Acoustic-based Smart Tactile Sensing in Social Robots

by

Juan José Gamboa Montero

A dissertation submitted by in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Electrical Engineering, Electronics and Automation

Universidad Carlos III de Madrid

Advisor(s):

Fernando Alonso Martín
José Carlos Castillo Montoya

Tutor:

Miguel Ángel Salichs Sánchez-Caballero

January 2023

A toda mi familia.

Acknowledgements

AFTER five years of hard work, victories and some defeats, I can only feel gratitude. Gratitude for the opportunity, for the help, for the good times, and ironically also for the bad times when I have been able to get the best out of myself. Thanks for the laughs, the jokes, and the smiles we have shared during this journey. I want to thank everyone for all the support, and I would like to apologise in advance for switching to my native Spanish to show my gratitude.

En primer lugar tengo que agradecer por todo el trabajo a mis directores, Fernando y José Carlos, y a mi tutor, Miguel Ángel, por toda su ayuda. Sin duda alguna, de no ser por ellos, todo este trabajo no podría haber sido posible. A Fernando quiero agradecerle su cercanía, sus ideas y todo ese dinamismo tremendamente contagioso que ha permitido que este trabajo tuviera nuevas opciones cuando parecía que habíamos tocado hueso. Quiero agradecerle las charlas, los consejos y los paseos juntos divagando sobre cualquier tema que se nos viniese a la cabeza. A José Carlos quiero darle las gracias por todo su apoyo, serenidad y buen juicio, por estar ahí siempre que lo he necesitado, y, sobre todo, por estar ahí cuando no sabía que le necesitaba. Quiero agradecerle tanto por su paciencia conmigo y su sensatez, como por los buenos momentos y las risas juntos. A Miguel Ángel tengo que agradecerle toda su vasta experiencia, por los consejos en momentos críticos del trabajo y por su guía y buen criterio. Él ha sido el que me ha brindado esta enorme oportunidad y, sobre todo, quiero agradecerle por la estabilidad y la seguridad que siempre me ha dado. Por último, aunque no forman parte 'oficial' de este trabajo, tengo que dar las gracias a María y a Álvaro también, por todo su apoyo. Siempre han estado ahí cuando he necesitado una mano o simplemente charlar, y también son una parte importante de este trabajo.

I would like to switch back to English in just this paragraph to thank the Intelligent Robots and Systems Group (IRSg) for all the help they gave me during my stay in Lisbon. It was an amazing experience for me I will always remember it. I want to thank Pedro Lima for the opportunity to spend my stay there, and, above all, I would like to thank Meysam Basiri for his support and mentorship. Muito obrigado!

Ahora quiero hablar de mi familia. Sin ellos, definitivamente nada de lo que he realizado hasta ahora habría sido posible. Siempre han estado entre bambalinas para levantarme cuando me he caído y para abrazarme cuando más lo he necesitado. Quiero agradecer a mis padres, a Juanjo y a Begoña, por todo el amor, el apoyo y la guía que han volcado sobre mí durante todo este tiempo. Siempre me han dado lo mejor de sí mismos y no tengo palabras para describir todo lo que les agradezco lo que han hecho y siguen haciendo por mí. A Guille, mi hermano, quiero darle las gracias también por todo su cariño, su ayuda, las risas que siempre compartimos y, sin duda alguna, por aguantarme todo este tiempo. Quiero darles las gracias a mis abuelos, a Mariano, a Julia, a Antonio y a Mercedes, por no darme otra cosa que no fuese su amor, por todo el orgullo que siempre me han demostrado y por apoyarme incondicionalmente en cada decisión que he tomado hasta ahora. No me quiero olvidar de mis tíos y de mis primos, haciendo una mención especial a mi prima Paloma, por todo el cariño que siempre me han mostrado. Por último, quiero incluir en este agradecimiento familiar a Julián y a Fátima, porque siempre han estado ahí cuando lo he necesitado y para mí siempre serán mis tíos.

Quiero también darle las gracias al Grupo de Robótica Social de la Universidad Carlos III, donde he desarrollado mi trabajo. Quiero darles las gracias por todos los momentos juntos a Jony, Sergio, Elena, Sara Carrasco, Javier Sevilla, Juan, Carlos, Carlos Manuel, Abel, Javi Burguete, Esther, Jesús, Marcos y Quique. Han sido y son parte de un grupo increíble de gente, muchas gracias por toda vuestra ayuda. Quiero darles las gracias también a Fernando San Deogracias, Ángela, Raúl Nouredine, Sonia y a Edu, por todo su trabajo y su ayuda siempre que lo he necesitado. Tampoco me puedo olvidar de mis compañeros de despacho, Jorge, David Estévez, Juanmi, David Martín, Pablo, Miguel, Lisbeth, Luis y David Serrano. Muchas gracias por los cafés y las charlas juntos. En este sentido, quiero extender también mi agradecimiento a todo el Departamento de Sistemas y Automática de la UC3M, por haberme ayudado en todo momento. Muchísimas gracias a ellos y a todos lo que en general han participado en mi trabajo, ya sea probando el sistema o participando en los experimentos. Por último, quiero darle las gracias de todo corazón a alguien que ha sido mi compañera de trabajo, mi compañera de despacho y es una de mis mejores amigas, Sara. Muchas gracias por estar ahí durante estos años, por los buenos y los malos momentos (que nos han hecho más fuertes) que hemos vivido y por apoyarme siempre.

Muchísimas gracias también a todos mis amigos, a los que habéis estado a mi lado desde bien chiquititos, Jesús, Héctor, Mario y Borja. Gracias por todas las tardes de juegos, de risas y también por los momentos más difíciles, en los que un rato juntos lo arreglaba todo. Gracias a mi grupo de poscomunión, a Oscar, David, Davicillo, Luis, Guille, David Gómez, a Lorena, María, Sara, Silvia, Noelia y a Sonia. Muchas gracias por todos los ratos juntos y por hacer un hueco para poder vernos, aunque a veces sea difícil. Gracias también a ese grupo de gente increíble que conocí en el grado y en el máster, gracias a Nuria, Sílvia, David, Fran y Juan. Espero poder devolveros todo lo que me habéis dado hasta ahora.

Por último, quiero darte miles de gracias, Silvia, por ser mi Luna, mi compañera y mi mejor amiga durante todo este viaje. Gracias por ser la muleta en la que me he apoyado en todo este tiempo, por completarme como persona y, más que nada, por quererme y soportarme tal y como soy.

Este trabajo va dedicado a todos vosotros. Gracias de todo corazón.

Published and submitted content

Journal

1. **Gamboa-Montero, J. J.**, Alonso-Martin, F., Marques-Villarroya, S., Sequeira, J., & Salichs, M. A. (2023). "Asynchronous federated learning system for human–robot touch interaction". *Expert Systems with Applications*, 211, 118510.
**This item is wholly included in the thesis, in Chapter 7. The material from this source included in this thesis is not singled out with typographic means and references.*

2. **Gamboa-Montero, J. J.**, Alonso-Martin, F., Castillo, J. C., Malfaz, M., & Salichs, M. A. (2020). "Detecting, locating and recognising human touches in social robots with contact microphones". *Engineering Applications of Artificial Intelligence*, 92, 103670. Q1.
**This item is wholly included in the thesis, in Chapters 4 and 5. The material from this source included in this thesis is not singled out with typographic means and references.*

3. 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. Q2.
**This item is wholly included in the thesis, in Chapters 2, 4 and 5. The material from this source included in this thesis is not singled out with typographic means and references..*

4. Salichs, M. A., Castro-González, A., Salichs, E., Fernández-Rodicio, E., Maroto, M., **Gamboa-Montero, J. J.**, Marques-Villarroya, S., Castillo, J. C., Malfaz, M., Alonso, F. (2020). "Mini: A New Social Robot for the Elderly". *International Journal of Social Robotics*, 12, 1231-1249. Q1.
**This item is partially included in the thesis, in Chapter 3. The material from this source included in this thesis is not singled out with typographic means and references..*

Conference

1. **Gamboa-Montero, J. J.**, Basiri, M., Marques-Villarroya, S., Castillo, J. C., & Salichs, M. Á. (2022), "Real-Time Acoustic Touch Localization in Human-Robot Interaction based on Steered Response Power", *2022 IEEE International Conference on Development and Learning (ICDL)*, pp. 231-237.
   **This item is wholly included in the thesis, in Chapters 4 and 5. The material from this source included in this thesis is not singled out with typographic means and references.*

2. Marques-Villarroya, S., **Gamboa-Montero, J. J.**, Jumela-Yedra, C., Castillo, J. C., & Salichs, M. Á. (2022), "Affect Display Recognition through Tactile and Visual Stimuli in a Social Robot", *In International Conference on Social Robotics. Springer, Cham.*
   **This item is wholly included in the thesis, in Chapter 6. The material from this source included in this thesis is not singled out with typographic means and references.*
   ***This work was awarded as Best Conference Paper.*

3. Alonso-Martín, F., Castillo, J. C., **Gamboa-Montero, J. J.**, & Salichs, M. Á. (2017). "Acoustic Sensing for Touch Recognition in a Social Robot". *In Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (HRI '17).* Association for Computing Machinery, pp. 65–66.
   **This item is partially included in the thesis, in Chapter 4. The material from this source included in this thesis is not singled out with typographic means and references.*

Other research merits

Journal

1. Marques-Villarroya, S., Castillo, J. C., **Gamboa-Montero, J. J.**, Sevilla-Salcedo, J., & Salichs, M. A. (2022). A Bio-Inspired Endogenous Attention-Based Architecture for a Social Robot. Sensors, 22(14), 5248.

Conference

1. Marques-Villarroya S., **Gamboa-Montero, J. J.**, Bernardino, A., Maroto-Gómez, M., Castillo, J. C. & Salichs, M. Á. (2022). "Real-time Engagement Detection from Facial Features", 2022 IEEE International Conference on Development and Learning (ICDL), pp. 231-237.

2. Alonso-Martín, F., Carrasco-Martínez, S., **Gamboa-Montero, J. J.**, Fernández-Rodicio, E., & Salichs, M. Á. (2022), "Introducing Psychology Strategies to increase Engagement on Social Robots", *In International Conference on Social Robotics. Springer, Cham*.

3. Menendez, C., Marques-Villarroya, S., Castillo, J.C., **Gamboa-Montero, J. J.** & Salichs, M.A. (2021). A Computer Vision-Based System for a Tangram Game in a Social Robot. In: Novais, P., Vercelli, G., Larriba-Pey, J.L., Herrera, F., Chamoso, P. (eds) Ambient Intelligence – Software and Applications . ISAmI 2020. Advances in Intelligent Systems and Computing, vol 1239. Springer, Cham.

4. Fernández-Rodicio, E., Castro-González, Á., **Gamboa-Montero, J.J.** & Salichs, M.A. (2020). Perception of a Social Robot's Mood Based on Different Types of Motions and Coloured Heart. In: Social Robotics. ICSR 2020. Lecture Notes in Computer Science(), vol 12483. Springer, Cham.

Resumen

El sentido del tacto es un componente crucial de la interacción social humana y es único entre los cinco sentidos. Como único sentido proximal, el tacto requiere un contacto físico cercano o directo para registrar la información. Este hecho convierte al tacto en una modalidad de interacción llena de posibilidades en cuanto a comunicación social. A través del tacto, podemos conocer la intención de la otra persona y comunicar emociones. De esta idea surge el concepto de *social touch* o tacto social como el acto de tocar a otra persona en un contexto social. Puede servir para diversos fines, como saludar, mostrar afecto, persuadir y regular el bienestar emocional y físico.

Recientemente, el número de personas que interactúan con sistemas y agentes artificiales ha aumentado, principalmente debido al auge de los dispositivos tecnológicos, como los *smartphones* o los altavoces inteligentes. A pesar del auge de estos dispositivos, sus capacidades de interacción son limitadas. Para paliar este problema, los recientes avances en robótica social han mejorado las posibilidades de interacción para que los agentes funcionen de forma más fluida y sean más útiles. En este sentido, los robots sociales están diseñados para facilitar interacciones naturales entre humanos y agentes artificiales. El sentido del tacto en este contexto se revela como un vehículo natural que puede mejorar la Human-Robot Interaction (HRI) debido a su relevancia comunicativa en entornos sociales. Además de esto, para un robot social, la relación entre el tacto social y su aspecto es directa, al disponer de un cuerpo físico para aplicar o recibir toques.

Desde un punto de vista técnico, los sistemas de detección táctil han sido objeto recientemente de nuevas investigaciones, sobre todo dedicado a comprender este sentido para crear sistemas inteligentes que puedan mejorar la vida de las personas. En este punto, los robots sociales se han convertido en dispositivos muy populares que incluyen tecnologías para la detección táctil. Esto está motivado por el hecho de que un robot puede esperada o inesperadamente tener contacto físico con una persona, lo que puede mejorar o interferir en la ejecución de sus comportamientos. Por tanto, el sentido del tacto se antoja necesario para el desarrollo de aplicaciones robóticas. Algunos métodos incluyen el reconocimiento de gestos táctiles, aunque a menudo exigen importantes despliegues de hardware que requieren de múltiples sensores.

Además, la fiabilidad de estas tecnologías de detección es limitada, ya que la mayoría de ellas siguen teniendo problemas tales como falsos positivos o tasas de reconocimiento bajas. La detección acústica, en este sentido, puede proporcionar un conjunto de características capaces de paliar las deficiencias anteriores. A pesar de que se trata de una tecnología utilizada en diversos campos de investigación, aún no se ha integrado en la interacción táctil entre humanos y robots.

Por ello, en este trabajo proponemos el sistema Acoustic Touch Recognition (ATR), un sistema inteligente de detección táctil (*smart tactile sensing system*) basado en la detección acústica y diseñado para mejorar la interacción social humano-robot. Nuestro sistema está desarrollado para clasificar gestos táctiles y localizar su origen. Además de esto, se ha integrado en plataformas robóticas sociales y se ha probado en aplicaciones reales con éxito. Nuestra propuesta se ha enfocado desde dos puntos de vista: uno técnico y otro relacionado con el tacto social. Por un lado, la propuesta tiene una motivación técnica centrada en conseguir un sistema táctil rentable, modular y portátil. Para ello, en este trabajo se ha explorado el campo de las tecnologías de detección táctil, los sistemas inteligentes de detección táctil y su aplicación en HRI. Por otro lado, parte de la investigación se centra en el impacto afectivo del tacto social durante la interacción humano-robot, lo que ha dado lugar a dos estudios que exploran esta idea.

Abstract

THE sense of touch is a crucial component of human social interaction and is unique among the five senses. As the only proximal sense, touch requires close or direct physical contact to register information. This fact makes touch an interaction modality full of possibilities regarding social communication. Through touch, we are able to ascertain the other person's intention and communicate emotions. From this idea emerges the concept of *social touch* as the act of touching another person in a social context. It can serve various purposes, such as greeting, showing affection, persuasion, and regulating emotional and physical well-being.

Recently, the number of people interacting with artificial systems and agents has increased, mainly due to the rise of technological devices, such as smartphones or smart speakers. Still, these devices are limited in their interaction capabilities. To deal with this issue, recent developments in social robotics have improved the interaction possibilities to make agents more seamless and useful. In this sense, social robots are designed to facilitate natural interactions between humans and artificial agents. In this context, the sense of touch is revealed as a natural interaction vehicle that can improve HRI due to its communicative relevance. Moreover, for a social robot, the relationship between social touch and its embodiment is direct, having a physical body to apply or receive touches.

From a technical standpoint, tactile sensing systems have recently been the subject of further research, mostly devoted to comprehending this sense to create intelligent systems that can improve people's lives. Currently, social robots are popular devices that include technologies for touch sensing. This is motivated by the fact that robots may encounter expected or unexpected physical contact with humans, which can either enhance or interfere with the execution of their behaviours. There is, therefore, a need to detect human touch in robot applications. Some methods even include touch-gesture recognition, although they often require significant hardware deployments primarily that require multiple sensors. Additionally, the dependability of those sensing technologies is constrained because the majority of them still struggle with issues like false positives or poor recognition rates. Acoustic sensing, in this sense, can provide a set of features that can alleviate the aforementioned shortcomings. Even though it is a techno-

logy that has been utilised in various research fields, it has yet to be integrated into human-robot touch interaction.

Therefore, in this work, we propose the ATR system, a smart tactile sensing system based on acoustic sensing designed to improve human-robot social interaction. Our system is developed to classify touch gestures and locate their source. It is also integrated into real social robotic platforms and tested in real-world applications. Our proposal is approached from two standpoints, one technical and the other related to social touch. Firstly, the technical motivation of this work centred on achieving a cost-efficient, modular and portable tactile system. For that, we explore the fields of touch sensing technologies, smart tactile sensing systems and their application in HRI. On the other hand, part of the research is centred around the affective impact of touch during human-robot interaction, resulting in two studies exploring this idea.

