phase sample the input signal at 44100 Hz, gathering the information in windows of 256 samples. For each set, the sound features are computed in two domains—time and frequency—, which are related to the time domain directly obtained from the sampled analogue signal that is acquired from the microphones. In the case of the features belonging to the frequency domain, the Fast Fourier Transform (FFT) is applied to the time-domain signal (Cochran et al., 1967). The system keeps extracting the instantaneous values of the features highlighted in Table 1 until the extracted information matches the decision rule shown in Eq. 1.

$$T_a = \begin{cases} TRUE, & if\ SNR_c > SNR_\tau \\ & AND\ C_2\ ...\ AND\ C_n \\ FALSE, & otherwise \end{cases} \tag{1}$$

Where T_a represents the touch activity (a touch gesture event), and SNR_c and SNR_τ are the current Signal to Noise Ratio and the SNR fixed in the threshold, respectively. The Signal to Noise Ratio is a comparison between the current signal volume (or signal energy) and the noise signal volume (or average ground noise energy) in the frequency domain (Tucker, 1966; Carlson, 1968). The average ground noise energy is updated whenever system is not perceiving contact. Thus, in each iteration the SNR is computed dividing the

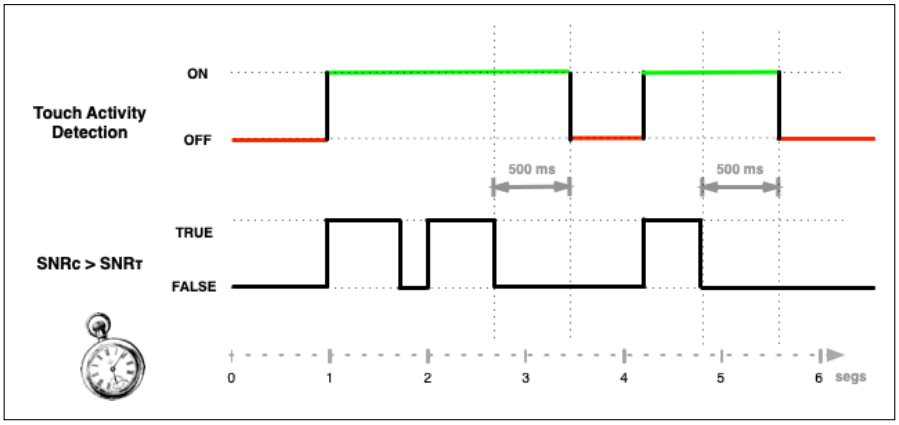


Fig. 4. Illustration of Touch Activity Detection based on analysis of some features, such as SNR, particularly in the figure using the relation between the SNR current (SNR_c) and a SNR threshold (SNR_τ). The beginning of the gesture is detected when the SNR_c is greater than SNR_τ and the end of the gesture is detected when SNR_c is lower than SNR_τ during a time span.

signal energy by the average ground noise energy. The system adds thresholds for the other features extracted (represented as the conditions C_i in the Eq. 1). These thresholds can be adjusted depending on the variability of factors such as the materials composing the robot, the type of microphones selected and where they are, and the input volume of the sound cards or the conditions of the experiment (e.g. external or internal noises). Since the platform stays static during the experiments, the SNR is the only feature in the decision rule (Lopes et al., 2011) for this experiment. Some more values served as thresholds in preliminary tests, but it is projected that a more extensive study is required in this subject.

It is worth noting that some touch gestures can be composed of more than one touch instance (e.g. tickles) so, it could happen that instead of detecting one gesture, the system detects several gestures in consecutive times. Therefore, to achieve a more stable output (e.g. several tickles grouped together), a 500 ms extra acquisition time begins when SNR_c drops below SNR_τ to consider the end the gesture (see Fig. 4). If this extra time expires without detecting touch activity, then the touch is considered as finished and the maximum, minimum, average, and range (max - min) values of each feature are computed (except for the duration). Finally, each script sends its piece of the gesture via ROS-topic to the next phase.

4.3 Instance creation

Implementing and managing multiple microphones simultaneously raises many challenges. The most important challenge is related to the detection of the touch gesture; that is, establishing its beginning and end. It needs to be

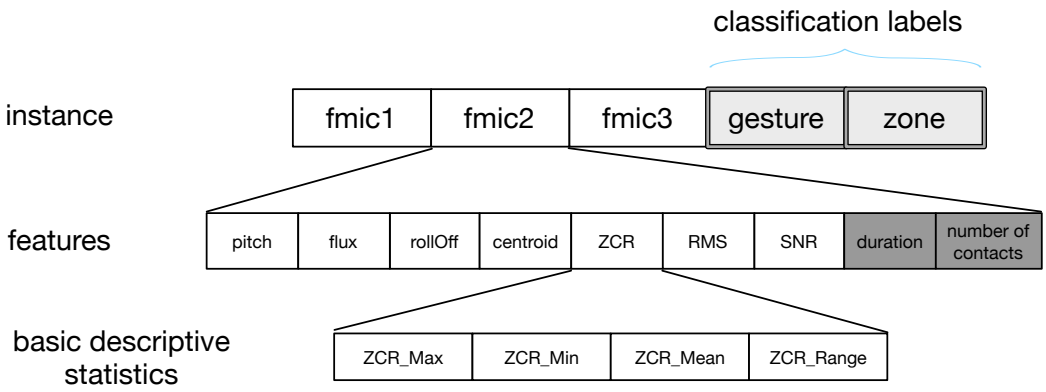


Fig. 5. One touch gesture is stored as a dataset instance. An instance is composed of several input features corresponding to each microphone, besides the classification labels (kind of gesture and zone). Additionally, every feature is composed of four statistics: max, min, mean, and range.

considered that the robotic platforms in which the contact microphones have been installed do not have physically isolated areas. Therefore, it is expected that a touch gesture executed on one of the body parts may activate multiple microphones and not just the closest, as described in Section 4.1. This paper aims to demonstrate that these combinations bring diversity to the samples, which allows the classification to be more precise.