Contents

List of Figures

List of Tables

List of Acronyms

ALSA Advanced Linux Sound Architecture

ANN Artificial Neural Network

API Application Programming Interface

ARFF Attribute-Relation File Format

ATR Acoustic Touch Recognition

ASD Autism Spectrum Disorder

BCC Bayesian Chain Classifier

BR Binary Relevance

CC Classifier Chains

CLI Command Line Interface

CNN Convolutional Neural Network

DOA Direction of Arrival

DC Dataset Creation ATR phase

DL4J Deep Learning for Java

DWT Discrete Wavelet Transform

EEG electroencephalogram

EIT Electrical Impedance Tomography

FE Feature Extraction ATR phase

FFT Fast Fourier Transform

FL Federated Learning

fMRI Functional Magnetic Resonance Imaging

FSR Force-sensitive Resistor

GEVA Gender and Emotion Voice Analysis

GCC Generalised Cross Correlation

GUI Graphical User Interface

GDPR General Data Protection Regulation

HAR Human Activity Recognition

HRI Human-Robot Interaction

IC Instance Creation ATR phase

IoT Internet of Things

IMI Intrinsic Motivation Inventory

LC Label Combination

LMT Logistic Model Trees

MEKA Multi-dimensional Environment for Knowledge Analysis

ML Machine Learning

MLP Multilayer Perceptron

NLP Natural Language Processing

NSR Nearest Set Replacement

OS Operating System

OSC Open Sound Control

OC Online Classification ATR phase

PCA Principal Component Analysis

PHAT Phase Transform

RF Random Forest

RMS Root Mean Square

ROS Robot Operating System

RPC Remote Procedure Call

SA Sound Signal Acquisition ATR phase

SMO Sequential Minimal Optimization

SMOTE Synthetic Minority Over-sampling Technique

SNR Signal-to-Noise Ratio

SRP Steered Response Power

SSL Sound Source Localisation

STFT Short-time Fourier Transform

STT Social Touch Technology

STS Smart Tactile Sensing

SVM Support Vector Machine

TAD Touch Activity Detection ATR phase

TC Touch Gesture Classification ATR phase

TCL Touch Gesture Classification and Localisation ATR phase

TDOA Time Difference of Arrival

TOA Time of Arrival

TOF Time of Flight

UDEV Userspace '/dev'

UES User Engagement Scale

UES-SF User Engagement Scale Short Form

VAD Voice Activity Detection

VPN Virtual Private Network

WEKA Waikato Environment for Knowledge Analysis

YAML YAML Ain't Markup Language™

Chapter 1

Introduction

Among the five senses, the sense of touch can be considered unique as it is the earliest and most basic modality to develop: it has an impact from conception, is crucial during childbirth, and continues to be essential for early childhood development through infancy [1–3]. This sense's main organ is the largest one in the human body: the skin —adults carry around 3.6 kilograms and almost two square metres of it—. The skin protects the body from extreme temperatures, toxic substances, and the sun's harmful rays by acting as an insulating and waterproof shield. It also produces vitamin D, which is necessary for turning calcium into solid bones, and antimicrobial chemicals that fight illness. More importantly, the skin serves as a massive sensor loaded with nerves that keeps the brain connected to the environment. Besides, touch is the only proximal sense, requiring close or direct physical contact to register information [4]. This fact makes touch an interaction modality full of possibilities regarding social communication [5].

1.1. Motivation

The sense of touch is an essential aspect of social interaction among humans [6]. Several works focus on using touch as a valid modality to ascertain the user's intention and claim the evidence of touch as a powerful way of communicating emotions. The role that the sense of touch plays in emotional communication in humans and animals has been widely studied, finding relations to attachment, bonding, stress, and even memory [5, 7]. In this sense, Hertenstein [5] found that anger, fear, disgust, love, gratitude, and sympathy are easier to detect than happiness and sadness. From these authors, focused on understanding the role that touch plays in social interaction, the term *social touch* began to be coined. *Social touch* is the act of touching another person in a social context [8]. It can serve a variety of purposes, such as greeting, showing affection, persuasion, and regulating emotional and physical well-being. The sense of touch is important for bonding between a child and its mother and has a significant role in social interactions later in life. Researchers have proposed the *social touch hypothesis* [9], which suggests that certain nerve fibres, called C-tactile (CT) afferent fibres, may act as a filter in conjunction with other mechanoreceptors to determine whether a touch has social relevance. CT afferents are sensitive to caressing touches, which are significant in human affiliative interactions [10, 11].

In recent years, the use of technology such as smartphones and smart speakers has led to a rise in the number of people interacting with artificial systems and agents, autonomous systems that make decisions based on the stimuli gathered from the environment, the users, and their experiences. As the use of artificial agents becomes more common, the latest advances in this field have been focused on improving these interactions to make them more seamless and helpful, leading to the development of social robots: devices designed to facilitate natural interactions between humans and artificial agents. Cynthia Breazeal [12] defined a social robot as "an autonomous robot that can communicate with humans in accordance to a social model that is applied by the human observers". Bartneck et al. [13] proposed a similar definition: "Autonomous or semi-autonomous robot able to interact and communicate with humans according to the behavioural norms expected by these humans". These definitions indicate that a social robot is one of the most appropriate platforms for studying human-robot touch interaction. Furthermore, for a social robot, the relationship between social touch and its embodiment is direct, having a physical body to apply or receive touches.

In this sense, Huisman [14] proposed the concept of Social Touch Technology (STT) as the use of touch-sensing technology for social interactions, including with artificial agents and through technology-mediated communication. His research is supported by the *Media Equation theory* [15], which states that when people interact with intelligent systems, they treat them as social actors. This concept has been expanded upon in the literature by giving artificial social agents a presence in order to express emotions during user interactions. Based on this logic, Huisman stated that touches made by virtual or physical artificial agents might be interpreted as social touches by the user and, more importantly, that effects of human-human social touch might occur in human-robot interaction settings [14].

Some works have focused on studying how humans communicate their emotional state to social robots and how they expect the social robot to behave when being touched. Jung et al. [16] explored this concept in depth and expanded the definition of a social robot by stating that to achieve a much deeper level of communication within robotics beyond visual and aud-

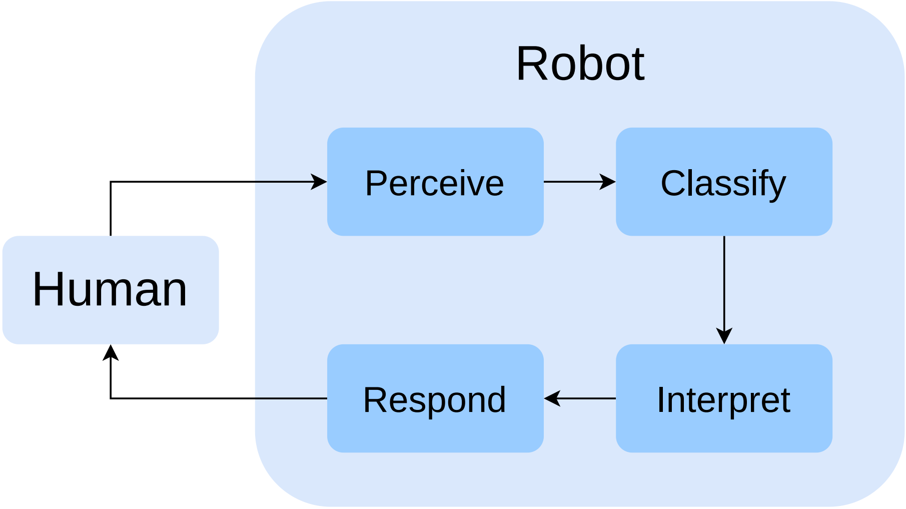


Figure 1.1: Human-robot tactile interaction scheme according to Jung [18].

itory interaction, a social robot must be able to perceive and recognise various tactile gestures. Additionally, it must comprehend these gestures and react appropriately (see Figure 1.1). The communication of affect through touch to enable a robot to understand touch gestures has also been a very relevant topic covered by the literature. In this line, Yohanan et al. [17] presented a touch dictionary of 30 items extracted from social psychology and human-animal interaction literature, identifying which ones are more likely to be used to communicate specific emotions and those which are not. Besides, the authors categorised the human's higher intents through affective touch, resulting in protective (*hold*, *hug*, *cradle*); comforting (*stroke*, *rub*, *finger idle*, and *pat*); restful (*massage*, *scratch*, and *tickle*); affectionate (*tickle*, *scratch*, *massage*, *nuzzle*, *kiss*, *rock*, *hug*, and *hold*); and playful (*lift*, *swing*, *toss*, *squeeze*, *stroke*, *rub*, *pat*, *scratch*, *massage*, and *tickle*). Their research revealed how touch is a natural interaction vehicle that can improve HRI and promote intelligent behaviour in social robots.

However, there is still work to do to improve and explore in the social human-robot touch interaction field. Shiomi et al. [19] devised a series of challenges that social touch research may face in the future. Among the most relevant ones mentioned, they suggested developing a robotic platform able to identify different kinds of social touch and using this platform to examine how variables such as age, gender, and appearance might influence touch interaction. This hypothetical robotic platform should handle social touch in real scenarios, such as learning activities or gaming contexts where touch could positively affect the user's experience.

From a technical standpoint, tactile sensing systems have been the subject of further research recently [20]. Most research on the topic has been devoted to comprehending this sense to create intelligent systems that can improve people's lives, given the significance of tactile sensing in both daily life and industry. Most robots integrate simple touch sensors only able to detect contact, whilst more sophisticated sensors that assess surface features like temperature, stiffness, and roughness can also be deployed. Most people are unaware of the countless uses for tactile sensing devices, including manual palpation and prosthetic limbs [21]. Current social robots commonly include technologies for touch sensing [22]. In these situations, robots may encounter expected or unexpected physical contact with humans, which can either enhance or interfere with the execution of their behaviours. There is, therefore, a need to detect human touch in robot applications. Some methods even include touch-gesture recognition, although they often require significant hardware deployments primarily that require multiple sensors [23]. Additionally, the dependability of those sensing technologies is constrained because the majority of them still struggle with issues like false positives or poor recognition rates.

Acoustic sensing, in this sense, can provide a set of features that can alleviate the aforementioned shortcomings. Even though it is a technology that has been utilised in various research fields, it has yet to be integrated into human-robot touch interaction. Although there are no examples in the field of social robotics, acoustic sensing has been used in numerous research fields, primarily those concerned with interactive displays, for touch detection, classification, and localization [24]. Through the years, works on this topic demonstrated the possibilities that sound-based systems offered in terms of touch recognition on solid surfaces. This is based on a physical fundamental property of sound signals since they propagate better in solids and liquids than through air. The work presented is hereby motivated by two aspects of human-robot touch interaction. Firstly a technical motivation centred on achieving a cost-efficient, modular and portable tactile system. But secondly, an essential part of the research conducted in this work revolves around the affective impact that touch interaction has during social human-robot interaction.

1.2. Objectives

In the search for a touch system to meet the challenges regarding the technical and affective aspects of tactile sensing in the social robotics field, several goals have been defined that converge into one main objective. This primary goal revolves around applying the ideas presented before regarding tactile sensing and human-robot touch interaction to **design a tactile sensing system able to improve human-robot social interaction**.

In order to achieve this major objective, we define the following set of subgoals. Since the work aims to address both the technical part and the human-robot interaction part, these subgoals are divided into three blocks, which correspond to the structure of the text: hardware, software and HRI. The first objectives have to do with the **physical design of the system**. As a perception system, it comprises sensors, data acquisition elements and, in this case, indirectly, the robotic platform where it will be integrated. The goals in this respect are as follows:

1. To design an intelligent touch system that adapts to the particularities of a social robot. Among the particularities of these platforms, we can find their curved surfaces and the combination of hard and soft materials in terms of their external appearance.

2. To integrate and evaluate the possibilities offered by acoustic sensing in an intelligent touch system. This involves exploring different technologies in terms of both sensory and data acquisition and evaluating the options available in terms of interfaces and sound cards.

3. The system should not hinder the tactile interaction with the robot; therefore, we seek to avoid superficial setups in the robotic platform. As mentioned above, social robots have curved or soft surfaces, and ensuring the best positioning of sensors is key to proper sound acquisition. Furthermore, the system should not be installed superficially to avoid altering the robot's physical appearance as much as possible. The main reason is that the appearance of a social robot is a crucial attribute to appeal to the user.

4. The complexity of the hardware deployment must be low; the aim should be, as much as possible, to avoid excessively complex ad-hoc systems or those that require a large number of sensors.

After considering the hardware elements of the system, the next step is the **design and software integration of the system**. The sub-objectives proposed in this aspect are the following:

1. As it happened with the hardware, the software deployment must also be of low complexity, prioritising that the system uses tools that avoid altering the platform's software and installing libraries or tools that, in the long term, may cause conflicts with the robot's software.

2. The system must be integrated into the software architecture present in the social robotics laboratory where the work is being developed.

3. The system must be able to identify and recognise how the user is touching the robot. That implies defining a set of different touch gestures adapted to the features of the robotic platform.

4. The system must also be able to localise the source of the touch contact.
5. The system must be open to different improvements that allow its integration not only in one platform but in multiple platforms of the same type, and the system can benefit from the tactile knowledge acquired from those platforms in order to achieve a common knowledge base.

The last facet of the system that poses subgoals is related to **human-robot interaction**. Therefore, the work presented in this document will not only focus on the technical aspects but also on the impact that the system designed and implemented throughout this work could have on the human interacting with the robotic platform. The proposed goals in this aspect are the following:

1. The system must be sufficiently modular to form part of more complex multimodal systems able to recognise higher-level affective information without requiring modifications to its design.
2. The system must function properly in a real environment and application, and therefore we rule out any purely offline integration of the system. In that sense, the system must not be perceived by the user as slow or inaccurate.
3. The introduction of the touch system must lead to a significant improvement in the user experience with the robot. That implies finding a suitable application where touch could be implemented and compared with a baseline.

Although the target of the touch sensing system, in this case, is to be integrated into a social robot, we do not want its design to be tailored specifically to these platforms to exclude the possibility of introducing it into another field of research. Therefore, the system must be designed with adaptability criteria that allow the system to be integrated into any other platform type. Although we do not specifically seek to evaluate this, it is not an objective per se; it is a secondary requirement that we intend to maintain throughout the work.

1.3. Overview of the Document

Finally, in this section, we define the structure and content of each chapter in this work. In essence, when designing the manuscript's structure regarding the system itself, we decided to start from its setup and main hardware elements to move on to the software implementation and evaluation. Afterwards, we introduced two HRI experiments that also represent two use cases of the system. Finally, we decided to finish by proposing a system enhancement focused on distributed learning. The content of each of these chapters is shown in more detail below:

- **Chapter 2:** In this chapter, we present the main works on which the proposal of this manuscript is based. It commences by presenting the concept of tactile interaction through Smart Tactile Sensing systems and focuses on the most relevant works in human-robot touch interaction. Afterwards, we explore the concept of Social Touch, its effects, and the developments and findings resulting from applying this concept to the field of HRI.

- **Chapter 3:** This chapter describes the various elements of the system setup. First, we describe the different robotic platforms involved in the development of the system. Then, we explain how the system's hardware components were integrated into these robotic platforms. Lastly, we detail the elements that compose the software architecture of the robots, where the system will also be integrated.

- **Chapter 4:** This chapter describes the system design at a software level, detailing each of its components. First, we explain the touch gesture classification block of the system. Afterwards, we detail the approaches implemented to achieve touch gesture localisation. On the one hand, we approached the problem using machine learning techniques, as in the previous block. On the other hand, we implemented sound analysis techniques to achieve a more precise localisation. Finally, we present the online integration of the system, essential for the experiments presented in Chapter 6.

- **Chapter 5:** In the experiments described in this chapter, we evaluated the different components described in the previous chapter. The system is divided into four different experiments. The first consists of a proof of concept with one acoustic receiver on a social robot to classify touch gestures. The second evaluation implements a larger number of sensing devices in order to classify both the touch gesture performed and its location. In the third, we implement sound signal analysis techniques to localise the contact's source more precisely. The last experiment evaluates the online classification module of the system.

- **Chapter 6:** The experiments from this chapter explored the application of the system to experiments related to social touch. We integrate the system into a more complex multimodal detector that combines information from touch interaction and visual face recognition to create a system capable of affect recognition. In the second experiment, we studied how actively interacting through touch with a social robot affected the user's behaviour.

- **Chapter 7:** In this chapter, we present a case study in which distributed learning is integrated into the touch system. The goal is to optimise touch system learning by training across multiple robotic platforms asynchronously. The paradigm presented in this chapter is unique in that it protects the users' privacy.

- **Chapter 8**: This dissertation ends by listing the main conclusions extracted from this work and proposing future research to improve the system in different aspects.

Chapter 2

Related Works

The work presented in this manuscript is oriented toward human-robot tactile interaction and is framed more specifically in the field of *Smart Tactile Sensing (STS) Systems* [21]. These systems receive physical contact information via a set of sensors that is afterwards converted to higher-level data. Through this technology, almost any surface could be converted into an intelligent surface that allows operations such as controlling a robot by using physical touch information as input. Tactile systems are of particular interest in the field of social robotics due to the fact that social robots are prone to experience expected or unexpected physical contact with humans. In this sense, touch has an affective component that helps convey high-level information that is relevant to human-robot interaction [25]. As such, for a social platform to effectively interact with humans, it must be able to properly interpret and understand these types of cues.

The structure of this chapter is organised around two main sections related to touch interaction. The first section analyses this interaction from a technical standpoint, exploring the various technologies implemented across multiple touch interfaces and helping to define the concept of STS. This section puts some emphasis on touch interfaces that use acoustic technology. Afterwards, it covers different kinds of touch interfaces applied in the robotics field and finishes by indicating the challenges involved in STS technology design. The second main section focuses on the social component of touch, defining this concept, discussing its relevance, and finally indicating how it is currently being implemented, emphasising robotics-related applications. Finally, a summary is provided that connects what has been discussed in the preceding sections to the thesis proposal.

2.1. Tactile Sensor Technology

Tactile sensing has been relatively neglected in the early years of robotics compared to other perception methods, such as vision and hearing [21]. Despite this, the research and the industry started directing their attention to this perception method in the 2000s. This shift materialised in a wide range of applications based on tactile sensing in fields such as biomedical engineering [26]. In this sense, primary efforts to develop this touch-sensing technology have been oriented towards developing high-performance tactile sensors using new materials [27].

This section will explore the main technologies applied to this field and their principles. Afterwards, we will emphasise several works introducing acoustic sensing technologies to these systems. Then, this section explores some applications in the robotics field. Finally, we will discuss the significant challenges this type of technology is currently facing and how they relate to the technical goals of the work.

2.1.1. Overview

As an essential element to enable the precise control of robots and also the safe interaction between humans and machines, tactile sensors have become the cornerstone of most intelligent systems [28]. Strain and pressure sensors are one of the main components of more complex tactile sensing systems. These can convert mechanical stimuli into a wide range of electrical and optical signals. Among the technical strategies behind a tactile sensing system, the more commonly used are capacitive, piezoresistive, piezoelectrical, and optical [21], described below:

- *Capacitive* sensors consist of a dielectric material between two electrodes that transmits the mechanical stimuli through a change in the capacitance. This kind of sensor shows an excellent frequency response, high spatial resolution and dynamic range. Despite their advantages, capacitive sensors have exhibited susceptibility to noise as their main weakness [26].

- *Piezoresistive* sensors convert mechanical stimuli into a change in the resistivity of the sensing structure [29]. They are easy to manufacture and integrate and less noise-resistant than capacitive technologies. However, they are affected by hysteresis, causing a lower frequency response than capacitive sensors [26].

- *Piezoelectric* sensors are based on the piezoelectric effect, where an electrical charge is generated by mechanically deforming the piezoelectric material [30]. This kind of sensor exhibits a very high-frequency response, making them a great choice for dynamic signal

Table 2.1: Summary of robot sensing technologies.

Sensor Technology	**Advantages**	**Disadvantages**
Capacitive	-High dynamic range. -Linear response. -Robust	-Susceptible to noise. -Some dielectrics are temperature sensitive. -Capacitance decreases with physical size, ultimately limiting spatial resolution.
Piezoresistive	-Wide dynamic range. -Durability. -Good overload tolerance.	-Hysteresis in some designs. -Elastomers need to be optimized for both mechanical and electrical properties. -Limited spatial resolution compared to optical sensors. -A large number of wires may have to be brought away from the sensor. -Monotonic response but often not linear.
Piezoelectric	-Wide dynamic range. -Durability. -Good mechanical properties of piezo materials. -Force sensing capability	-Inherently dynamic: output decays to zero for constant load. -Difficulty of scanning elements. -Good solutions are complex
Optical	-Very high resolution. -Compatible with vision sensing technology. -No electrical interference problems. -Processing electronics can be remote from sensor. -Low cabling requirements.	-Dependence on elastomer in some designs. -Some hysteresis.

sensing, for example, measuring vibrations. Despite this, their main weakness resides in the fact that they are unable to measure static deformations because of their large internal resistance [26].

- *Optical* tactile sensors are implemented as a result of coupling the phase, polarization, intensity, or wavelength of a light wave with a geometric change of electromagnetic waveguide. [31]. This sensor family tends to have a high dynamic response range and high spatial resolution [26]. Their main weaknesses are integration complexity and power consumption.

Table 2.1 expands the comparison between the sensing materials listed before. The properties of these materials allow them to effectively sense the geometry, presence, and point of contact of touched objects; they also gather additional information like flexion and torsion. However, in reality, humans use their sense of touch combined with other sensory modalities, such as hearing or vision [32]. Tactile sensing systems that rely only on a single sensor might have several limitations, such as data uncertainties or limited spatial coverage. For this reason, designing a tactile system should consider integrating multiple sensors and signal sensory modalities. In this way, the system can collect more information from the environment. Combining the requirements above is the first step towards achieving a *STS system*. A STS system combines signal transduction, signal conditioning, data transmission, signal processing, and a control system to emulate the human tactile sensing system.

In general, combining information from multiple sensors can yield better system performance. An effective option to achieve this is by using *sensor fusion* [21, 33]. Sensor fusion can be implemented at any stage of a STS system. Depending on the collected signal and the problem to be solved, a designer can adopt different approaches to combine the information from the sensors. For example, if the sensors measure the same physical phenomena, the signals could be directly merged. Otherwise, if the data is generated from different sources, it might be preferable to fuse it in other phases of the smart sensing system, like in the feature extraction or decision-making stage. For example, Jia et al. [34] fuse features from three modalities: electrode impedance, internal fluid pressure, and vibration. There are even more sophisticated examples, like in the proposal from Mittendorfer et al. [35, 36], that proposes an array of tactile modules. Each of these modules combines proximity, temperature and acceleration sensors. Our proposal strives to apply some of these ideas by **designing a STS system able to implement sensor fusion to improve human-robot social interaction**.

2.1.2. Tactile Perception based on Acoustic Sensing

Traditional touch-sensing technologies suffer from drawbacks such as hardware complexity, high manufacturing cost, and high power consumption. These drawbacks can also affect the platform where they are present. For example, they can introduce cross-talk with other electronics in the device or reduce optical performance and transparency on touch screens [37]. Therefore, we propose the implementation of acoustic devices as an attractive alternative for touch interaction in social robotics. Our approach includes piezoelectric devices—mentioned before as part of the 'traditional technologies'—, but instead of using them as force or strain sensors, we plan to implement them as passive acoustic sensing devices.

Despite not having examples in the social robotics field, acoustic sensing has been applied to touch detection, classification and localisation in multiple research fields, predominantly related to interactive displays. For example, in 2002, Paradiso and Checa [24] presented a system

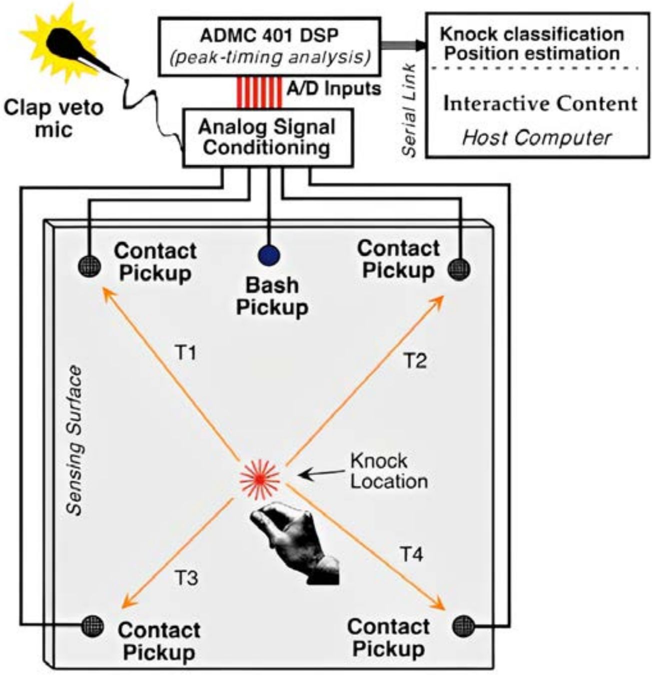


Figure 2.1: Layout for the acoustic tap tracker system by Paradiso and Checa [24].

to locate and classify touch interactions such as *taps* and *knocks* on an interactive square glass surface. They placed four contact microphones (also known as piezoelectric pickups) on the corners of the interactive screen and implemented touch localisation using Time Difference of Arrival (TDOA) measurements through cross-correlation of the sound signals. Authors reported an accuracy of 2 to $4cm$ on a surface of 2.24 square metres. The system was also able to classify touch gestures (i.e. *knock*, *tap* or *bang*) but without specifying the technique employed for this purpose. Figure 2.1 shows the complete setup of their system. Later, Lopes et al. [38] proposed a prototype 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 a glass surface of 1.12 square meters using different hand parts and expressing gestures such as finger *taps*, a *knock*, a *slap*, and a *punch*. Despite being one of the primary inspirations for our proposal, this work did not include details of the performances in contact source localisation or touch gesture classification.

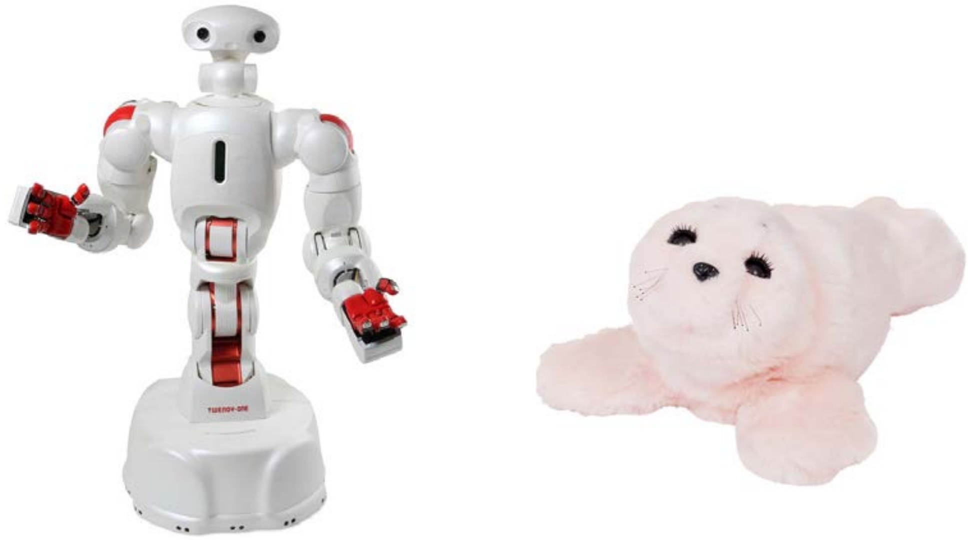
Two years after the proposal by Lopes et al., in 2013, Ono et al. [39] introduced 'Touch and Activate', an acoustic touch sensing technique that comprises an actuator (the speaker) and a sensor (the contact microphone) attached to the object's surface. It allowed recognising some touch interactions with the object, namely *support*, *hold*, or *grasp*. This method had some drawbacks, as it was tested with small objects and only with solid materials like wood, metal, plastic or ceramic. Nikolovski et al. [40], in the same year, proposed a similar approach based on a combination of actuators and sensors but centred only on contact localisation. They developed a $10mm$ thick and 1 square metre screen panel for locating low-energy fingernail taps. The system implemented a touch localisation method based on Lamb wave absorption [41]. Following the work of Nikolovski, Firouzi et al. [42] presented an ultrasonic touch screen system also based on the Lamb wave principle. Their proposal detected multiple touch contacts simultaneously and had high contact sensitivity. More specifically, it presented a resolution of 0.5 square centimetres. The main disadvantage these three works have in common is that they require active transducers to create a sound signal on the sensing surface, thus increasing power consumption and deployment complexity.

In 2014, Xiao et al. [43] presented 'Toffee', a sensing approach that extended touch interaction onto ad-hoc adjacent surfaces, mainly tabletops. They proposed a portable approach that required only a tabletop with piezo microphones located at the device's four corners. By placing the laptop on a surface, the system gave touch-sensing capabilities to that surface due to gravity. The localisation error of their proposal was $10.2cm$. More recently, in 2021, Jeong et al. [44] presented 'Knock&Tap', an audio-based approach capable of performing gesture classification and gesture localisation through deep transfer learning. The proposal comprises a single 4-microphone array to record the sound of the user's knocking and tapping gestures on a wood/glass panel. The system can differentiate between 7 touch gestures on both wood and glass panels. Knock&Tap classifies the gesture type and location with an accuracy of up to 97.24% and 92.05%, respectively. Since the system integrates air microphones as its main sensor, it is susceptible to ambient noises. In the same year, Seshan presented 'ALTo' [45], offering a low deployment complexity with just a set of four piezo microphones. They reported an error in the localisation of $1.45cm$ on the x-axis and $2.72cm$ on the y-axis. The last two systems [44, 45] presented the same drawback: they must preprocess the sound signals offline, meaning that they do not include a touch activity detection phase in their pipeline.

The systems mentioned in this subsection focus on providing the highest hit rate in touch gesture recognition and localization. Despite this, the main drawback they all have in common is that none of the proposals above finds a balance between portability, integration, and accuracy rate. In our proposal, we see this balance as critical to integrating this type of technology into a social robotic platform.

2.1.3. Smart Tactile Sensing Systems in Robotics

A robot's physical interaction with its environment is based on its perceptual and learning abilities, and tactile perception is an important element in this interaction process. Among its



(a) TWENDY-ONE, a hard-skinned robot [48]. (b) Paro, a soft-skinned robot [49].

Figure 2.2: Examples of hard and soft-skinned robots.

many applications, it primarily ensures the robot's stability, safety, and compliance [28]. By sensing the geometry, texture, presence, and position of touched objects, they enable the accurate recognition and safe interaction of humans and robots. Despite the possibilities offered by the sense of touch, the limitations of existing sensors, perception, and learning methods have caused robotic tactile research to lag far behind other sensing modalities, such as vision and hearing.

In robotics, several sensing technologies have traditionally been applied to touch detection and gesture recognition. Nicholls et al. [46] conducted one of the first studies to explore proposals based on endowing robots with skills related to touch recognition. These authors found that touch interaction in robotics implemented predominantly 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. Liu et al. [47] made a broader study, reviewing the theory and methods of robotic embodied tactile intelligence. This work also presented the challenges this field has to face. Their work concludes that the design of tactile perception and learning methods for embodiment intelligence should be based on developing new large-scale tactile array sensing devices.

Argall et al. [22] focused on how social robots integrate tactile technologies in their designs. This article classified robots by considering their shell's consistency, distinguishing between soft

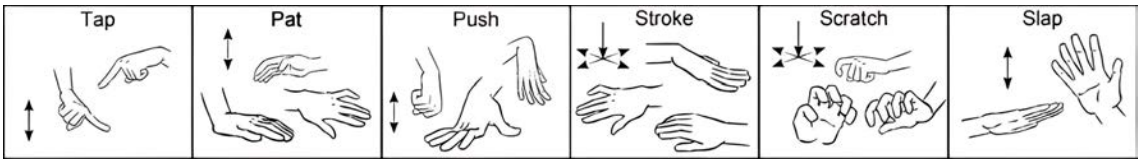
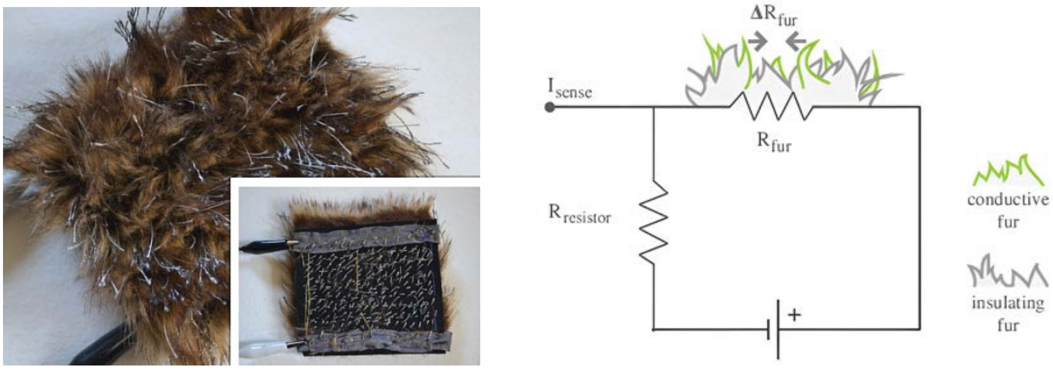


Figure 2.3: The six touch gestures recognised by Silvera et al. [53].

and hard-skinned robots. As they pointed out, robots such as WENDY [50], or its successor TWENDY-ONE [48] (shown in Figure 2.2a), belonged to the hard-skinned group. According to the survey, hard-skin robots usually integrate Force-sensitive Resistor (FSR), accelerometers, capacitive sensors, force/torque, and deformation sensors to enable touch detection. Even though these robots can detect physical contact, they cannot locate its source or differentiate the kind of contact performed. The other social robots studied in this survey are soft-skinned robots, such as Paro [49] (shown in Figure 2.2b) and CB2 [51]. Soft-skinned robots are equipped with piezoelectric, FSR, capacitive sensors, potentiometers that provide kinesthetic information, temperature sensors (thermistors), electric field sensors and photo reflectors to detect physical contact. PROBO, a soft-skinned robot shaped like a huggable animal-like creature [52], is equipped with 1000 force sensors. This proposal implemented the force sensors in a grid to detect the amount of pressure exerted. In addition to force sensors, PROBO also integrated around 400 temperature sensors and nine electric field sensors. This combination of sensors successfully detected contact and recognised some touch gestures.

There have been multiple works in the literature focused on touch gesture classification. In their research, Silvera et al. [53, 54] designed an advanced touch interaction system to identify gestures in a robotic arm. They carried out several experiments using an artificial arm covered with a skin layer based on the principle of Electrical Impedance Tomography (EIT) [55]. Their proposal successfully differentiated six different kinds of touch gestures (see Fig. 2.3). Their touch classification module was based on a LogitBoost algorithm, allowing the system to achieve an accuracy of 0.740 in cross-validation using a dataset composed of 1050 instances from 35 users. Following this concept of creating sensitive artificial skin, a group of scientists at Columbia University developed a new type of haptic sensor based on conductive fur in 2012 (shown in Fig. 2.4). This fur comprises a series of conductive wires that identify tactile gestures based on the electrical current that flows through them [56]. Three different gestures were trained using a machine learning classifier, yielding an overall accuracy of 0.820 in the experiment results. Albawi et al. [57] continued this line of work by designing an artificial arm covered with a sensible skin layer. Their proposal registered the pressure applied to the arm and processed it with a Convolutional Neural Network (CNN). This set-up achieved an accuracy of 0.637 using cross-validation. The authors proposed a complete set of 14 touch gestures in their approach.



(a) Artificial skin based on conductive fur.

(b) Electrical schematic.

Figure 2.4: Touch sensor based on conductive fur presented by Flagg et al. [56].

Muller et al. [58] proposed a combination of capacitive and pressure sensors mounted on an assistive robot. Their proposal achieved an accuracy of 0.740 when differentiating four possible gestures. With respect to their classification module, the authors integrated Gaussian Mixture Models and validated the system performance through cross-validation. Hughes et al. [59] also proposed using deep learning to deal with touch recognition on social robots. This system achieves an accuracy of 0.613 with a combination of CNN and Recurrent Neural Networks (RNN). Zhou et al. [60] proposed an evolution of these techniques using a 3D CNN that achieved an accuracy of 0.761 using the same database as Hughes to train its model.