This phase gathers the features from all of the active receivers under the same touch gesture event. For this reason, the IC phase needs to be able to coordinate and synchronise the responses of the microphones. When one FED script establishes the beginning of a gesture, the IC node checks how many receivers have perceived the gesture within the same time period. In this regard, the system starts recording data from each microphone detecting the contact (a delay of miliseconds may apear due to the sound transmission). The system then waits until each and every microphone involved in the interaction reports the contact has ended. This helps to prevent fractioning a long gesture or a contact overlap both due to false positives. Once the current gesture ends, the IC node creates an instance with the data gathered by each FED script. This instance will represent the touch gesture event, and it will be composed by the readings of each microphone, whether or not it detected activity. In case a microphone was not activated, the node will fill its corresponding values within the instance with zeroes. An instance will follow this pattern within the classification file: $I = (fmic_1, fmic_2, \ldots, fmic_n)$ where n is the number of contact microphones (three receivers in this case). Each sub-set $fmic_i$ is defined by $fmic_i = (feature_1, feature_2, \ldots, feature_m)$ where m is the number of features computed (see Fig. 5.)

Up to this point, the system is designed to create unlabelled individual instances from gesture events received by the contact microphones. The next

step is to record instances from subsequent events and label them in order to create a complete dataset with labelled instances. Once the dataset is ready, it will serve as a training set for different machine learning algorithms.

Each training instance is formed by an unlabelled instance (composed by the features gathered in previous phases) and one or two labels; that is, $I = (F, l_{gl})$, where the class labels l_{gl} in this case are the name of the touch gesture and its location. The complete dataset D, composed by a set of labelled instances follows the next structure:

$$D = \{I_1, \ldots, I_m\} \tag{2}$$

where m is the number of training instances of the dataset.

4.4 Touch classification and localisation using machine learning

This phase is designed to check if the information extracted by multiple contact microphones and processed through our software is good enough to identify the kind of gesture that is performed by the user and the contact zone on the robot's surface. If the system yields positive results, then we plan to design an online classification phase for future implementations.

As discussed in Section 2, few works have raised the idea of locating and recognising touch gestures at the same time. In this approach, we propose to solve this problem by using machine learning algorithms. Because each instance of the dataset has two different labels or categories, the problem can be solved using two types of machine learning algorithms: Multiclass and Multitarget algorithms.

(1) *Multi-class approach:* In our case, we use multi-class algorithms to classify instances in one class from among the total set of possibilities, such as taps, strokes, slaps, etc. Multi-class algorithms use two independent classifiers to recognise the gesture and to identify the part of the robot touched. In this first approach, we have employed the third-party Weka framework (Holmes et al., 1994), which, by default, integrates 82 classifiers [11]. It also enables the incorporation of new classifiers (third-parties classifiers).

[11] A complete list of the classifiers available in Weka can be found here: `http://weka.sourceforge.net/doc.dev/weka/classifiers/Classifier.html`

In our tests, we have included the whole set of default algorithms integrated in Weka, and other 44 third-party algorithms developed by the community (see the complete list of classifiers added to Weka in Appendix A). Weka's algorithms can be categorised within the most common families of machine learning classifiers, such as: meta-classifiers (which include several single classifiers), decision trees, rule-based classifiers, fuzzy classifiers, neural network and deep learning based, Bayesian, nearest neighbours, and support vector machines.

(2) *Multi-target approach:* The other learning approach uses *multi-target* algorithms. In this type of learning, a single classifier is able to simultaneously recognise the type of gesture and its location (Bielza et al., 2011; Appice & Dzeroski, 2007). In multi-target classification, the model can find dependencies between the different classes—in our case, dependencies between the kind of gesture and the place where it is performed.

In this approach, we have used the third-party framework, known as Meka (Read et al., 2016; Kavitha C.R, 2016). Meka is an extension of Weka, and is specially designed for *multi-target* scenarios. It contains all of the basic problem transformation methods, advanced methods and many varieties of classifier chains. By default, Meka integrates several kinds of meta-classifiers [12], such as: Binary Relevance (BR), Classifier Chains (CC), Classifier Trellis (CT), Label Combination (LC), Ensembles of Pruned Sets (EPS), Ensembles of Classifier Chains (ECC), and Bayesian Chain Classifier (BCC). An updated list of classifiers available in Meka can be found here: `http://meka.sourceforge.net/methods.html`. In the same way as Weka, Meka can be extended with third-party algorithms.

5 Methods

This section describes the set of gestures selected in this work. These gestures are composed of a series of datasets that are built after the users have interacted with the robots. This information was then used to train the classifiers and, through a series of metrics, assess their performance.

[12] The meta-classifiers are able to join several single classifiers. In the case of Meka, these meta-classifiers, "under-the-hood", use the Weka classifiers.

Table 2
Characterization of the touch gestures employed. The last column shows an example of how each gesture can be performed.

Gesture	Contact Area	Contact Intensity	Duration
Stroke	med-large	low	med-long
Tickle	med	med	med-long
Tap	small	low	short
Slap	small	high	short

5.1 Selecting the set of gestures

Previous studies have proposed different sets of gestures to be recognized during HRI, emphasizing the relevance of the non-verbal communication in this kind of interactions (Tang et al., 2015; Madeo et al., 2016). More specifically in touch interaction, Yohanan & MacLean (2012) presented a set of 30 gestures extracted from the social psychology and human–animal interaction literature (e.g. kiss, massage, pat, pinch, hit, grab, nuzzle, among others). Similarly, Altun & MacLean (2015) proposed a set of 26 gestures (e.g. press, poke, pat, nuzzle, trample, among others). However, Silvera-Tawil et al. (2011) proposed a set of just 6 gestures (i.e. tap, pat, push, stroke, scratch, and slap). In our approach, we have partially adopted the set of touch gestures proposed by Silvera, considering those more relevant for interaction with a social robot. Nevertheless, and according to Kim et al. (2010), push and pat gestures are not considered because *pat* is very similar to *tap* (especially the sound generated by both), and push does not happen often during HRI.

Table 2 offers a classification of the gestures regarding their contact area, perceived intensity and duration.