Lastly, we must highlight the proposal by Cooney et al. [61]. Their work presented a survey focused on recognising 20 affective contacts on a humanoid robot (strokes on the cheek, kisses, handshakes, hugs, etc.). This approach combined artificial vision techniques with Kinotex tactile sensors. In this case, they implemented an Support Vector Machine (SVM) classifier as their machine learning classifier of choice. They showed an accuracy of 0.905 validating through cross-validation. Their dataset consisted of 340 instances gathered from 17 users. One of the main drawbacks of their proposal is the need for external cameras that their approach requires. Table 2.2 summarises the previous works centred on touch gesture classification. The table includes the platforms used, the technologies implemented, the number of gestures the system can distinguish, and the techniques' results using cross-validation and accuracy as a metric.

The majority of the works presented in this subsection emphasise that traditional touch sensors used in HRI have some flaws, such as short range, a proclivity for false positives, susceptibility to noise, inability to recognise touch gestures, poor scalability, and, in some cases, high complexity.

Table 2.2: Comparison of gesture recognition using several different techniques. The works are ordered according to their accuracy.

Study	**Platform**	**Technologies**	**Num. of gestures**	**Accuracy**
Hughes et al. [59]	Human-animal affective robot	Pressure-sensitive robotic skins	4	0.613
Albawi et al. [57]	Artificial robotic arm	Pressure-sensitive robotic skins	14	0.637
Silvera et al. [54]	Artificial robot arm	Pressure-sensitive robotic skins	6	0.740
Muller et al. [58]	Socially Assistive Robot	Capacitive and pressure-array touch sensors	5	0.740
Zhou et al. [60]	Human-animal affective robot	Pressure-sensitive robotic skins	5	0.761
Flagg et al. [56]	Human-animal affective robot	Conductive fur	3	0.820
Cooney et al. [61]	Humanoid robot mock-up (foam-covered mannequin)	External cameras, built-into optical sensors	20	0.905

2.1.4. Challenges and Design Considerations

This section covers the main challenges in the design of STS systems. Although the development of these technologies has drawn increasing research attention since the turn of the century, making it an active area, the application of smart tactile systems in the robotics industry is still in its infancy. As Zou et al. pointed out [21] and as some of the limitations of techniques covered in the literature revealed, this research field has to face some challenges. These difficulties significantly impacted the requirements that our approach must meet. Some of the issues that were listed by Zou et al. are as follows:

1. *Cost*: Since most of the existing tactile systems reported in the literature are still in the experimental stage, one of the difficulties the researchers face is determining how to reduce the cost of tactile sensor systems.

2. *Hardware*: The challenges in this aspect are related to improving tactile systems' performance concerning physical aspects (e.g., conformability, spatial resolution), tactile sensor arrangement, sensor performance (e.g., ability to measure various parameters, sensitiv-

ity), wireless communication, and crosstalk. In this sense, the industry is already exploring nanotechnology and microfabrication as suitable solutions for integrating signal processing units and multiple sensing modalities and providing a high-density array of sensors.

3. *Software*: Even though numerous tactile sensors with interesting properties, such as mimicking the human sense of touch, have already been developed, tactile sensors are rarely used in real-world applications. Practical tactile sensing systems require appropriate hardware and powerful software, especially for systems operating in unconstructed environments. Tactile sensing development necessitates better sensors and efficient and effective data processing techniques.

4. *Modularity and portability*: Another issue that should be addressed is the ease of assembly and disassembly. Hardware and software for the tactile sensing system are typically developed using task-specific criteria. Modularised designs that make switching between different robotic platforms easier are highly desired from a design standpoint.

Our proposal faces the challenge related to cost by integrating piezoelectric contact microphones for touch recognition. These sensors are cheap, and thanks to how sound propagates, especially on solid surfaces, the whole exterior of a robotic platform could be covered without needing many receivers. Regarding the cost of the software, our objective is to implement a system depending only on free and open-source software. The second challenge of a STS system designer is related to the hardware. As Section 3.2.3 will describe in more detail, our proposal improves the classic microphone-soundcard combo by integrating both elements into the same device. This aspect impacts not only the cost but also the modularity and portability of the design.

The next challenge our proposal has to face involves software development. Our system tries to balance accuracy, overall performance, and computational cost. This balance is relevant when a designer must consider that a robotic system integrates many other processes with its own computational cost that must run simultaneously. Concerning robotics, specifically social robotics, the design also needs to consider non-functional requirements such as response times. This requirement is essential when striving to achieve a natural human-robot interaction. Lastly, modularity and portability greatly conditioned some of the choices made during the design of the system. Some approaches in the literature overlook this aspect in exchange for better results. But, if we aspire to integrate our proposal into a real-world and real-time platform, this needs to be one of the primary objectives. One of the prominent examples resides in the choice of sensing devices. As mentioned above, a setup based on piezoelectric microphones would not require excessive receivers. In the software aspect, our system is encapsulated through software packaging so that it can be easily installed and configured on different platforms.

2.2. Touch Interaction

Touch has been described as the most direct and fundamental form of communication with the outside world; it is crucial to human development, social relationships, and emotional communication. Despite this, in the field of robotics, the sense of touch has not received the attention it deserves [17, 21]. This situation has been attributed to the difficulty of studying it, both technically and socially, as well as the emphasis on verbal, visual, or both combined [62]. In 1957, Frank [2] first acknowledged this situation, focusing on the psychophysics of touch and the varying cultural patterns linked to this sense. Geldard [63] raised in 1960 a similar concern. However, his research was mainly focused on increasing the understanding of the low-level mechanics of touch communication. Later in 2006, Hertenstein et al. [64] restated this concern by documenting three times the number of audition-centric publications and 13 times the number of vision-related published works. This work also indicates that methodological and philosophical influences could be the reason for the diminished research interest in the study of touch.

Most research on the sense of touch has focused on addressing its discriminative aspects [14]. These are related to the use of the touch as an exteroceptive organ in charge of detecting, discriminating, and identifying stimuli that happen outside the body to conduct the behaviour [25]. In other words, from this discriminative point of view, the sense of touch is designed to obtain information about the external world. In addition to this function, the sense of touch also plays an important affective and interoceptive role [25]. When considering unpleasant or painful sensations and pleasant sensations or stimulation of erogenous zones, the hedonic aspects of the human sense of touch are easily understood [5]. Despite this, the neurophysiology of this role, where affective touch is the basis, is still not mature [65, 66].

Through this Section, we explore this second facet of the sense of touch, defining the concept of social touch. We will also cover how social touch affects humans positively and negatively. Afterwards, this Section entangles all these concepts with technology, emphasising its applications to robotics and, more specifically, to social robotics. And finally, we discuss the challenges that the study of social touch currently has to face.

2.2.1. What is Social Touch?

Social or interpersonal touch is often defined as touch occurring between two or more individuals in co-located space [8]. The use of social touch is diverse, ranging from its use during greetings to showing affection and support [9]. Although it is a less frequent social signal than facial expressions, for example, touch can deeply influence social interactions [5]. As an example, touch can serve a persuasive function [67]; it can also lead to more favourable evaluations of the *toucher*[1] [68], and it also helps regulate emotional and physical well-being [69]. Touch is the first sense to develop in the womb [1, 3]; it is necessary to bond successfully between a child and its mother [70]. And in addition to this, the sense of touch has a very relevant role in later social life [5]. From a purely biological point of view, Olausson et al. [65], looking at the role of nerve fibres in the affective properties of touch, concluded that the essential role of C-tactile (CT) afferent fibres is to provide or support emotional, hormonal, and behavioural responses to skin-to-skin contact. These findings have led researchers to propose the *social touch hypothesis* [66]. According to this hypothesis, CT afferents may act as a filter in conjunction with other mechanoreceptors (e.g., those for discriminative touch) to help determine whether a certain touch has social relevance. The following idea supports this hypothesis: caressing touches, to which CT afferents are sensitive, are significant in human affiliative interactions [10, 11].

One of the first findings on social touch came from Harlow and Zimmermann in 1958 [71, 72]. The prevailing view at the time was that the primary role of the caregiver was to satisfy the infant's direct drives, e.g. hunger, thirst, and pain. Contrary to this belief, they proved the contact comfort theory, which postulates that to increase affective bonds, a primary role of nursing is maintaining direct, physical contact between the infant and the mother. Both researchers conducted a series of studies with infant monkeys. These experiments consisted of separating them at birth from their mothers. The monkeys would then be raised by two inanimate surrogates, one providing a greater degree of tactile comfort concerning the other. In one study, all monkeys had access to both surrogates, but a soft cloth mother fed one group, and a rigid wire mother fed another. Their experiment demonstrated that monkeys sought the cloth mother much more frequently in the presence of a fear stimulus. When this stimulus was not present, they would spend much more time in physical contact with this surrogate.

In 1983, Heslin and Alper developed one of the first efforts to achieve a 'taxonomy of touching' [73]. Their taxonomy distinguished five 'situations/relations' of social touch: functional/-professional, social/polite, friendship/warmth, love/intimacy, and sexual/arousal. The authors considered harmful touch types rare occurrences, so they were not included in their taxonomy.

[1] In the manuscript we use the term *toucher* to describe the person or the agent who is actively touching, as opposed to the one who is touched.

Furthermore, this arrangement implied a progression of increasing levels of intimacy. A few years later, Jones and Yarbrough carried out one of the more extensive social touch studies by observing participants' daily touches at a university over an extended period and recording them [9]. The information gathered included elements like the location and person who initiated the touch, the social setting, the presence of others, and the reason and kind of touch. In addition, they made notes about the other person's gender, familiarity, age, and social position. From these data, Jones and Yarbrough differentiated seven main touch groups from their findings: positive affect, playful, control, ritualistic, hybrid, task-related, and accidental. Using these works as a foundation, Yohanan et al. [17] advanced a step forward by presenting in 2012 a complete dictionary of touch gestures composed of 30 items adapted from the literature on human-animal interaction and social psychology. In addition, they also reported patterns of gesture use for emotional expression, physical properties of the likely gestures, and analysis of the human higher intent in communication.

Touch and many other forms of non-verbal communication have a cultural and social dimension, and there are studies on differences in how people interact through touch across societies and geographic regions [74]. Researchers McDaniel and Andersen conducted a study in 1998 that analysed tactile data from 154 people from 26 different nations [75]. The study concluded that touch varies between people of different nationalities, showing how the average number of body areas touched changed in different societies. The influence of interpersonal relationships on tactile interaction was also observed, with more body areas touching the closer the relationship between subjects. More recently, a study by Sorokowska et al. in 2021 analysed different cultural and individual variables that could influence affective touch [76]. These factors were regional environmental temperature, degree of ideological conservatism, religion, gender and age. To gather the necessary data, they surveyed 14487 participants from 45 countries. The questionnaire presented four forms of affective touch (*embrace*, *hug*, *kiss*, and *stroke*) and asked whether they had expressed these forms of affection with friends, partners, family members, or children in the week before the study. The results indicated that affectionate touch was more diverse in warmer, less religious and less conservative countries among young and liberal people. It was also more prevalent in couples and parent-child relationships.

2.2.2. Effects of Social Touch

To better understand how robots could be designed for social touch, it is necessary first to evaluate the effects of human-human social touch. In the following paragraphs, we discuss how social touch affects various important areas studied in the literature, such as attitude and behaviour change, attachment and bonding, communication of affect, and physical and emotional well-being [14]. We will also discuss some works focusing on the effects when social touch is deprived. Finally, we will review some studies centred on social touch's impact on the toucher.

The **attitude and behaviour** of the person who another person is touching may be affected by social touch. A light touch to the hand, arm or shoulder can positively affect how the recipient feels about the peer [68, 77–79], how they are feeling emotionally [78], and how they feel about the environment in which the touch is occurring [78, 79]. In this aspect, Fischer conducted one of the earliest studies in 1976, which focused on the consequences of interpersonal touch in professional and functional situations [78]. The experiment consisted of the following; The library clerks alternated between returning library cards to library students by briefly touching or not touching their hands. After the interaction, these students (101 in total) were asked to assess the library staff and answer a series of questions. The results showed that women responded more positively to the questions on the questionnaire and felt more affectively positive in touch than in non-touch conditions. At the same time, for males, the responses were more varied.

In addition to altering the recipient's attitude toward the touch, social touch can also affect the his/her behaviour. The *Midas touch effect* [67], which describes the benefits of social touch on pro-social behaviour, is a commonly studied phenomenon in social touch research. With a wide variety of observed pro-social behaviours, this effect has been proven in numerous ecologically sound settings. Increases in willingness to return lost money [80], influence over purchase decisions [79], and in restaurants, increases in tipping [67, 79, 81] or improved compliance with menu item suggestions in a restaurant after a touch by a waitperson [82] are all examples of the Midas touch effect.

The next area affected by social touch is **attachment and bonding**. According to *attachment theory*, a baby will seek out its caregiver (usually the mother) when it's upset [83–86]. A crucial signal for safety and security is physical contact [85]. The infant's persistent attempts to make physical contact with the mother and the mother's reactions to those attempts form the attachment relationship [83, 86]. For low birthweight infants to form secure attachment relationships, nurturing touch, rather than touch frequency, has been found to be crucial [85]. However, children who receive less physical contact from their parents report higher levels of current depression and exhibit less secure attachment patterns as they age [87]. Other ideas about interpersonal relationships, such as love and intimacy, are closely related to attachment [88]. Adult romantic attachment can be explained using attachment theory, which has long been acknowledged [89]. Non-sexual physical affection positively correlates with relationship and partner satisfaction and can help romantic couples resolve conflicts [90]. Furthermore, the release of the oxytocin hormone that occurs when someone is touched plays a vital role in the dynamics of the couple since it may mediate the bonding effects of social touch [91–93].

Social touch has also shown its impact on **human well-being**. For example, infants are thought to need social touch to develop appropriately, as lacking physical contact in the first few months of life can later have detrimental effects on their well-being [94]. The research focused on orphans deprived of social and sensory stimulation demonstrated these children lacked cognitive, social and emotional development [95–97]. In later life, particularly for those in romantic relationships, the beneficial effects of social touch on physical and emotional well-being are also important. For example, holding a partner's hand lowers pain ratings when receiving a painful stimulus than holding an object or a stranger's hand [98]. Before a stressful task, partner contact in the form of hand holding, hugs, or massage reduces stress responses, as indicated by cortisol levels, blood pressure, and heart rate [99, 100]. These effects of touch on stress responses are more potent when touched by a spouse than when touched by a stranger. Nevertheless, these studies also indicated that the quality of the marriage might influence stress responses [101]. Despite the differences between a touch from an acquaintance versus a stranger, the research found that a stranger's social touch can also lower a person's heart rate [102], and such effects can be beneficial for stress reduction in healthcare settings. For example, a nurse's touch before surgery has been shown to positively impact a patient's affective state and stress level [103]. However, it is important to note that these effects were discovered only in contact between women and women, while touch had the opposite effect in male patients.

Regarding affect communication by touch, to study the relationship between touch and emotions, or so-called **affective touch**, Hertenstein presented a paper in 2001 in which he tested how the way a baby is touched by its mother can influence the baby's emotions and behaviour [104]. In this work, multiple objects were presented to babies. It was observed that those babies touched by their mothers by tightening their fingers around the abdomen interacted less with these objects and displayed more negative emotions than babies touched with a more relaxed grip. Further studies demonstrated that touch communicates not only positive or negative affective states but the nature of the touch itself can be used to communicate discrete emotions. These works, conducted by Hertenstein between 2006 and 2009, found that it is possible to convey emotions through touch, focusing on eight emotions (anger, fear, happiness, sadness, disgust, love, gratitude and sympathy). Hertenstein also demonstrated that gestures are commonly associated with specific emotional states [64, 105]. The results of these studies showed that the prediction ratios for emotions conveyed by touch ranged between $50 - 70\%$, values similar to those obtained for visual and hearing emotion communication [106].

More recently, Fotopoulou et al. [107] proposed a novel approach focusing on social touch and its function as an **affective regulation** mechanism, including its embodied, cognitive, and metacognitive processes. They found that social touch appears to aid affective regulation in three distinct but related ways. It first controls affects by confirming embodied predictions about social proximity and attachment. Secondly, as *caregiving touch*, it controls affect by socially enacting homeostatic control and co-regulating physiological states. Finally, *affective touch*, such as gentle stroking or tickling, controls affect through allostatic regulation of the salience and epistemic gain of distinct experiences in specific contexts. In their work, they emphasised that social touch contributes to affective regulation through various functions ranging from direct, physiological co-regulation to the development of allostatic, cognitive, and metacognitive models of regulation and social cognition.

All the works presented in these paragraphs covered social touch's effects on humans. However, there seems to be a lack of studies that cover the impact that causes its deprivation. This fact is mainly related to its almost ubiquitous presence in human life. The COVID-19 pandemic and its restrictions, such as social distancing, allowed von Mohr et al. [108] to study the relationship between social distancing, tactile experiences and **mental health**. In their study, 1746 participants conducted an online survey that inquired about professional, friendly and intimate touch experiences during COVID-19-related restrictions, the extent to which touch deprivation results in craving touch and the overall impact on mental health. They found that, despite intimate contact being the most experienced during the pandemic, its deprivation during the pandemic restrictions was associated with greater anxiety and greater loneliness. Another important finding was that craving touch during COVID-19 depended on individual differences in attitudes, attachment style and experiences towards the touch. Their work emphasised the role of interpersonal, specifically, intimate touch in times of uncertainty and distress.

Lastly, we close this subsection by addressing our concerns regarding **active touch** since it is an essential element of the proposal this thesis covers. Active, interpersonal touch has the unique quality of being reciprocal; touching someone without being touched in return is impossible. The affective experience of caressing another person and its psychological implications for the active individual are still unknown, even though many studies have looked into such dual properties concerning non-affective, discriminatory touch [109]. In that sense, as the reader might have deduced, the literature focused on the effects of human-human active social touch —that is, from the toucher's perspective— from an affective, interoceptive standpoint is still lacking. Moreover, little is understood about what drives and upholds the pro-social human tendency to interact with others. Gentsch et al. [110] tried to shed some light on this question by conducting a series of six experiments to test the hypothesis that active stroking produces greater sensory pleasure on others' skin than on one's own. They called this phenomenon the *social softness illusion*. They also discovered that the receiver's neurophysiological system for affective touch was only selectively activated when the touch occurred, producing this softness illusion. More importantly, they confirmed that the expectation of inducing a positive bodily state in someone else seems to influence the perception of active touch-giving. From all these findings, they concluded that this sensory deception supports a brand-new bodily mechanism of socio-affective bonding and heightens our desire to touch others.

Our research seeks to build on the work of Gentsch et al. [110] by focusing on the role of active touch in tactile interaction and applying it to social robotics. As it will be discussed in Chapter 6, one of our goals is to observe the impact that active contact with a social robot that can perceive, distinguish, and react to touch has in the user's experience during human-robot interaction.

2.2.3. Social Touch Technologies

Huisman [14] defines the use of touch-sensing technology for social touch interactions as Social Touch Technology (STT). This concept includes interactions with artificial social agents that are capable of responding to and applying social touches [111, 112], as well as situations where human communication partners engage in social touch mediated by technology [113]. According to the *Media Equation* theory, when people interact with an intelligent system, they do so as if the system were a social actor [15]. This concept is extended by research on embodied conversational agents and social robots, which give artificial social agents a virtual or physical embodiment that can be used, for instance, to express emotions during interaction with a user [114, 115]. By applying this logic, touches made by a virtually or physically embodied artificial agent may be interpreted as social touches by the user. Huisman also pointed out that genuine human-to-human social contact may also manifest in situations where artificial social contact is produced. In this sense, social robots constitute a very appropriate agent during tactile interactions, since there is a direct relationship between social touches and a social robot's embodiment. This is due to the fact that a social robot can give and receive physical contact using its physical body.

Tactile detection and HRI are the two main topics in tactile human-robot interaction [22]. We covered the former in Section 2.1.3, and in this section, we will be focusing on the latter, more specifically on the effects related to social touch —mentioned in the previous subsection— applied to HRI. In tactile HRI, physical interactions between the robot and a human can be studied from two perspectives: the perspective of behaviour development and execution by the robot [22] or the perspective of safety. In this section, we are interested in the former. Therefore, we will concentrate on the effects that touching or being touched by a robot might have on the other peer, the user involved in the interaction.

Social touch has already been applied to healthcare settings to improve people's well-being. Robins et al. [116] proposed the application of tactile interaction with social robots to help children with Autism Spectrum Disorder (ASD) that often suffer from hypertactility. The objective was for them to get accustomed to social touch, potentially making them more comfortable being touched by another person. The experiment consisted involved children with ASD physically interacting with a social robot in a free play session. One of the primary findings was that the children used different touch behaviours to engage with the robot [116, 117]. The children eventually developed a more natural touch interaction style with the social robot through these experiments. In 2015, in one of the few experiments centred on the effects active touching over a robot has on the user, Costa et al. presented an approach where the robot gave appropriate multimodal feedback to being touched. Their experiment demonstrated its positive effects on the body awareness of children with ASD [118].

A robot's simulation of social touch might be utilised to positively influence the user's attitude toward the robot, potentially leading to the user behaving more favourably towards the platform. As an example, Hieida et al. [119] studied how physical contact in the form of hand-holding in the early phases of child-robot interaction might positively influence a child's attitude toward the robot. In another study, Fukuda et al. used electroencephalogram (EEG) to measure an index called the median frontal negativity when a robot stroked a human hand. This experiment allowed them to verify the type of human physiological response caused by this [120]. Haans et al. employed a vibrating device to test how touch interaction influences humans' helping behaviour [121]. Another report by Bevan et al. connected physical contact with improved prosocial behaviour. For this purpose, they designed a negotiation task using a telepresence robot [122]. Lastly, Nakanishi et al. demonstrated by conducting multiple studies that conversations through a physically huggable humanoid cushion device, called Hugvie [123], enhanced interest and maintained trust in the partner [124, 125], reduced negative imagination about a topic [126], and improved attention and memory retention [127] when reading to children.

Touch interaction between those close to each other is also known to cause physiological changes, including hormonal changes [128]. Similarly, some studies have tried to prove that touch with robots also causes some of these physiological changes. For example, by verifying the urinary hormone balance, Wada and Shibata reported reduced stress in older adults that regularly touched Paro for two weeks [129]. In addition, conversations with others through Hugvie change the hormone levels in blood and saliva, related to the stress value, compared to observations when the subjects conversed with others using a simple mobile phone [130].

As we have discussed up to this point, a social robot must be able to perceive and recognise different tactile gestures to achieve a much deeper level of communication within robotics beyond visual and auditory interaction. According to Jung et al. [18], it also must be able to interpret these gestures and respond accordingly (see Figure 1.1). In pursuit of granting a robot the ability to interpret touch gestures, some studies have explored the communication

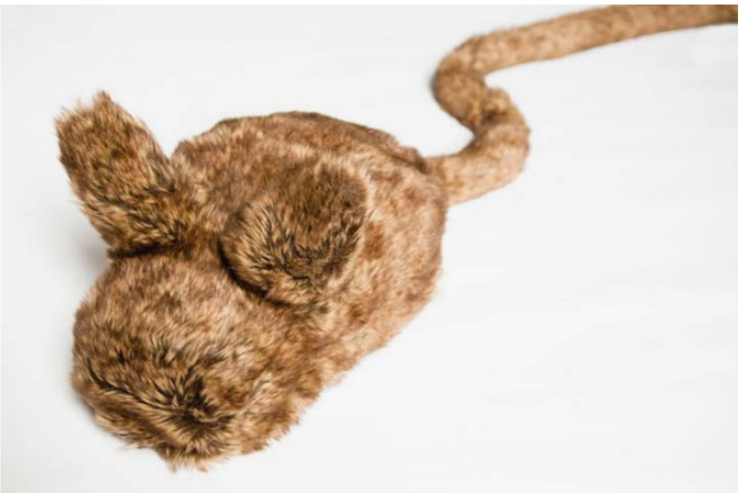


Figure 2.5: The Haptic Creature designed by Yohanan [17].

of affect through touch. Among these works, we can highlight the research Yohanan et al. [17] carried out in 2012. This research examined first how humans communicate emotional states through touch, as well as the prediction of the emotional state that the robot would have as a consequence of human interaction. The approach adopted in this study was inspired by human-animal interaction. To this end, the researchers designed a zoomorphic robot known as the *Haptic Creature*. To collect information about the locations of the touch contacts, they covered the inner surface of the robot with haptic sensors and accelerometers. Then, to classify the subject's emotional states and predict those of the creature, the researchers based their work on J. A. Russell's theory of emotions [131, 132]. This theory decomposes emotions in a multidimensional model, more specifically, in two dimensions: *valence* (level of pleasantness/unpleasantness) and *arousal* (level of arousal). This work also served as the basis for creating a gesture dictionary for interacting with the robot, mentioned in Section 2.2.2, consisting of 30 different gestures with their respective definitions. In a subsequent publication in 2015, Altun et al. used the signals collected by the force sensors to implement machine learning algorithms to classify the emotional state conveyed by the gesture [7]. They implemented a touch gesture classifier to study the correlation between touch gestures and emotional states, obtaining better results in classifying emotional states when the touch gestures have already been classified. They concluded by recommending a multimodal approach to improve the results in terms of emotion recognition.

Jung et al. also proposed using a zoomorphic robot to investigate human-robot interaction [18], structuring the recognition and interpretation of tactile gestures in three main points. First is extracting low-level parameters such as gesture contact area, duration and intensity. Then, they described and segmented the tactile gestures performed on the robot (they employed five different touch gestures in their study). Finally, they defined high-level social messages to be transmitted through these gestures. The results of the experiment showed that the social messages communicated by the participants through touch varied according to the emotional state they were in and also according to the social role of the robot perceived by the subject (emotional support, pet companion, etc.). Occasionally, participants were observed interacting with the robot in ways other than touches, such as through speech or eye contact, indicating the importance of multimodal human-robot interaction.

2.2.4. Challenges and Factors in the Study of Social Touch

In 2021, Saarinen et al. [133] designed a study to identify various psychosocial and situational factors and toucher characteristics that modulate the immediate experiences and responses to social touch. They concluded that depending on an array of contextual factors, the same touch gesture might not be experienced as pleasant and may not have potential long-term positive effects of touch [134, 135]. To produce pleasant touch experiences, it would be necessary to adjust psychosocial situational factors carefully so that, as likely as possible, they help perceive the contact as secure, appropriate, and pleasant. The factors mentioned in this work are as follows:

1. *Level of acquaintanceship*: In order to produce pleasant touch experiences, social touch could be utilized at a certain level of acquaintanceship (not in the first meeting but later meetings).

2. *Body part touched during the interaction*: Suvilehto et al. [136, 137] reported that members outside the family circle are allowed to touch only 20% of the body (primarily hands). For that, touch could be directed to a restricted body region such as the hands.

3. *Context*: Some evidence tentatively suggested that touch may more likely be experienced as pleasant if it occurs in a situationally appropriate way in a natural context, whereas repetitive touches in environments such as a laboratory may not necessarily be experienced in a positive way [133].

4. *Environmental factors*: Many touch-related experiments have been conducted in formal (laboratory) settings where there may be, for example, some unexpected unpleasant odours that might affect the touch experience negatively.

Overall, adjusting these situational factors during touch exposure could increase the likelihood that social touch could produce positive responses in the target person. We concur with the authors that at least some of these factors should be considered when preparing the experiments focused on studying either social touch in human-human interactions or those oriented towards human-robot interaction. Furthermore, the experimental conditions should be reported, and the results of experiments should be discussed related to these factors. In this work, we did not consider the acquaintanceship factor. Still, we addressed the rest by avoiding forcing the participant to touch the robot in zones that might be considered inappropriate by the volunteer and moving the robot to a more suitable environment of an office.

Most of the research Saarinen et al. conducted focused on human-human touch interaction settings. However, surveys like the one presented by Shiomi et al. [19] focused on research on social touch interaction in robotics. In their work, they devise the various challenges that social touch research may face in the future, at that time, in 2020. Some of the issues they raised helped to guide our design decisions. The following are the challenges they propose:

1. Clarify the difference in the effect of social touch between a human and a robot and between humans. To date, no attempt has been made to compare these effects directly. For this, the authors propose that developing a robot that can handle different kinds of social touch can help to investigate further in this direction.

2. Additionally, by using a robot, we can more easily examine the effects that variables such as age, gender, and appearance might have in touch interaction. Experiments that alter some of these variables could provide new information on the mechanism underlying the effects of social touch from a cognitive science and neuroscience perspective.

3. Another challenge is finding a solution to using a robot that can handle social touch in a real environment. The authors propose a lengthy experiment in a natural setting, moving the environment where social touch research occurs from the laboratories. First, the long-term experiment itself is complex, and issues with the robot's operability, like durability, must be considered. Investigations into the effects of social touch must also consider interpersonal relationships and hygiene issues. Issues that Saarinen et al. already discussed, as mentioned before.

We have designed our proposal to address the first challenge directly by designing a system that can endow a robot with the ability to handle different kinds of social touch. The second challenge is not covered in this work, but the data gathered for the experiments in Chapter 6, as it will be seen, could allow such studies in the future. Regarding the third, even though the experiments do not involve a long-term interaction, in the experiment from Section 6.2, we try to set up a real-world scenario where the user interacts by playing with a robot. In summary, the objective is to provide a touch-sensing system to allow further investigation that can answer some of the challenges Shiomi et al. propose in their study. As they pointed out, very few robots implement social touch as an interaction tool, opting instead for conversation as their primary means.

2.3. Summary

In this chapter, we have laid the foundations for the work presented from two perspectives: a technical perspective, focusing on the concept of acoustic sensing and another dealing with the concept of tactile interaction and, in particular, the idea of social touch.

The first section analysed this interaction from a technical standpoint, first defining why touch technology is essential today, exploring the various technologies implemented across multiple touch interfaces and defining the concept of Smart Tactile Sensing (STS). After this, we discussed in detail different STS systems based on acoustic technology, one of the main contributions this work intends to make. From here, this section contained examples of Smart Tactile Acoustic Sensing systems applied to the field of robotics. Finally, we ended this section by discussing the challenges that STS systems face and how they conditioned our work.

In the second section, we focused on tactile interaction, defining what it is and how it relates to social touch. This concept is particularly relevant because of the field in which the proposed system is introduced: social robotics. From here, we described the effects of social touch on human-human interaction, and afterwards, we connected these effects to the field of HRI. Finally, as in the previous section, we discussed the various challenges facing the study of social touch, both generically and specifically as applied to social robotics.

Chapter 3

Acoustic Touch Recognition System Setup

In Chapter 2, we described the related works that provided the basis for this work. The current one will contain different elements that were part of the Acoustic Touch Recognition System setup during its development. As a result, this chapter was designed to serve two purposes. On the one hand, it has an introductory function, listing and describing these elements that have been a part of the system or were involved during its evaluation. On the other hand, this chapter serves as a reference to which the reader can return when he or she needs more information about these elements when they appear —more succinctly explained— in the next chapters, as this chapter is linked to the next ones and vice versa.

In terms of chapter structure, and keeping in mind that this work is oriented toward social robotics, we first present the various robotic platforms in which the system has been tested, as well as their connection to the sections in which the robotic platform is particularly relevant. These agents and their primary functionalities will be described in the first section of the chapter. Following a description of the robotic platforms involved in the system's integration, we will discuss how the STS system has been integrated into each platform. This allows us to create a timeline that shows how the system's hardware has evolved from its conception to its current state. It should be noted that the details of the system's software design will not be covered in this chapter. The final section of this chapter lists and describes the software architecture in which our system must be integrated. This way, we establish a link between this chapter and the next, which discusses system software design and integration.

3.1. Robotic Platforms

The work presented in this manuscript has been carried out mostly in the Robotics Lab of the Carlos III University of Madrid, in the Social Robotics Group, and partly in the Instituto Superior Técnico, in Lisbon, during a stay. The Social Robotics Group's main research focuses on developing social robots and multiple applications centred on human-robot interaction. Besides human-robot interaction, this group's other research lines include cognitive stimulation, decision-making systems, dialogue management, expressiveness management and robotic perception. The Social Robotics Group has participated in research projects with Spanish and European companies and institutions. Among the different projects, we highlight the following, from the most recent to the oldest:

- **Social robots to mitigate loneliness and isolation in the elderly (SOROLI):** This project, which started in 2022 and will finish in 2024, is aimed at designing and implementing social robots in environments where they could help to mitigate loneliness of older adults.

- **Design of a social robot to help the elderly:** Starting in 2019 and finishing in 2021, this project aimed to design a low-cost social robot for assisting older adults that suffer from mild cognitive impairment. The robot was called Gero and was designed in collaboration with Arquimea[2], a company that develops healthcare technology-based applications.

- **Social robots for physical, cognitive, and affective stimulation for older adults (ROSES):** This project focuses on using robotic platforms to perform cognitive, physical, and affective stimulation therapies with older adults. This project was developed from 2019 to 2021.

- **Development of social robots for assisting older adults with cognitive impairment (ROBSEN):** This project, developed between 2015 and 2018, aimed to develop a social robot to assist older adults suffering from mild cases of cognitive impairment. In this case, the robot helps the caregiver (without replacing the person) and assists the patient in four scenarios: personal assistance, entertainment, stimulation, and security and safety.

- **Multi-Robot Cognitive Systems Operating in Hospitals (MOnarCH [138]):** This research project was developed from 2013 to 2016, and it was focused on using social robots to interact with children, staff and visitors in the pediatric unit at the Portuguese Oncology Institute of Lisbon. A new social robot, covered later, was specifically designed

[2] Arquimea webpage: https://www.arquimea.com/es/

> and built for this project. The work developed in the Social Robotics Lab focused on the robot's HRI capabilities. Although this project did not intersect with the lifespan of this work. The platform developed here was used during the research stay.

Since its foundation, the group has developed multiple robotic platforms oriented towards research purposes. The group's first platform completely designed and built was the robot Maggie [139], a personal social robot designed as a research platform for studying HRI, robot cognition, and robot autonomy. The next platform developed, in this case, in collaboration with other research groups as part of the MOnarCH project, was Mbot [140], a child-sized mobile robot designed to interact with paediatric patients in an oncological hospital. Both robots are shown in Figure 2.1 In the frame of the more recent research projects, a new platform designed to assist older adults that suffer from mild cognitive impairment was developed: Mini. We have employed these three platforms to evaluate our system, and in Mini, we propose a full software integration (described in Section 4.3). This is motivated by the fact that Mini is currently the only active platform in our laboratory. From a software point of view, it contains an updated version of its software architecture. All the platforms will be presented in the following subsections, from the oldest robot to the newest.

3.1.1. Maggie

The main concept behind the development of Maggie, shown in Figure 3.1, was to create a human-friendly robotic platform for social interaction research [139]. The robot needed to be an element a human could always enjoy interacting with, and, at the same time, the platform should allow being improved with new features and functions. Some of the design concepts taken into account when designing Maggie were its attractiveness to appeal to humans, therefore encouraging them to interact with the robot easily. Another concept was the robot's expressiveness. For this reason, Maggie uses body/arm/eyelids movements to interact with the user. The following design requirement was the robot's multimodality. Because a social robot needs to naturally interact with humans using multimodal interfaces, one design consideration was integrating multimodal interaction using tactile, facial/body expressions and verbal communication. And finally, the last design concept was Maggie's mobility to support applications like handling various house duties, assisting elderly and disabled people or tour guiding.