5.2 Experimental setup of the training phase

We apply our learning scheme to the location and recognition of touch gestures. This learning process starts with a training phase, which was performed by 40 different users. Because we use two social robots, the participants were separated into two groups (20 for each robot). Both training phases, one per robot, were carried out separately. The participants were always accompanied by a supervisor, and the interactions with the robots were made by one participant at the time. During those sessions, the procedure for both robots was as follows: first, each participant accessed the test area. A supervisor provided specific instructions about the robot's parts to be

touched and the set of different gestures performed during the experiments (shown in Table 3 and Table 4). The participants were also informed that the robots would provide some hints about how to perform each gesture before running each test. Additionally, the robot shows a video tutorial [13] on its tablet (Maggie has one integrated in its chest, and Mini has an external tablet next to it) of how to perform one of the gestures. The users then perform, in all areas considered, that gesture on the robot as many times as they want. Finally, after all of the gestures are explained and performed, the experiment finishes and the participant leaves the laboratory.

The video tutorials were designed to try to standardise how the users perform each gesture because touch gestures may differ between cultures or manners.

5.3 Building the datasets

All of the audio signals generated in these interactions are collected by the contact microphones and are then stored to build the dataset. Because the system proposed in this work has been integrated in two different social robots, we have collected two independent datasets [14]. Both datasets are structured using a plain text format, known as *ARFF* (*Attribute-Relation File Format*), which is compatible with machine learning frameworks such as Weka and Meka.

Given that the size of a dataset is closely related to the ability of our system to generalise and learn how to recognise the gestures, it was necessary to collect a relatively high number of instances. We have collected 3572 instances for the Maggie dataset and 2777 instances for the Mini dataset. Tables 3 and 4 present a summarized view of the number of instances per gesture and location gathered in the datasets. These numbers are the result of letting the users perform each gesture freely as many times as they wanted.

5.4 Evaluation metrics for data analysis

In traditional classification problems (e.g. multi-class), *Precision*, *Recall* and the *F-score* are some of the most commonly used evaluation criteria. *F-score*, also known as *F-measure*, is a single value metric that indicates the accuracy of a learning system, taking into account both its precision and recall. These metrics are calculated as shown in Equations 3, 4, and 5. In our case, we use

[13] The video tutorials are available online: `https://vimeo.com/channels/1426407`
[14] The datasets are available at `https://github.com/UC3MSocialRobots/PublicDataSets.git`

Table 3
Touch instances disaggregated by kind of gesture and location in the Maggie dataset

Maggie	**Head**	**Body Left**	**Body Right**	**Total**
Tap	299	300	323	922
Slap	280	262	319	861
Stroke	296	218	337	851
Tickle	328	251	359	938
Total	1203	1031	1338	3572

Table 4
Touch instances disaggregated by kind of gesture and location in the Mini dataset

Mini	**Head**	**Arm Left**	**Arm Right**	**Total**
Tap	253	269	226	748
Slap	231	246	218	695
Stroke	226	201	219	646
Tickle	238	212	238	688
Total	948	928	901	2777

the weighted F-score because it takes into account not only the F-score of each group to classify (in this case the kind of gesture and the localisation) but also the number of instances of each group.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

In our multi-class problem, we run two training phases to see which classifier works better with gesture recognition and also with gesture localisation. Therefore, we eliminate the gesture or location information from the datasets. Thus, both classifiers perform different and independent tasks and the accuracy of the whole system depends on the success rate of each of them separately. Because we are classifying two independent events, the probability of the intersection of such events is equal to the multiplication of the probabilities of each of them. Therefore, we have to multiply the F-score obtained for each classifier to get the global F-score of our system. The F-score

from each classifier is obtained using ten-fold cross-validation. Usually, five- or ten-fold cross-validation provides a good tradeoff between variance and bias when estimating the error (Breiman, 1992; Hastie et al., 2009).

Because our system encompasses the task of detecting both touch gestures and their localisation, we have decided to explore another family of classification techniques—multi-target algorithms. It is important to note that the main difference in multi-target classification is the fact that the prediction can be fully correct, partially correct (with different levels of correctness) or fully incorrect. Therefore, the evaluation of a multi-target classifier is more challenging than the evaluation of a single label classifier because none of the previously mentioned evaluation metrics capture such notion in their original form. In this case, it is appropriate to use the *Hamming-score*, as shown in Equation 6, which is defined as the proportion of the predicted correct labels ($Y_i \cap Z_i$) to the total number (predicted Y_i and actual Z_i) of labels for that instance (Godbole & Sarawagi, 2004). The overall *Hamming-score* is the average across all instances.

$$\textit{Hamming-score} = \frac{1}{n}\sum_{i=1}^{n}\frac{Y_i \cap Z_i}{Y_i \cup Z_i} \tag{6}$$

6 Results

This section presents the results for the two possible classification approaches in both robots. We have reported the results collected in four tables. Although only the top ten classifiers are included, more than a hundred classifiers have been trained and evaluated with various configurations. Each classifier was trained more than once, particularly 10 times, using different configuration parameters to find the best scores. While we are aware that finding the best configuration parameters constitutes a complex problem, known as *Combined Algorithm Selection and Hyperparameter*(CASH), this is outside the scope of this work (Marques et al., 2015).

6.1 Results for Maggie robot

The first platform of choice for testing both classifier approaches is Maggie, since is the first in which the system was installed. This section shows both multi-class and multi-target results.

Table 5
F-score in gesture recognition in Maggie robot (multi-class)

#	Classifier	Description	F-score
1	Random Forest	It consisting of many individual learners (trees). The random forest combined multiple random trees that votes on a particular outcome.	0.858
2	FURIA	It stands for **F**uzzy **U**nordered **R**ule **I**nduction **A**lgorithm. It is a type of fuzzy inference system	0.833
3	JRIP	It implements a propositional rule learner, **R**epeated **I**ncremental **P**runing to Produce Error Reduction (RIPPER)	0.804
4	SMO	Implements John Platt's sequential minimal optimisation algorithm for training a Support Vector Machine (SVM)	0.797
5	Neural Network	Neural Network implementation based on Multilayer Perceptron	0.792
6	J48	It generates a pruned or unpruned C4.5 decision tree	0.791
7	DeepLearning4J	Deep Convolutional Network implemented in Java and Weka	0.789
8	CHIRP	It is based on composite hypercubes on iterated random projections	0.789
9	IBK	K-nearest neighbors classifier	0.758
10	Navie Bayes	They are a family of probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features	0.645

6.1.1 Multi-class algorithms

In this first testing stage, we compared the performance (F-score) of the different classifiers with two instances of the Maggie dataset. First, a training phase considered all input features and just the touch gesture performed, omitting the location.