To meet these design concepts, Maggie has an artistic design of a 135cm high girl-like doll. Mobility is provided through a mobile base equipped with 12 bumpers, 12 infrared optical and 12 ultrasound sensors and a laser range finder. The upper part of the robot incorporates the interaction elements. The robot presents an anthropomorphic head with two degrees of freedom,

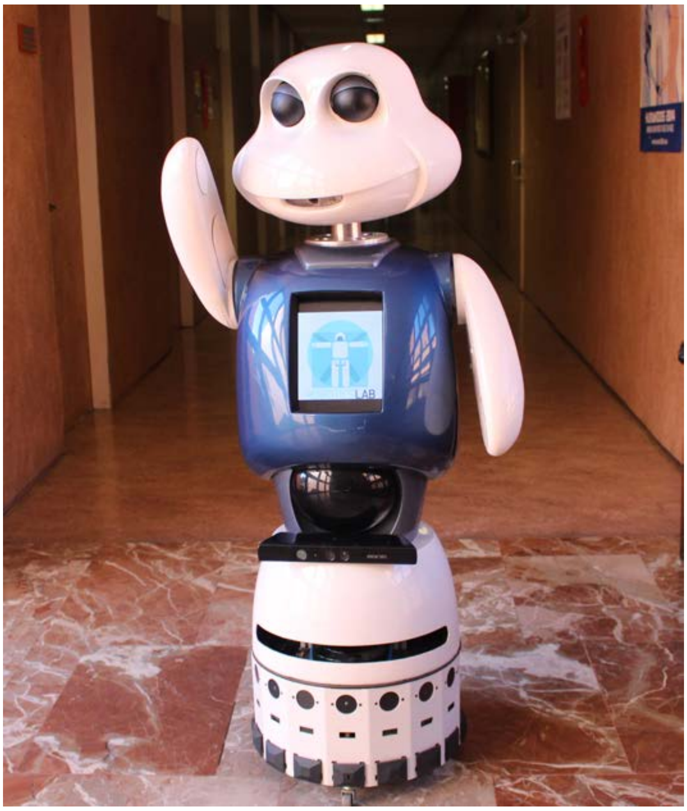


Figure 3.1: Maggie, a human-friendly robotic platform for social interaction research.

allowing turning left/right and up/down. The robot's head has two black eyes, a mouth shape, an invisible webcam, synchronised lights with the speech behind the mouth, and two mobile and controllable eyelids. In addition, two 1-DOF arms without end-effectors are built on both sides of the robot's trunk to provide nonverbal expressiveness through body movement. Most of the head, arms, and trunk material consists of a fibreglass curve-shaped shell. Therefore, according to Argall [22], it can be considered a hard-skinned robot. Maggie incorporates a tablet PC in the chest to provide audiovisual feedback and render images responding to tactile screen events. A Bluetooth-enabled wireless microphone and two speakers are connected to this tablet PC. the robot has a Text-To-Speech (TTS) system to speak Spanish.

In the upper half of the robot, hidden capacitive sensors work as tactile sensors. Each capacitive sensor has a range of $5cm^2$, approximately. The robot has one capacitive sensor on each shoulder, one on the head, two on the chest, two on the abdomen, two on the torso's back, and three on each arm. Maggie has three Ethernet-connected computers. Maggie is self-contained; all components (computer, tablet PC, sensors, cameras, microphone, speaker, etc.) are housed in its body structure. This robotic platform has been employed for the evaluation tests in Sections 5.1 and 5.2.

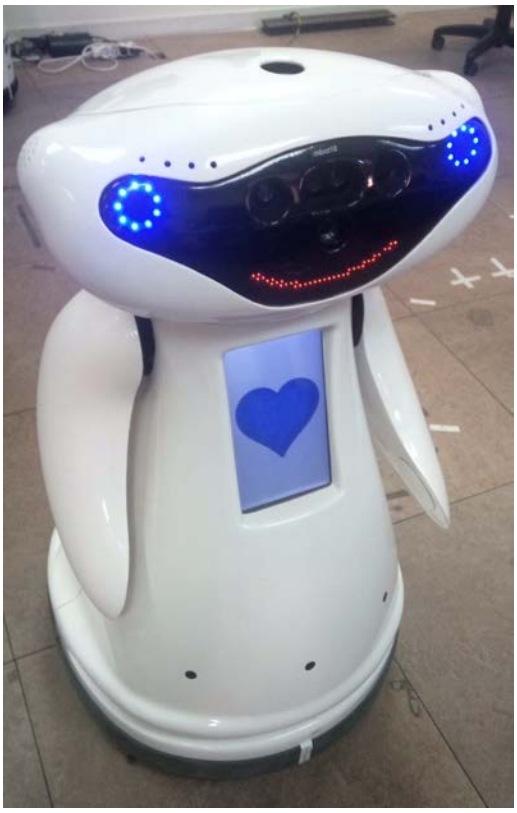


Figure 3.2: Mbot, a social robot created to interact with children, staff and visitors in a pediatric unit at a hospital.

3.1.2. Mbot

As mentioned earlier, the MOnarCH[3] project was an ongoing FP7 project that explored introducing social robots in real human social environments with people and studying the relationships that appear between robots and humans [138]. By establishing the pediatric ward of an oncological hospital as the case-study environment, this project's final objective was to introduce a team of robots in that environment that cooperatively engage in activities to improve inpatient children's quality of life. The robotic platform that resulted from this project is the Mbot, shown in Figure 3.2.

The Mbot platform is suitable for various applications that extend beyond the MOnarCH case study: by combining several high-level actuators and sensors, the platform may be utilised in office, household, and industrial settings [140]. The MOnarCH project addressed the link between autonomous and networked robotics and interfaces for human-robot interaction and expressive robots, having robots playing specific social roles, coping with the uncertainty com-

[3] Reference: FP7-ICT-2011-9-601033. Website: `https://cordis.europa.eu/project/id/601033`

mon in social environments, and interacting with humans under tight constraints. This translated into physical constraints on the robot platform, such as its maximum allowable dimensions and velocities, and behavioural constraints that can condition the techniques to control the platform, such as its navigation algorithms.

Some constraints that conditioned the robot's design were the ability to move naturally in its environment, with velocities in the same order as those used by humans moving around it. For this reason, the mobility of the robot was a critical issue. Based on this evidence, Mbot has an omnidirectional robot platform based on four Mecanum wheels to increase its manoeuvrability and performance. The omnidirectional base allows the robot to reach speeds of 2 to $2.5 m/s$. The robot's physical presence greatly influences how bystanders perceive the robot and its intentions. Children must perceive the physical dimensions of the robot neither as a menace nor as a physically diminished social entity. Since the average height of an under-teen (11 years) is around $145 cm$, this measure determined the maximum height of the Mbot. The volumetry of the robot is designed to avoid tilting under high accelerations or decelerations. To be more appealing aesthetically, the robot has a fibreglass shell —making it a hard-skinned robot— coated in white, with a softly curved shape.

Perception, navigation, interaction, environment, and low-level safety sensors are all included in the robot. The robot uses encoders to control the motors' velocity during locomotion. In contrast, an inertial sensor and a laser range finder are used during navigation to identify obstacles and the geometry of the surrounding space. The robot will use microphones, a depth camera for people tracking, face analysis, body gesture identification, and other technologies for perception and interaction. The robot will be fitted with temperature and humidity sensors for environmental sensing. Finally, low-level safety sensing is provided by the bumpers and sonar sensors. Unexpected collisions trigger can be detected at the hardware level, bypassing all decision levels to stop the robot. The robot contains various other sensors and methods to improve localisation's robustness, including RFID, IR, and UWB. The Mbot robotic platform has been the base for the tests that Section 5.3 describes.

3.1.3. Mini

The only robot that, as of this work's date, is currently active is Mini, depicted in Figure 3.3. Mini is a social robot designed to assist older adults and their carers in daily activities in nursing homes or at the users' homes. It was designed to be a tool for assisting physicians. Mini allows users to play games with it, request various multimedia content (music, photos, movies...), and request the news or the weather report. It also provides complete cognitive stimulation exercises and therapies. Mini, modelled after the robot Maggie, has an anthropomorphic shape, but its

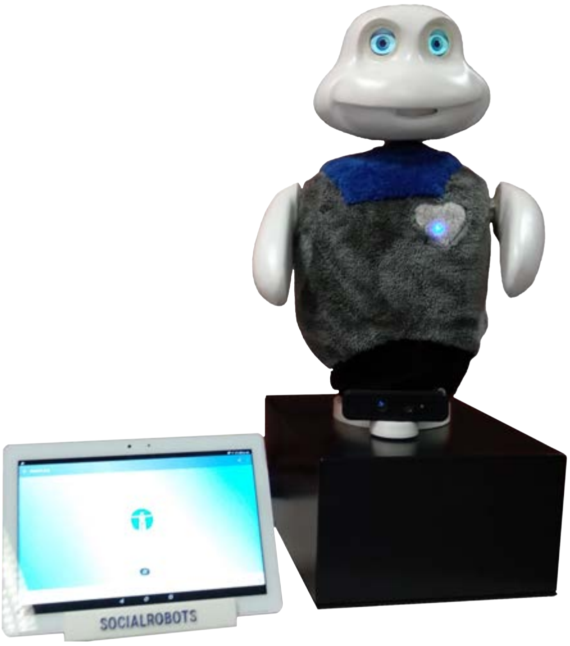


Figure 3.3: Mini, a social robot designed for assisting older adults suffering from cognitive impairment.

appearance is more cartoon-like. Mini's exterior design was created with the notion that it must be perceived as a living being rather than a machine.

Mini is a 50*cm* high desktop robot designed with the aid of expert feedback in the field of assistance care. Mini comprises two parts: the lower base, where most of the electronic components are placed, and its body. The main structural components of these two components were designed and produced in the laboratory using a 3D printer. The robot's main materials were acrylonitrile butadiene styrene (ABS) and polylactic acid (PLA). ABS has been used mainly inside the base and for the robot's torso since it is a material that supports higher temperatures. The parts that compose the robot shell, such as the arms or the head, and other mechanical parts located in the robot's head, were made of PLA. This choice was supported by the fact that these elements would suffer lower thermal and mechanical stresses. The torso 'skeleton' is covered with foam and a cloth vest over the foam, giving the robot a 'squeezable' appearance. To allow removing the vest, both robot arms are detachable, and they are held to the torso using a magnet placed in the joint between both robot's parts. Having this combination of soft and hard materials, Mini can be considered a *hybrid-skinned* robot.

The robot's internal skeleton contains the microcontroller, which controls the sensors and actuators. The robot's main computer is placed in the box that serves as the robot's base. An RGB-D camera's perceptual capabilities are used for user detection in short-distance human-robot interactions. A unidirectional mono microphone is used to record the user's verbal activity. In this sense, the robot has an Automatic-Speech Recognition (ASR) module for speech extraction. Mini also has capacitive touch sensors in its belly and shoulders for tactile interactions. Finally, the robot has an external touch screen that can interact with users through menus and display multimedia content.

Mini has multiple actuators that give the platform its expressiveness capabilities. Mini's body has five degrees of freedom: two in the neck, one on each shoulder, and one on the waist. The robot can move these joints using servomotors controlled in position or velocity. Coloured LEDs are placed in the robot's cheeks and chest, enabling Mini to express its internal state. An LED array in the mouth is synchronised with the robot's audio output, functioning as a volume unit (VU) meter. Mini also has a chest LED that simulates the robot's heart, and its intensity, heart rate, and colour are modulable. In its face, the robot has two screens representing the robot's eyes. Through these screens, Mini can show eye expressiveness by displaying a series of predefined GIFs. Finally, a speaker in its chest allows Mini to emit verbal and non-verbal sounds. The robot's speech is generated using a TTS module.

Since Mini is the most active platform in the Social Robotics Group, being in constant improvement, it has been the platform that had the most influence in the design of some of the hardware elements of the proposed acoustic touch system. Therefore this robot is the platform that appeared the most in this work. Mini participated in the evaluation tests from Sections 5.2 and 5.4 and in the experiments from Chapter 6 and 7. The content related to the Mini robot from this subsection has been published in the following journal publication.

Publication

Salichs, M. A., Castro-González, A., Salichs, E., Fernández-Rodicio, E., Maroto, M., Gamboa, J.J., Marques-Villarroya, S., Castillo, J. C., Malfaz, M., Alonso, F. (2020). "Mini: A New Social Robot for the Elderly". *International Journal of Social Robotics*, 12, 1231-1249. (Q1)

3.2. Hardware Integration of the Acoustic Touch Recognition System

This section offers insights into how the system's elements proposed in this work are integrated at a hardware level in the robotic platforms described before. This implies identifying the key components of such a system because they will be present in all of the system's iterations. Each of these integrations shows how the system has changed in terms of hardware, allowing a timeline of the work to be created. They have also been tailored to the needs and constraints of the robotic platform on which the system is installed. Several design criteria, such as the price of the various system components, their volume, weight, or the portability they provide, have also impacted this evolution. Additionally, the design had to be modified in some instances to fit the research topic in question, or in some cases, due to the evolution of the technologies involved in the setup.

Acoustic sensors, specifically piezoelectric microphones, also known as piezoelectric (or 'piezo') pickups or contact microphones[4], will be the system's main components. The microphones will be connected to a soundcard or a sound interface, depending on the experiment or setup to which they correspond. Another relevant element, as we will explain in the next chapter, are the capacitive touch sensors. In this case, capacitive sensors will only appear as part of the proposed STS system if they are specifically installed or if any changes are made to those already installed on the robotic platform involved in the setup. As a result, this section will only cover elements that have been explicitly installed on the robot.

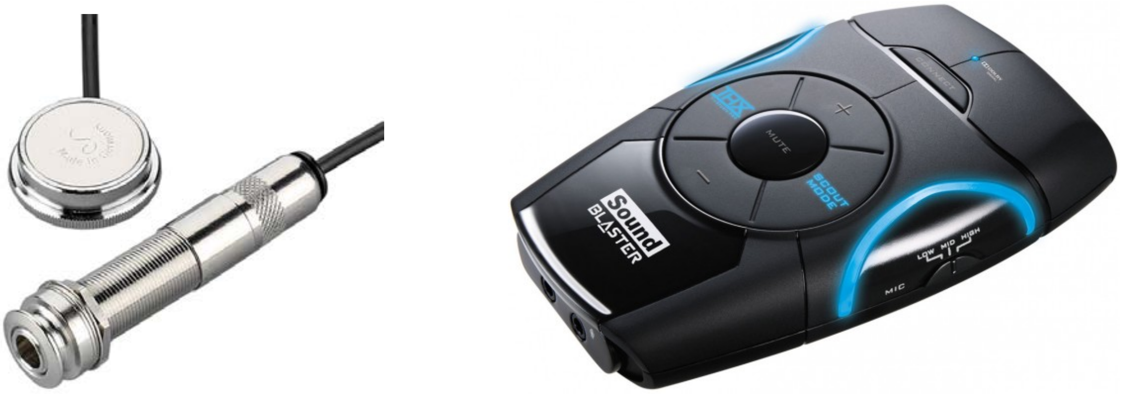
3.2.1. Integration in Maggie

As we mentioned, we used contact microphones as the primary element of the system. The main reason to use this device on a solid surface to perceive touch is that sound vibration propagates much better in solids than through air. This property is important since a contact microphone can perceive slight touches in the material and would be less affected by environmental noises, such as the human voice. Maggie was the first platform in which the system was tested. Since the integration in this platform was a proof of concept, the first step was choosing a contact microphone with high-quality sound acquisition.

We aimed for a more professional device choice in this setup iteration. For this reason, we opted for an *Schaller Oyster S/S*[5] contact microphone, shown in Figure 3.4a. This contact microphone consists of a polished and chromed oyster-shaped piezoelectric pickup with a chrome silver cover pre-wired to a standard instrumental cable. This device provides advantages, such as requiring no active circuitry or pre-amplification. It also presents a *resistance* of 13.1 KOhm, an *inductance* of 6.4 H, and a *maximum detectable resonance* of 15 dB. Despite these advantages, this contact microphone, as it can be seen, is considerably bulky and would require an adapter

[4] Along the manuscript, the terms piezoelectric microphones, piezoelectric or piezo pickups and contact microphones are used interchangeably.

[5] Schaller Oyster microphone website:
https://schaller.info/en/megaswitches-preamp-pickups/410/oyster-s/s?c=19



(a) Schaller Oyster contact microphone.

(b) Sound Blaster Recon3D USB soundcard.

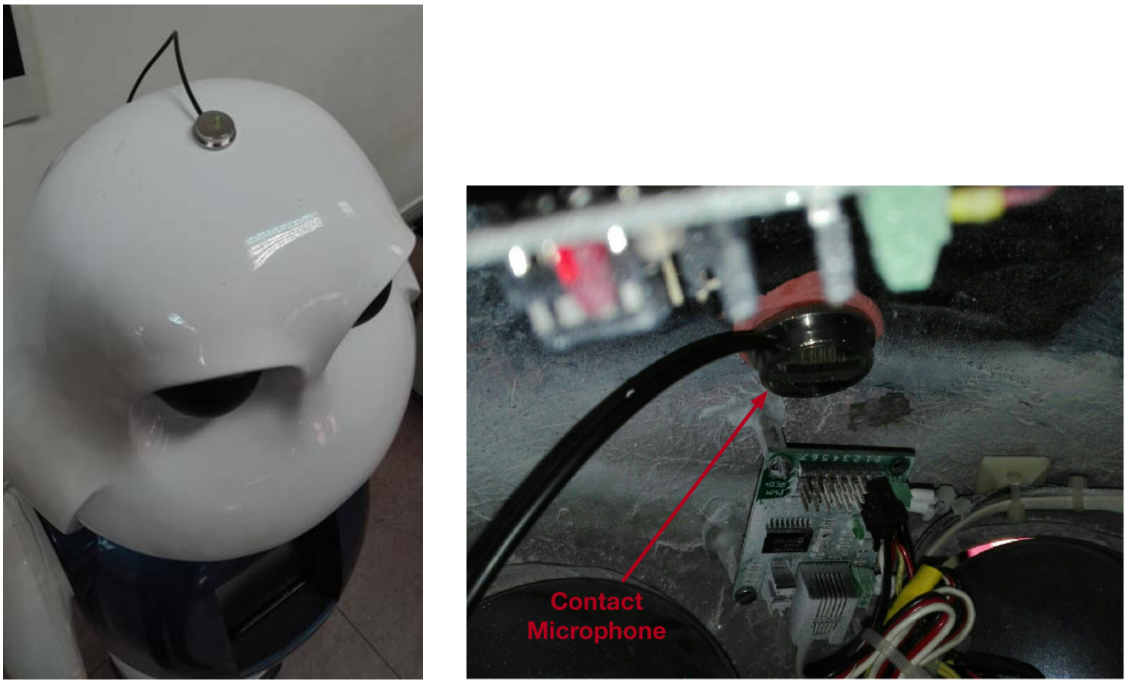
Figure 3.4: Hardware elements from the setup in the robot Maggie.

to convert the more music-oriented 6.35mm Jack to a 3.5mm adapter that could be connected to a USB soundcard.

The soundcard choice, as it happened with the microphone, was related to good sound acquisition and performance rather than minimising the setup cost or total size. The soundcard of choice, in this case, was a *Creative Sound Blaster Recon3D USB*[6]. The Recon3D is an external soundcard, connected by USB rather than the PCI or PCI-E bus, that has a four-core design that combines a digital signal processor (DSP), digital-to-analogue converters (DACs) and analogue to digital converters (ADCs) that enable it to handle audio as ably as its PCI-E, resistor-covered soundcards. The soundcard specifications are a sample rate of 48 kHz and a bit depth of 24 Bits. The device is shown in Figure 3.4b.

Regarding the installation of the microphones, the first iteration consisted of placing a single microphone over the robot's shell, more specifically, in the head. Compared with the inner side of the fibreglass, the outer side was smoother due to the coating and had more places to adhere to the piezo microphone. This is shown in Figure 3.5a. Despite this, the main problem was related to the fact that the microphone's position or the cable could interfere with the touch interaction. For this reason, we explored placing the pickup beneath the surface, in the inner shell of the robot's head (see Figure 3.5b). Because the internal part of the shell is concave and rough, it was necessary to use clay to achieve a smooth and homogeneous surface to maximize the contact between the microphones and the shell. Since as we mentioned before, a smooth fitting between the microphone and the shell is crucial to achieving good sound acquisition.

[6] Sound Blaster Recon3D website: https://support.creative.com/Products/ProductDetails.aspx?prodID=20835&prodName=Sound+Blaster+Recon3D



(a) Microphone placed at the top of Maggie's head.

(b) Microphone placed beneath the top of Maggie's head.

Figure 3.5: Different setups for the contact microphone placed in Maggie's head.

This setup, with a single contact microphone placed beneath the robot shell, was used for the tests described in Section 5.1.

Following the demonstration of the system's capabilities in tactile gesture classification, shown later in Section 5.1, we included two contact microphones on the robot's chest. More specifically, the microphones were placed on the left and right shoulders, again under the robot's shell, leaving the setup as shown in Figure 3.6. At this point, it should be noted that each soundcard had only one input port, so increasing the number of microphones implied increasing the volume of the tactile system. Although this could be a problem, as we will see later, in the case presented in this section, the elements had plenty of room inside the robot's casing. Furthermore, the robot has a compartment on its back where the sound cards can be inserted and easily connected to its computer. The volume and cost of the setup were a significant problem in cases like the one we'll see in Section 3.2.3, with a desktop robot like Mini and several microphones. Maggie's tests with multiple microphones are part of the content described in Section 5.2.

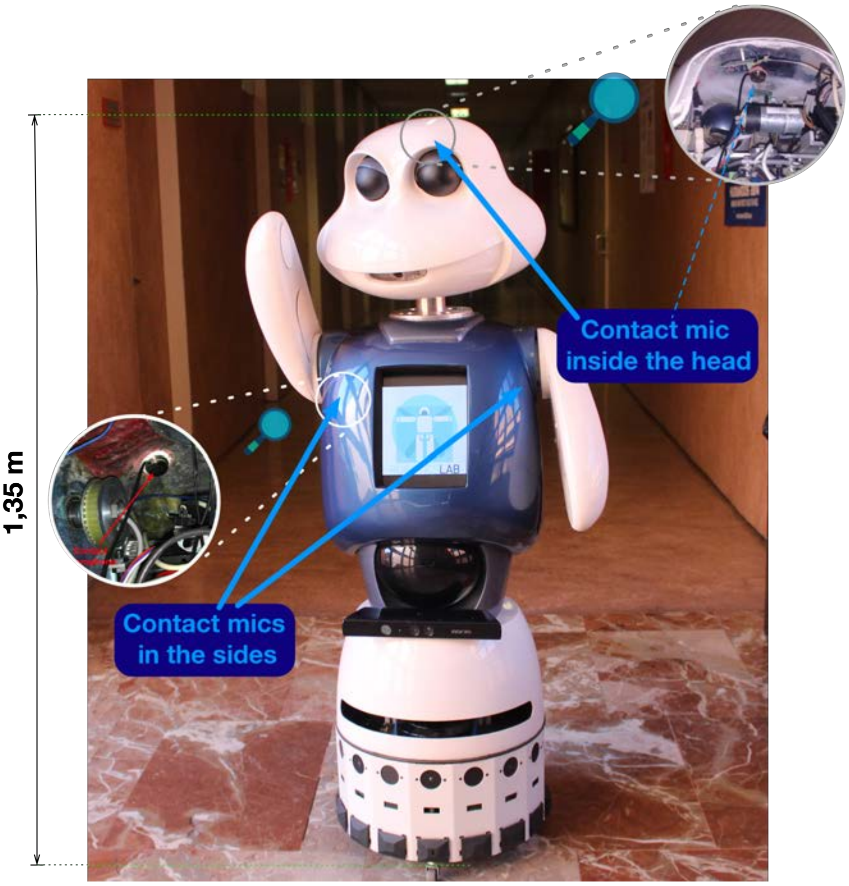
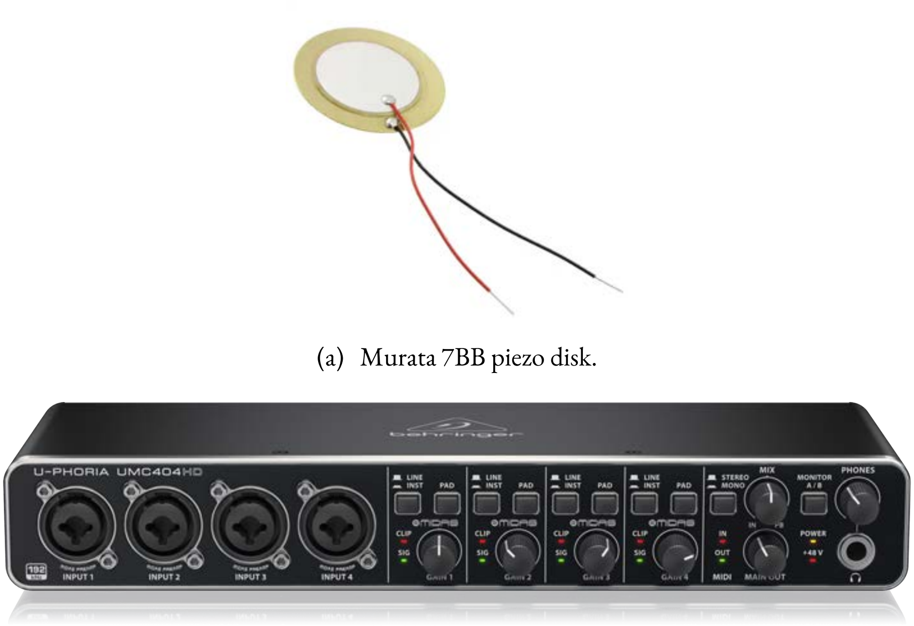


Figure 3.6: Setup of the microphones in Maggie.

3.2.2. Integration in Mbot

The next platform where we integrated a version of the system was the robot Mbot. For this setup, the main objective was to analyse the sound signal obtained by a set of microphones to localise the source of the sound with precision. This requirement made us shift from a single soundcard per microphone to a more sophisticated sound interface that could allow higher sampling frequencies. A high sampling frequency would allow us to measure the time difference between audio signals perceived by multiple microphones when users touch the surface where the devices are installed. The location of the origin of physical contact could then be determined using the difference in the arrival of the signal at the different microphones and other features derived from this difference. More specifically, in this work, we propose a case study that aims to locate touches on the head of the robot platform.

The common elements with the previous approach are mainly that the system used again an array of microphones attached to the inner part of the robot's outer shell. For the piezoelectric microphones, we opted for a more cost-effective and lighter option to detect sound propagation



(a) Murata 7BB piezo disk.

(b) Behringer Uphoria UMC404HD sound interface.

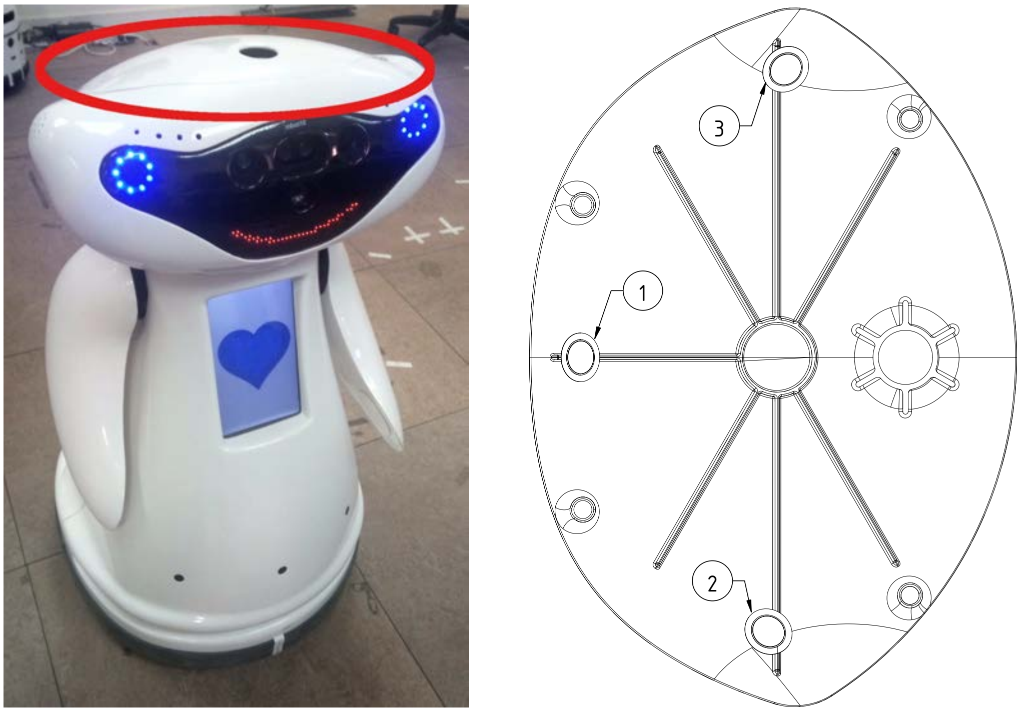
Figure 3.7: Hardware components from the Mbot setup.

through a hard surface in this setup. The device, in this case, is a Murata piezo disc[7], shown in Figure 3.7a. This kind of receiver fits in this project because they are inexpensive and compact, and collecting observations from these sensors incurs little additional energy cost to the system. In addition, they are light and relatively easy to attach to any surface. For this setup, we also opted for a light adhesive to install the microphones on the surface, allowing easy removal or replacement of the sound receivers.

In this system implementation, a professional audio interface provided the system with an increased sampling rate than the previously used soundcards. Sound propagates significantly faster on solid materials than on the air; for this reason, the sound interface of choice would require frequencies higher than the 48 kHz that the Sound Blaster offers to perceive time differences between microphones installed on this kind of material. We proposed the *Behringer Uphoria UMC404HD*[8], a cost-efficient 4-channel interface that meets this requirement (shown in Figure 3.7b). Its specifications include a 4x4 USB 2.0 Audio/MIDI interface with MIDAS Mic Preamps, a bit depth of 24 Bits, and a sample rate of 192 kHz.

[7] Murata piezoelectric microphone https://www.murata.com/en-US/products/productdetail?partno=7BB-15-6

[8] Audio interface: https://www.behringer.com/product.html?modelCode=P0BK1



(a) The SO model of the MOnarCH project's robotic platform. The red ellipse marks the zone where the system is installed.

(b) Schematic of the piezo microphones setup in Mbot's head cover.

Figure 3.8: Experimental setup in the Mbot robot.

In this case, the platform would be the Mbot robot platform [140], specifically the SO model (shown in Fig. 3.8a). The top cover of the robot's head will be used for this experiment, as it is a slightly curved area and is considered prone to physical contact. The microphones are attached to the robot's surface as shown in Figure 3.8b. More specifically, they are installed on the inner side of the surface to avoid interfering with physical contact and not to alter the robot's appearance. This setup includes adhesive putty to ensure perfect contact of the microphone with the robot shell's rough and irregular inner surface. The touch localisation experiments performed in the Mbot are contained in Section 5.3.

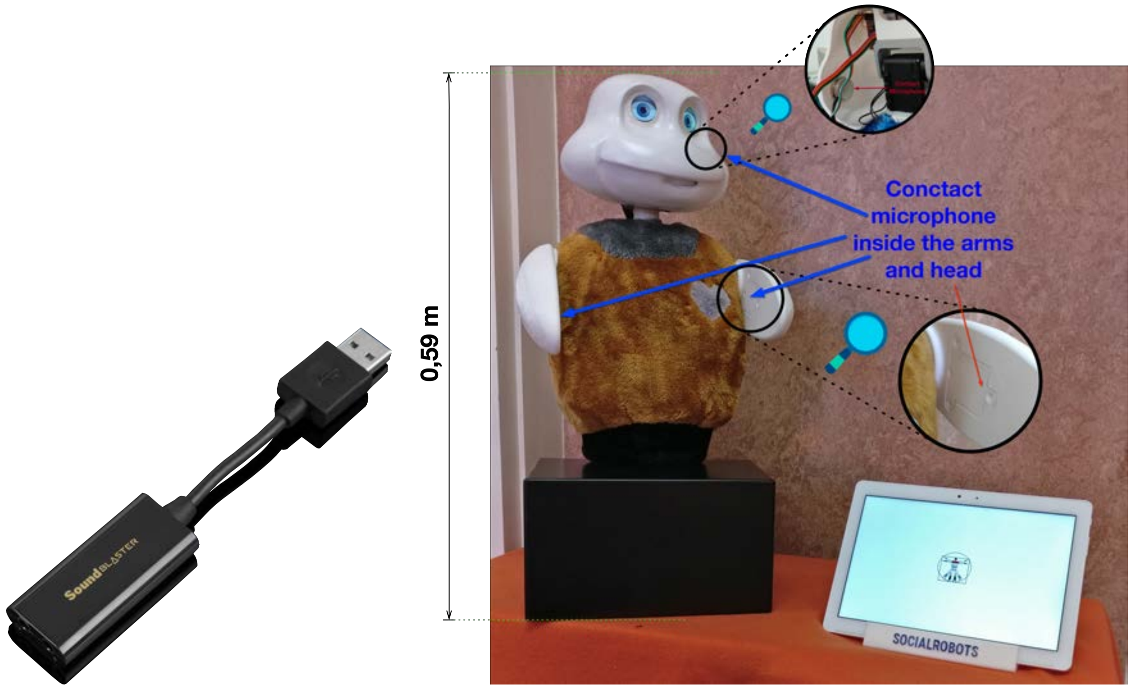
3.2.3. Integration in Mini

The last robotic platform where the system has been integrated is Mini. This robot is also the platform that has introduced more changes to the system since it is a robotic platform currently in development. Furthermore, this last fact also implied that integrating the system into the robot has motivated some design changes in the platform. However, we have to clarify that some of these changes were driven by the desire to obtain the best performance on this robotic platform and to explore the possibilities that acoustic technology offers when installed in the materials Mini is composed of. Therefore, most of these changes did not involve changing the core aspects of the robot. Regarding previous iterations of the system, the setups we present in this subsection are more related to Maggie's setup than Mbot's. As we've explained, the work carried out in Mbot corresponds to a particular use case where the shape of its surfaces and the robot's dimensions play an important part. Furthermore, the integration in the robot Mini started practically since the whole work present in this manuscript started. In contrast, the tests carried out in Mbot occurred during a fixed time period.

The first constraint that conditioned the system design was the robot's dimensions. Mini is a desktop robot whose height does not surpass half a meter. Therefore, this platform does not have a lot of extra space for installing the receivers and hiding the soundcards. As we mentioned in Section 3.2.1, one of the flaws that the setup in Maggie had was its size. Also, the element that occupied more space were the soundcards, and for this reason, in this setup, they were the first element to change between both setups. We opted for a more compact, light, and cost-effective solution without changing the sound quality. For this reason, we opted for a more modern soundcard from the same brand. The *Creative Sound Blaster PLAY! 3*[9], shown in Figure 3.9a, had the same specifications as the *Recon3D* but in a more compact format and half its cost. Concerning the contact microphones, the *Oyster* microphones were located on the robot's rigid surfaces: the head and both arms. The main reason to discard Mini's shoulders was that Mini's torso is made of foam. For the first experiment in this platform, we wanted to evaluate the performance using one type of material. As Figure 3.9b shows, in the head, the microphone was placed on the robot's left cheek. To install the remaining two microphones, we designed and built compartments in the forearm area of the arms. This setup is used for the system evaluation from Section 5.2.