Under these conditions, the *Random Forest* classifier achieved the highest performance with an *F-score* of 0.858. These results are summarised in Table 5. The same test was conducted considering location instead of gesture as an output. In this case, several classifiers achieved high performance, as shown in Table 6.

Given that both classifiers are trained independently, the probability of a correct classification is the result of multiplying the probabilities of each classifier, considering the best classifier found for both tasks. In the case of Maggie robot, this is: 0.858 multiplied by 1, which results in 0.858.

6.1.2 Multi-target algorithms

In the second approach, we have evaluated many multi-target classifiers with the Maggie dataset. In this case, the best-performing algorithm was one again the BCC based on Random Forest, with 0.904 of Hamming-score. Table 7 shows the best 10 classifiers.

Table 6
F-score in gesture localisation in robot Maggie (multi-class)

#	Classifier	Description	F-score
1	Neural Networks	Neural Network implementation based on Multilayer Perceptron	1.000
2	SMO	Implements John Platt's sequential minimal optimisation algorithm for training a Support Vector Machine (SVM)	1.000
3	FURIA	It stands for **F**uzzy **U**nordered **R**ule **I**nduction **A**lgorithm. It is a type of fuzzy inference system	1.000
4	Naive Bayes	They are a family of probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features	1.000
5	Random Forest	It consisting of many individual learners (trees). The random forest combined multiple random trees that votes on a particular outcome.	1.000
6	IBK	K-nearest neighbors classifier	0.995
7	DeepLearning4J	Deep Convolutional Network implemented in Java and Weka	0.995
8	JRIP	It implements a propositional rule learner, **R**epeated **I**ncremental **P**runing to Produce Error Reduction (RIPPER)	0.995
9	J48	It generates a pruned or unpruned C4.5 decision tree	0.995
10	CHIRP	It is based on composite hypercubes on iterated random projections	0.987

Table 7
Hamming-score in gesture recognition and localisation together in Maggie robot (multi-target)

#	Multitarget classifier (metaclassifier and its associated classifier)	Hamming-score
1	BCC [15] based on Random Forest	0.904
2	BCC based on Simple Logistic	0.890
3	BCC based on LogitBoost	0.882
4	BCC based on SMO (SVM)	0.882
5	BCC based on Neural Networks	0.875
6	BCC based on J48	0.869
7	BCC based on PART	0.865
8	BCC based on JRIP	0.855
9	BCC based on Decision Table	0.820
10	BCC based on Naive Bayes	0.745

6.2 Results for Mini robot

This subsection describes the results of system performance in the second robot chosen for this purpose: Mini. As in the previous subsection, the results of the multi-class and multi-target algorithms are shown, respectively.

Table 8
F-score in gesture recognition in Mini robot (multi-class)

#	Classifier name	Description	F-score
1	Logistic	Multinomial logistic regression model with a ridge estimator	0.870
2	Logistic Model Trees	They are classification trees with logistic regression functions at the leaves	0.851
3	Random Forest	It consisting of many individual learners (trees). The random forest combined multiple random trees that votes on a particular outcome	0.844
4	FURIA	It stands for **F**uzzy **U**nordered **R**ule **I**nduction **A**lgorithm. It is a type of fuzzy inference system	0.832
5	SMO	Implements John Platt's sequential minimal optimisation algorithm for training a Support Vector Machine (SVM)	0. 810
6	Neural Network	Neural Network implementation based on Multilayer Perceptron	0.787
7	DeepLearning4J	Deep Convolutional Network implemented in Java and Weka	0.773
8	IBK	K-nearest neighbors classifier	0.736
9	HiperPypes	For each category a HyperPipe is constructed that contains all points of that category	0.56
10	VFDR	It learns decision rules	0.544

6.2.1 Multi-class algorithms

In the case of the multi-class approach, the results are similar to those obtained for Maggie; that is, *Logistic* achieved the best F-score, 0.870 (see Table 8). Additionally, *Logistic Model Trees* and *Random Forest* obtained competitive results with F-scores of 0.851 and 0.844, respectively. Regarding localisation, the same accuracy as Maggie is achieved 1.0 F-score (see Table 9). Thus, the whole accuracy of the system according this first approach is 0.870 F-score for Mini.

6.2.2 Multi-target algorithms

In the second approach in Mini, multi-target classifiers were also evaluated (see Table 10). In this case, *Bayesian Clasifier Chains (BCC) based on Random Forest* achieved the highest Hamming-score of 0.912.

6.2.3 Real-time operation in Mini robot

The tests run in both robots helped assessing the feasibility of the technique for touch gesture recognition and localisation. In those tests, we installed three microphones in the two social robots and built the datasets as described in Section 5.2. With that information, we trained a series of machine learning techniques to find the most suitable for our data. With this information, the

Table 9
F-score in gesture localisation in Mini robot (multi-class)

#	Classifier	Description	F-score
1	Neural Network	Neural Network implementation based on Multilayer Perceptron	1.000
2	SMO	Implements John Platt's sequential minimal optimisation algorithm for training a Support Vector Machine (SVM)	1.000
3	DeepLearning4J	Deep Convolutional Network implemented in Java and Weka	1.000
4	Random Forest	It consisting of many individual learners (trees). The random forest combined multiple random trees that votes on a particular outcome.	1.000
5	Naive Bayes	They are a family of probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features	1.000
6	CHIRP	It is based on composite hypercubes on iterated random projections	1.000
7	FURIA	It stands for **F**uzzy **U**nordered **R**ule **I**nduction **A**lgorithm. It is a type of fuzzy inference system	1.000
8	J48	It generates a pruned or unpruned C4.5 decision tree	1.000
9	NNge	Nearest-neighbor-like algorithm using non-nested generalised exemplars	1.000
10	JRip	It implements a propositional rule learner, **R**epeated **I**ncremental **P**runing to Produce Error Reduction (RIPPER)	0.994

Table 10
Hamming-score in gesture recognition and localisation together in Mini robot (multi-target)

#	Multitarget classifier (metaclassifier and its associated classifier)	Hamming-score
1	BCC [16] based on Random Forest	0.912
2	BCC based on Simple Logistic	0.900
3	BCC based on LogitBoost	0.897
4	BCC based on Neural Networks	0.893
5	BCC based on PART	0.89
6	BCC based on JRIP	0.889
7	BCC based on SMO (SVM)	0.889
8	BCC based on Naive Bayes	0.845
9	BCC based on Hoeffding Tree	0.844
10	BCC based on Decision Table	0.824

next step was to integrate a real-time version of our system in the social robot Mini.

The real-time version of the system required updates in the operation pipeline. Instead of gathering samples and storing them in datasets to train the classifiers, in this case, we selected a multi-class approach in which two Random Forest-based classifiers run in parallel, one recognising touch gestures, and the other dealing with the classification problem. The acquisition process

executed the Touch Activity Detection mechanism to record data instances, synchronizing the data from the three microphones as shown in Section 4.3. The following link shows a video of the robot Mini running the whole system in real-time: `https://vimeo.com/389963022`.

Shepard (1967)

7 Discussion

The results described in the previous section show that this proposal provides high accuracy with a less complex deployment (i.e. low amount of sensors and relatively simple installation) when compared to other systems presented in the literature (see Section 2). The first learning approach is the multi-class technique, which has obtained a high F-score for both robots. For the social robot Maggie, the global accuracy was F-score = 0.858; for Mini, competitive results were also obtained: F-score = 0.870. The second approximation uses the multi-target technique and the best result was obtained using *BCC based on Random Forest*, with a 0.904 and 0.912 Hamming-score for Maggie and Mini respectively.

Although both approaches offer high performance, their model validation metrics are not exactly equivalent: F-score for the multi-class approach and Hamming-score for multi-target algorithms. For the kind of problem presented in this work, we argue it is more appropriate to use the multi-target classification algorithms (our second approximation) because this approach avoids training one classifier for each label (location and type of gesture), which reduces the computational weight of the system in a future online-classification iteration. Multi-target algorithms also take advantage of the possible influence between the two labels to classify, which leads to better results overall.

Finally, in Table 11 we summarise the results of the similar works reviewed in Section 2 that perform touch gesture recognition. The table shows the results of the techniques using cross-validation and accuracy as a metric. Here, we present Accuracy (see Eq. 7) instead of F-score as a metric because the other works using the same metric. In our work, this result corresponds to the accuracy of gesture recognition when taking the average from the best-performing multi-class algorithm, Random Forest, in robot Maggie (accuracy 85.1%) and Logistic boost in the robot Mini (accuracy 87%). In this table, we observed that Silvera and Albawi, using a sensitive skin, achieved a lower accuracy. Muller, Hughes, and Zhou using deep learning achieved a lower accuracy too. The work of Cooney et al. obtains the best accuracy, but working with a combination of embedded optical sensors and external cameras.

Table 11
Comparison of gesture recognition using several different techniques

Studies	Accuracy	Platform	Technologies
Silvera-Tawil et al. (2014)	74%	Artificial robot arm	Pressure-sensitive robotic skins
Albawi et al. (2018)	63.7%	Artificial robotic arm	Pressure-sensitive robotic skins
Hughes et al. (2017)	61.35%	Human-animal affective robot	Pressure-sensitive robotic skins
Zhou & Du (2016)	76.1%	Human-animal affective robot	Pressure-sensitive robotic skins
Müller & Gross (2018)	74%	Socially Assistive Robot	Capacitive and pressure-array touch sensors
Cooney et al. (2012)	90.5%	Humanoid robot mock-up (foam-covered mannequin)	External cameras, built-into optical sensors
Our proposal	**86.05%**	Two social robots (irregular and rigid surfaces)	Built-into contact microphones

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Since the source codes from these works were not available to test our datasets, we extracted the performance metrics provided in the papers and used those to compare with the performance of our own proposal. Under these conditions, our system achieves competitive results using fewer sensors that are mounted into the robotic platform.

8 Conclusions

Social robots are expected to form part of our daily life, so endowing the robot with the ability to recognize different kinds of touch gestures performed by the user poses an important challenge in HRI. Consequently, this paper proposes a new touch-sensing technology in the field of social robotics that is able to detect, recognise and localise touch gestures in a whole robot shell using a few sensors. In contrast, although traditional touch sensing technologies (e.g. resistive, capacitive or piezoelectric, among others) present good durability and robustness, their spatial resolution is poor. Furthermore, they tend to provide a binary output, which just signals contact or not, which is in practice quite limited when aiming to detect touch gestures as in natural interaction. Therefore, contact microphones offer many interesting proprieties to overcome the limitations of the traditional technologies. These devices are also robust against ambient noise when detecting touch gestures in entire robot parts.

In our approach, we have demonstrated how a single sensor is enough to detect touch gestures in a whole robot part (e.g. in a robot's head). The scalability of the system as also been tested since we integrated three contact microphones in each of the robots, which allowed gesture localisation. This required a study of the sound propagation phenomenon in connected rigid parts (the inner structure and shell of the robots). This phenomenon caused that different sensors detected a touch gesture. Ideally, it was expected that the closest sensor registered the highest sound intensity, but this did not happen always as explained in detail in Section 4.1. Another interesting effect was the influence of ambient noises and how those are registered by the contact microphones. In our experiments, this rarely happened as the intensity of sounds propagating in the air lowers when changing to a solid material. Nevertheless, we acknowledge that intense ambient noises, if they could exert vibrations on the robot's shell, could be registered by the contact microphones and therefore cause false positives. This limitation, although not frequent, could be alleviated introducing traditional touch sensing technologies, such as capacitive ones.

We have successfully run our tests acquiring touch gestures from two different social robots, the robot Maggie and the robot Mini. The accuracy in gesture recognition in these robots was among the highest when compared with the literature, as shown in Table 11, taking into account that our proposal uses only 3 embarked sensors. Moreover, both robots obtained high performance in both approaches: multi-class and multi-target.

Our system has also observed several limitations, which represent future goals of the system. First, it is necessary to use a calibration phase to establish the touch activity detection phase thresholds. Second, the recognition and localization of touch gestures is currently limited to the robot's rigid parts, which makes the results unpredictable if the users touch other areas, such as the foam covered by a layer of soft fabric in Mini's torso. Consequently, we are currently working on improving the system by adding air-microphones that are integrated into the soft areas. Finally, another limitation is that the system has only been tested with one particular type of contact microphone and with only two kinds of social robots. Therefore, we are currently extending this system to the social robots that we have in development.

Acknowledgment

The research leading to these results has received funding from the projects: "Robots Sociales para Estimulación Física, Cognitiva y Afectiva de Mayores (ROSES)", funded by the Spanish "Ministerio de Ciencia, Innovación y Universidades" and from RoboCity2030-DIH-CM, Madrid Robotics Digital

Innovation Hub, S2018/NMT-4331, funded by "Programas de Actividades I+D en la Comunidad de Madrid" and cofunded by Structural Funds of the EU.

References

Albawi, S., Bayat, O., Al-Azawi, S., & Ucan, O. N. (2018). Social Touch Gesture Recognition Using Convolutional Neural Network. *Computational Intelligence and Neuroscience*, 2018, 1–10.

Alonso-Martin, F., Castro-González, Á., Gorostiza, J., & Salichs, M. A. (2013a). Multidomain Voice Activity Detection during Human-Robot Interaction. In *International Conference on Social Robotics (ICSR 2013)* (pp. 64–73). Bristol: Springer International Publishing.

Alonso-Martín, F., Gamboa-Montero, J. J., Castillo, J. C., Castro-González, Á., & Salichs, M. Á. (2017). Detecting and Classifying Human Touches in a Social Robot Through Acoustic Sensing and Machine Learning. *Sensors*, 17(5), 1138.

Alonso-Martin, F., Malfaz, M., Sequeira, J., Gorostiza, J., & Salichs, M. A. (2013b). A Multimodal Emotion Detection System during Human-Robot Interaction. *Sensors*, 13(11), 15549–15581.

Alonso-Martin, F., Ramey, A., & Salichs, M. Á. (2014). Speaker identification using three signal voice domains during human-robot interaction. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction - HRI '14* (pp. 114–115). Bielefeld (Germany): ACM Press.

Altun, K. & MacLean, K. E. (2015). Recognizing affect in human touch of a robot. *Pattern Recognition Letters*, 66, 31–40.

Appice, A. & Dzeroski, S. (2007). Stepwise induction of multi-target model trees. In *Machine Learning: ECML 2007*, volume 7 (pp. 502–509).: Springer Berlin Heidelberg.

Argall, B. D. & Billard, A. G. (2010). A survey of Tactile Human–Robot Interactions. *Robotics and Autonomous Systems*, 58(10), 1159–1176.

Bielza, C., Li, G., & Larrañaga, P. (2011). Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, 52(6), 705–727.

Breiman, L. (1992). The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. *Journal of the American Statistical Association*, 87, 738–754.

Carlson, A. (1968). *Communication systems: an introduction to signals and noise in electrical communication.* McGraw-Hill electrical and electronic engineering series.

Cho, H.-s., Ji, J., Chen, Z., Park, H., & Lee, W. (2015). Accurate Distance Estimation between Things: A Self-correcting Approach. *Open Journal of Internet Of Things (OJIOT)*, 1(2), 19–27.

Cochran, W., Cooley, J., Favin, D., Helms, H., Kaenel, R., Lang, W., Maling, G., Nelson, D., Rader, C., & Welch, P. (1967). What is the fast Fourier transform? In *Proceedings of the IEEE*, volume 55 (pp. 1664–1674).

Cooney, M. D., Nishio, S., & Ishiguro, H. (2012). Recognizing affection for a touch-based interaction with a humanoid robot. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1420–1427).: IEEE.

Firouzi, K., Nikoozadeh, A., Carver, T. E., & Khuri-Yakub, B. P. T. (2016). Lamb Wave Multitouch Ultrasonic Touchscreen. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63(12), 2174–2186.

Gallace, A. & Spence, C. (2010). The science of interpersonal touch: An overview. *Neuroscience & Biobehavioral Reviews*, 34(2), 246–259.

Godbole, S. & Sarawagi, S. (2004). Discriminative Methods for Multi-labeled Classification. In H. Dai, R. Srikant, & C. Zhang (Eds.), *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings* (pp. 22–30). Berlin, Heidelberg: Springer Berlin Heidelberg.

Gonzalez-Pacheco, V., Ramey, A., Alonso-Martin, F., Castro-Gonzalez, A., & Salichs, M. A. (2011). Maggie: A Social Robot as a Gaming Platform. *International Journal of Social Robotics*, 3(4), 371–381.

Goris, K., Saldien, J., Vanderborch, B., & Lefeber, D. (2011). Mechanical design of the Huggable robot Probo. *International Journal of Humanoid Robotics*, 08(03), 481–511.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. New York, NY: Springer.

Hertenstein, M., Holmes, R., McCullough, M., & Keltner, D. (2009). The communication of emotion via touch. In *American Psychological Association*, volume 9 (pp. 566–573).

Hertenstein, M. J., Verkamp, J. M., Kerestes, A. M., & Holmes, R. M. (2006). The communicative functions of touch in humans, nonhuman primates, and rats: a review and synthesis of the empirical research. *Genetic, social, and general psychology monographs*, 132(1), 5–94.

Holmes, G., Donkin, A., & Witten, I. (1994). WEKA: a machine learning workbench. In *Proceedings of ANZIIS '94 - Australian New Zealnd Intelligent Information Systems Conference* (pp. 357–361).: IEEE.

Hughes, D., Krauthammer, A., & Correli, N. (2017). Recognizing social touch gestures using recurrent and convolutional neural networks. In *Proceedings - IEEE International Conference on Robotics and Automation* (pp. 2315–2321).: IEEE.

Jung, M. M., Cang, X. L., Poel, M., & MacLean, K. E. (2015). Touch Challenge '15: Recognizing Social Touch Gestures. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15 (pp. 387–390). New York, NY, USA: ACM.

Kavitha C.R, M. T. (2016). A comparison of multi-label classification methods using Meka on benchmark datasets. *IJRET: International Journal of*

Research in Engineering and Technology, 5(9), 330–335.

Kim, Y. m., Koo, S. y., Lim, J. G., & Kwon, D. s. (2010). A robust online touch pattern recognition for dynamic human-robot interaction. *IEEE Transactions on Consumer Electronics*, 56(3), 1979–1987.

Lopes, P., Jota, R., & Jorge, J. A. (2011). Augmenting touch interaction through acoustic sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11* (pp.5̃3). New York, New York, USA: ACM Press.

Madeo, R. C. B., Peres, S. M., & Lima, C. A. d. M. (2016). Gesture phase segmentation using support vector machines. *Expert Systems with Applications*, 56, 100–115.

Marques, R. Z., Coutinho, L. R., Borchartt, T. B., Vale, S. B., & Silva, F. J. (2015). An Experimental Evaluation of Data Mining Algorithms Using Hyperparameter Optimization. In *2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI)* (pp. 152–156).: IEEE.

Minato, T., Yoshikawa, Y., Noda, T., Ikemoto, S., Ishiguro, H., & Asada, M. (2007). Cb2: A child robot with biomimetic body for cognitive developmental robotics. In *2007 7th IEEE-RAS International Conference on Humanoid Robots* (pp. 557–562).

Morita, T., Iwata, H., & Sugano, S. (1999). Development of human symbiotic robot: Wendy. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 4 (pp. 3183–3188 vol.4).

Müller, S. & Gross, H. M. (2018). Making a Socially Assistive Robot Companion Touch Sensitive. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10894 LNCS (pp. 476–488).: Springer, Cham.

Murray-Smith, R., Williamson, J., Hughes, S., & Quaade, T. (2008). Stane: synthesized surfaces for tactile input. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08* (pp. 1299). New York, New York, USA: ACM Press.

Nicholls, H. R. & Lee, M. H. (1989). A Survey of Robot Tactile Sensing Technology. *The International Journal of Robotics Research*, 8(3), 3–30.

Nikolovski, J.-P. (2003). Device for transmitting/receiving acoustic waves in a plate and method for making same.

Nikolovski, J. P. (2013). Moderately reverberant learning ultrasonic pinch panel. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 60(10), 2105–2120.

Paradiso, J. & Checka, N. (2002). Passive acoustic sensing for tracking knocks atop large interactive displays. In *Proceedings of IEEE Sensors*, volume 1 (pp. 521–527).: IEEE.

Read, J., Reutemann, P., Pfahringer, B., & Holmes, G. (2016). Meka: A Multi-label/Multi-target Extension to Weka. *Journal of Machine Learning Research*, 17, 1–5.

Robinson, S., Rajput, N., Jones, M., Jain, A., Sahay, S., & Nanavati, A. (2011). TapBack. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11* (pp. 2733). New York, New York, USA: ACM Press.

Sabanovic, S., Bennett, C. C., Chang, W.-L., & Huber, L. (2013). Paro robot affects diverse interaction modalities in group sensory therapy for older adults with dementia. In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on* (pp. 1–6).: IEEE.

Salichs, M., Barber, R., Khamis, A., Malfaz, M., Gorostiza, J., Pacheco, R., Rivas, R., Corrales, A., Delgado, E., & Garcia, D. (2006). Maggie: A Robotic Platform for Human-Robot Social Interaction. In IEEE (Ed.), *2006 IEEE Conference on Robotics, Automation and Mechatronics* (pp. 1–7). Bangkok: IEEE.

Salichs San Jose, E., Castro-González, Á., Malfáz, M., & Salichs, M. Á. (2016). Mini: a social assistive robot for people with mild cognitive impairment. In M. Heerink, B. Vandenborght, J. Albo Canals, A. Barco Martelo, C. Datta, M. Sheutz, C. Gustafsson, C. Huijnen, & J. Broekens (Eds.), *New Friends 2016: The 2nd International Conference on Social Robots in Therapy and Education* (pp. 31–32). Barcelona: OmniaScience.

Schmid, A. J., Hoffmann, M., & Woern, H. (2007). A Tactile Language for Intuitive Human-robot Communication. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, ICMI '07 (pp. 58–65). New York, NY, USA: ACM.

Sharkey, A. & Wood, N. (2014). The Paro seal robot: Demeaning or enabling? *AISB 2014 - 50th Annual Convention of the AISB*, (pp.5̃).

Shepard, R. N. (1967). Recognition memory for words, sentences, and pictures. *Journal of Verbal Learning and Verbal Behavior*, 6(1), 156–163.

Silvera-Tawil, D., Rye, D., & Velonaki, M. (2011). Touch modality interpretation for an EIT-based sensitive skin. In *2011 IEEE International Conference on Robotics and Automation* (pp. 3770–3776).: IEEE.

Silvera-Tawil, D., Rye, D., & Velonaki, M. (2014). Interpretation of social touch on an artificial arm covered with an EIT-based sensitive skin. *International Journal of Social Robotics,*, 6(4), 489–505.

Stiehl, W. D., Lieberman, J., Breazeal, C., Basel, L., Lalla, L., & Wolf, M. (2005). Design of a therapeutic robotic companion for relational, affective touch. In *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.* (pp. 408–415).

Tang, D., Yusuf, B., Botzheim, J., Kubota, N., & Chan, C. S. (2015). A novel multimodal communication framework using robot partner for aging population. *Expert Systems with Applications*, 42(9), 4540–4555.

Thomson, W. T. (1950). Transmission of Elastic Waves through a Stratified Solid Medium. *Journal of Applied Physics*, 21(2), 89–93.

Tucker, D. (1966). *Applied underwater acoustics.* Pergamon Press.

Walker, G. (2012). A review of technologies for sensing contact location on the surface of a display. *Journal of the Society for Information Display*, 20(8),

413–440.

Wang, Y., Cang, S., & Yu, H. (2019). A survey on wearable sensor modality centred human activity recognition in health care. *Expert Systems with Applications*.

Wilhelm, F. H., Kochar, A. S., Roth, W. T., & Gross, J. J. (2001). Social anxiety and response to touch: incongruence between self-evaluative and physiological reactions. *Biological Psychology*, 58(3), 181 – 202.

Yohanan, S. & MacLean, K. E. (2012). The Role of Affective Touch in Human-Robot Interaction: Human Intent and Expectations in Touching the Haptic Creature. *International Journal of Social Robotics*, 4(2), 163–180.

Zhou, N. & Du, J. (2016). Recognition of social touch gestures using 3D convolutional neural networks. In *Communications in Computer and Information Science*, volume 662 (pp. 164–173).: Springer, Singapore.

A Third-Party Weka Classifiers Employed

Name	Family	Developed by	Available on
EBMC	Bayesian	A. Lopez Pineda	https://github.com/arturolp/ebmc-weka
Discriminant Analysis	Funtions	Eibe Frank	http://weka.sourceforge.net/doc.packages/discriminantAnalysis
Complement Naive Bayes	Bayesian	Ashraf M. Kibriya	http://weka.sourceforge.net/doc.packages/complementNaiveBayes
IBKLG	K-Nearest neighbor	S. Sreenivasamurthy	https://github.com/sheshas/IBkLG
Alternating Decision Trees	Decision Trees	R. Kirkby et al.	http://weka.sourceforge.net/doc.packages/alternatingDecisionTrees
HMM	Hidden Markov Model	Marco Gillies	http://www.doc.gold.ac.uk/~mas02mg/software/hmmweka/index.html
Multilayer Perceptrons	Neural Network	Eibe Frank	http://weka.sourceforge.net/doc.packages/multiLayerPerceptrons
CHIRP	Hypercubes	Leland Wilkinson	http://www.cs.uic.edu/~tdang/file/CHIRP-KDD.pdf
AnDE	Bayesian	Nayyar Zaidi	http://weka.sourceforge.net/packageMetaData/AnDE/index.html
Ordinal Learning Method	Metaclassifier	TriDat Tran	http://weka.sourceforge.net/doc.packages/ordinalLearningMethod
Grid Search	Metaclassifier	B. Pfahringer et al.	http://weka.sourceforge.net/doc.packages/gridSearch
AutoWeka	Metaclassifier	Lars Kotthoff et al.	https://github.com/automl/autoweka
Ridor	Rules	Xin Xu	http://weka.sourceforge.net/doc.packages/ridor
Threshold Selector	Metaclassifier	Eibe Frank	http://weka.sourceforge.net/doc.packages/thresholdSelector
ExtraTrees	Decision Trees	Eibe Frank	http://weka.sourceforge.net/doc.packages/extraTrees
LibLinear	Large Linear Classification (funtions)	B. Waldvogel	http://liblinear.bwaldvogel.de/
SPegasos	SVM	Mark Hall	http://weka.sourceforge.net/doc.packages/SPegasos
Clojure Classifier	Funtions	Mark Hall	http://weka.sourceforge.net/doc.packages/clojureClassifier
SimpleCART	Decision Trees	Haijian Shi	http://weka.sourceforge.net/doc.packages/simpleCART
Conjuntive Rule	Rules	Xin XU	http://weka.sourceforge.net/doc.packages/conjunctiveRule
DTNB	Bayesian	Mark Hall et al.	http://weka.sourceforge.net/doc.packages/DTNB
J48 Consolidated	C4.5 decision tree	J. M. Perez	http://www.aldapa.eus
Lazy Associative Classifier	Rules	Gesse Dafe et al.	https://code.google.com/archive/p/machine-learning-dcc-ufmg/wikis/LACLazyAssociativeAlgorithmCpp.wiki
DeepLearning4J	Deep Learning	C. Beckham et al.	http://weka.sourceforge.net/doc.packages/wekaDeeplearning4j
HyperPipes	HyperPipes	Len Trigg et al.	http://weka.sourceforge.net/doc.packages/hyperPipes
J48Graft	C4.5 decision tree	J. Boughton	http://weka.sourceforge.net/doc.packages/J48graft

Name	Family	Developed by	Available on
Lazy Bayesian Rules Classifier	Bayesian	Zhihai Wang	http://weka.sourceforge.net/doc.stable/weka/classifiers/lazy/LBR.html
Hidden Naive Bayes classifier	Bayesian	H. Zhang	http://weka.sourceforge.net/doc.packages/hiddenNaiveBayes
Dagging meta-classifier	Metaclasifier	B. Pfahringer et al.	http://weka.sourceforge.net/doc.packages/dagging
Multilayer-PerceptronCS	Neural Networks	Ben Fowler	http://weka.sourceforge.net/doc.packages/multilayerPerceptronCS
Winnow and Balanced Winnow Classifier	Funtions	J. Lindgren	http://weka.sourceforge.net/doc.packages/winnow
Nearest-neighbor-like Classifier	k-nearest neighbors	Brent Martin	http://weka.sourceforge.net/doc.packages/NNge
Naive Bayes Tree	Bayesian	Mark Hall	http://weka.sourceforge.net/doc.packages/naiveBayesTree
Kernel Logistic Regression	Funtions	Eibe Frank	http://weka.sourceforge.net/doc.packages/kernelLogisticRegression
LibSVM	SVM	FracPete	https://www.csie.ntu.edu.tw/~cjlin/libsvm/
Fuzzy Unordered Rule Induction	Fuzzy	J. C. Hühn	http://weka.sourceforge.net/doc.packages/fuzzyUnorderedRuleInduction
Best First Tree	Decision Tree	Haijian Shi	http://weka.sourceforge.net/doc.packages/bestFirstTree
MetaCost meta-classifier	Metaclassifier	Len Trigg	http://weka.sourceforge.net/doc.packages/metaCost
Voting Feature Intervals Classifier	Voting	Mark Hall	http://weka.sourceforge.net/doc.packages/votingFeatureIntervals
ordinal Stochastic Dominance	Ordinal Stochastic Dominance Learner	Stijn Lievens	http://weka.sourceforge.net/doc.packages/ordinalStochasticDominance
RBFNetwork	Funtions	Eibe Frank	http://weka.sourceforge.net/doc.packages/RBFNetwork
MODLEM rule algorithm	Decision Trees	S. Wojciechowski	https://sourceforge.net/projects/modlem/
The Fuzzy Lattice Reasoning Classifier	Fuzzy	I. N. Athanasiadis	http://weka.sourceforge.net/doc.packages/fuzzyLaticeReasoning
Functional Trees	Decision trees	C. Ferreira	http://weka.sourceforge.net/doc.packages/functionalTrees