The Mini setup's next iteration involved evaluating the system's effectiveness on combined materials. In this case, PLA is for the arms and foam for the robot's trunk. The issue arose when attempting to combine the foam and the piezoelectric microphone because the weight of the Schaller Oyster prevented a proper bond between the two elements. As a result, it was decided to use the Murata piezoelectric discs, which had already been used for a similar purpose in the Mbot setup in Section 3.2.2. In the case of the arms, the microphones would be placed in the same locations, but a small slot in the foam would be created to introduce the piezo in this material. In this manner, the receiver was also properly secured and pressed against the surface, enhancing sound acquisition. The complete setup is shown in Figure 3.10a. Furthermore, the

[9] Sound Blaster PLAY! 3 soundcard: https://es.creative.com/p/sound-blaster/sound-blaster-play-3



(a) Sound Blaster PLAY! 3 soundcard.

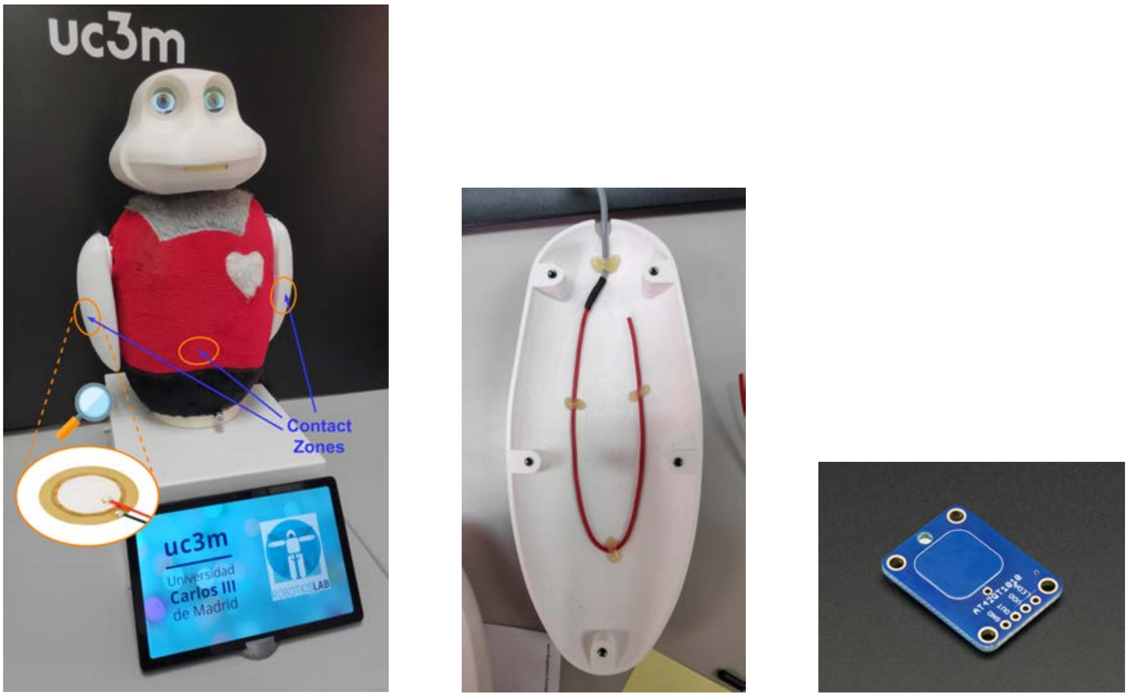
(b) Microphone setup in the robot Mini.

Figure 3.9: Experimental setup in the Mini robot.

capacitive sensors of the robot were modified in this setup to cover the inside of the robot's arms beside the shoulders and belly. Mini is equipped with *Standalone Momentary Capacitive Touch Sensor Breakout - AT42QT1010*[10] from Adafruit as its capacitive touch sensors (show in Figure 3.10c). These sensors enable the sensing surface to be extended by connecting different materials to a signal port. In our case, we chose a cable that extends inside the arm, as shown in Figure 3.10b. In the case of the belly, we used the same procedure to expand the capacitive sensing area. This setup was used for the tests that appear in Section 5.4 and Chapter 7.

Lastly, in this section we present an alternative to the *Murata/Sound Blaster PLAY! 3* combination. Although this setup iteration provided a good balance between compactness, affordability and performance, it still presented some restrictions. First, the input microphone port consists of a Jack connection, which in our case, was unnecessary because the soundcard would only be connected to the piezo. In addition, the piezo microphone cable we used for this receiver, the longest cable in this case, could emit noises when rubbed or struck. Finally, the cable was relatively thin, so it might break if submitted to excessive stress. Due to these factors, we decided to explore an integrated solution. Although these issues did not appear during the experiments in Chapters 5 and 6, they might arise in long-term scenarios.

[10] Capacitive sensor webpage: `https://www.adafruit.com/product/1374`



(a) Microphone setup in the robot Mini.

(b) Capacitive sensor cable beneath Mini's arm.

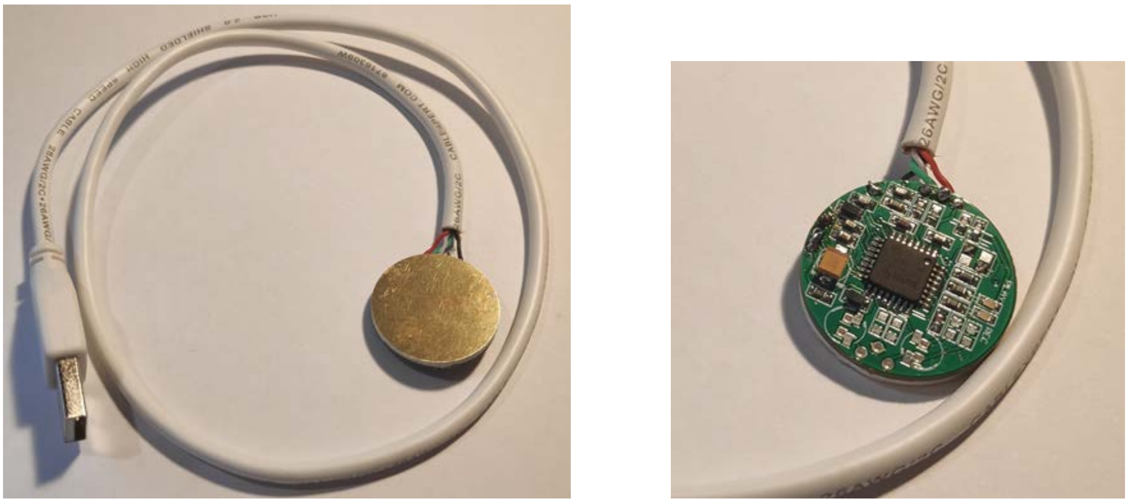
(c) Adafruit capacitive sensor.

Figure 3.10: Experimental setup in the Mini robot.

This resulted in the device depicted in Figure 3.11, an option designed and developed in the laboratory that incorporated the soundcard and contact microphone in the same place. It was based on the *PCM2912APJTR Audio CODEC* interface[11], from Texas Instruments. This way, we could employ a single device with only a USB cable, more resistant to stresses resulting from the joint's continuous movement. As Table 3.1 shows, we could achieve better robustness and compactness without increasing the cost of the setup. We have to note here that, although preliminary tests of the system revealed its successful performance and compatibility with the previous system's datasets, this iteration has not been included in the experiments presented in this work in Chapters 5 and 6.

Throughout this section, we have demonstrated how the system's compactness improved, but the changes in the components also affected the setup's cost. The evolution of the acoustic sensing setup cost is depicted in Table 3.1. As the table shows, the cost of the setup decreased progressively with each iteration.

[11] PCM2912APJTR webpage: `https://www.ti.com/product/PCM2912A/part-details/PCM2912APJTR`



(a) Piezo side.

(b) Zoom of the soundcard side.

Figure 3.11: Soundcard based on the PCM2912APJTR Audio CODEC interface assembled.

Table 3.1: Comparison between the costs of each setup regarding acoustic sensing.

Robot	Microphone	Soundcard/Interface	Mic amount	Soundcard/ Interface amount	Mic cost per unit (€)	Soundcard/ Interface cost per unit (€)	Setup total cost (€)
Maggie	Schaller Oyster	Creative Sound Blaster Recon3D	3	3	37	80	351
Mbot	Murata 7BB	Behringer UPHORIA UMC404HD	3	1	0.77	133	135,31
Mini	Murata 7BB	Creative Sound Blaster PLAY! 3	3	3	0.77	19.99	62,28
Mini	Murata 7BB	Texas Instruments PCM2912APJTR	3	3	0.77	20.49	63,78

3.3. Software Architecture

In this section, we provide a detailed outline of the software architecture where the STS system proposed in this work integrates. Our research group used the Robot Operating System (ROS) framework to develop the software architecture. ROS offers a number of benefits, including flexibility, scalability, and modularity. ROS's robust and simple mechanisms make it possible to isolate each robot function in a different package or project without affecting the architecture as a whole. Conceptually, ROS uses nodes as computation-performing processes (functionalities), services for sending synchronous information between nodes (clients-service), and topics for acting as communication channels between nodes. Figure 3.12 gives a general view of our robots' architecture. It consists of a multimodal HRI system (composed of three modules, Perception manager, HRI manager and Expression manager), a Liveliness module, a

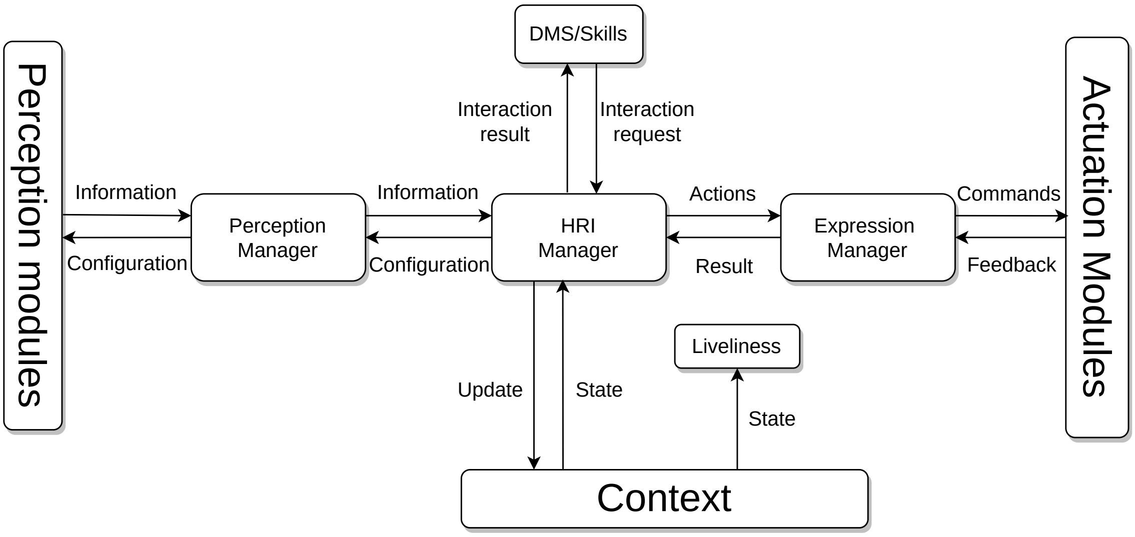


Figure 3.12: General view of the architecture integrated into the social robots of the Social Robotics Group.

memory that stores data about the robot and the user (the Context module) and a Decision-Making System (DMS) with skills the robot can perform. The perception and actuation modules contain detectors connected to the sensors and drivers connected to the actuators.

The multimodal HRI system manages the communication between the robot and users. The system comprises three main elements: the Perception manager, the HRI manager and the Expression manager. The perceptual and expressiveness capabilities of the robot are controlled respectively by the Perception manager and the Expression manager, and the HRI manager controls the flow of the dialogue based on the information captured by the robot's sensors. In short, these modules connect and allow communication between the robot's sensory and actuation devices with the rest of the modules that handle human-robot communication.

The Liveliness module is meant to enhance the actions of the rest of the architecture to make the user perceive the robot as a living being. The Context module allows the robot to adapt its communication and interactions by storing relevant data. This data includes information about the robot and the various users who communicate with it. Finally, the robotic platforms described before can carry out a wide range of *Skills*. *Skills* are flexible, modular applications that can easily be added to and removed from robots. In our architecture, a Decision Making System (DMS) manages controls and orchestrates the execution of the *Skills* [141].

Because these are the elements with which our system interacts most directly, in our work, we will detail the elements of the multimodal human-robot interaction system, especially the perception manager, the module connected to the detectors.

3.3.1. Perception Manager

The Perception manager is the multimodal HRI system module responsible for controlling all the input information processing [142]. Its primary function is to receive raw data from each robot's sensor and produce a unified message that the other system modules can understand; more specifically, it filters, formats and packages the information according to different criteria (type of data, connections, time window, etc.). The Perception manager has three abstraction levels: 0, 1 and 2, called respectively Translation, Aggregation and Fusion.

- Level 0, or Translation, comprises modules that take information from specific sensors or perception units and convert it into a standardized format. This format consists of an array of key-value pairs, where the keys are tags that identify the information being sent, and the values are numerical or discrete values associated with those tags. Perception units are intermediary layers that handle data preprocessing.
- Level 1, or Aggregation, takes in information from Level 0 and combines it to create a single message that contains relevant perception information from one or more perception units over a period of 1 second.
- Level 2, or Fusion, consists of high-level modules that take in information from the previous levels and combine it to create more complex, high-level messages that integrate information from different sensors. The goal of this level is to enhance the sensory information available to other modules of the robot, so that future decisions can be based on more reliable data and avoid potential perception errors.

These three levels communicate directly with the HRI manager to inform it of changes in the environment that the robot's sensors have detected.

3.3.2. Human-Robot Interaction Manager

The HRI manager is located at the core of the HRI architecture and is the module that ensures the success of human-robot interactions by processing the different Communicative Acts (CAs) requested or given by the robot or the user. CAs are the atomic units into which we can decompose a dialogue or message and are the basic actions used to model any human-robot interaction. They are communicative components that the user can use to ask for or receive information. Since they offer a consistent and distinctive channel through which information about the user flows, these units play a significant role in abstracting and simplifying the communication process with the user. According to whether the robot or the user initiates the information exchange and the direction of that communication, the CAs can be categorised as *robot asks for information*, *robot gives information*, *user asks for information*, and *user gives information* [143].

3.3.3. Expression Manager

The Expression Manager receives orders based on sensor information from the HRI manager to control the actuation capabilities of the robot for communicating. It also receives orders from the Liveliness module, which generates spontaneous actuation commands to make the robot's movements more natural, especially when the robot is not performing a specific task, enhancing its expressiveness. The primary task of the Expression Manager is to decompose complex expressions and gestures into individual actuation commands and send them to the correct actuators at the appropriate time to ensure the continuous adaptation of the robot's expressiveness to its surroundings.

3.4. Summary

This chapter contained different elements that were part of the ATR system setup during its development. First, we present the different social robotic platforms in which the system has been tested Maggie, Mbot and Mini. The first two, Maggie and Mbot, are two 1.5m tall robots covered with a fibreglass shell. The last one, Mini, is a desktop robot that contains both soft and hard materials. These agents and their primary functionalities will be described in the first section of the chapter.

Afterwards, we described the evolution of the ATR integration in each social robotic platform. This section defines a timeline that shows how the system's hardware has evolved from its conception to its current state. Each of these setups was composed by the same elements: one or a group of piezoelectric microphones and a soundcard or sound interface. We started from the first prototype integration in the robot Maggie, using more expensive and bulkier elements, to gradually move on to lighter and more cost-effective components present in the robot Mini. We connected each of these setups with the corresponding later sections, each time the setup is part of an experiment, to serve as a reference to the user

Finally, in the last section of this chapter, we defined and explained the software architecture from the Social Robotics Group that is present in its platforms. We focused on the core elements that belong to the HRI system: Perception manager, HRI manager and Expression Manager. These elements are the ones the system will interact with the most.

Chapter 4

Towards an Integrated Acoustic Touch Recognition System

The previous chapter described the system's hardware, which is the first step in clarifying the topic of this chapter: the design and implementation of the system's software aspects. The structure is separated into three distinct blocks, with the tactile gesture classification system constituting the first. The next part describes the design of the module for touch gesture localisation. Finally, the integration of the system in a online setting is explained.

The ATR is a system primarily designed to recognise and localise touch gestures made as a consequence of the contact of a human over solid surfaces. In our case, we propose using of our system over the surfaces of different social robotic platforms. This system includes a novel application of piezoelectric pickups in social robotics, specifically human-robot touch contact. A piezoelectric pickup contains a piezo crystal, which converts the vibrations directly to a changing voltage. These devices can detect the sound vibrations generated when a user touches a surface, in this case, the robot's shell. The working principle is based on the fact that the perturbations induced by a physical contact over a surface propagate through the robot's rigid parts (its shell and inner structure), leaving a distinct wave signature that allows the system to identify the type of contact and its location on the robot's surface. In contrast to other approaches described in the literature on touch recognition [54, 59, 60], the presented system employs a small number of sensors.

The system draws inspiration from Human Activity Recognition (HAR) systems that rely on inertial sensors. Activity recognition aims to recognise the actions and goals of one or more agents from observations of the agent's actions and the environmental conditions. In the works we are interested in, activity recognition is accomplished by utilising data from inertial sensors such as accelerometers [144, 145]. Our system focuses in the real-time signal processing phase of these systems. More specifically, both our system and these proposals share the capability of sampling, processing, and storing the features of a continuous signal in real-time. The signal, in their case, is the acceleration collected by the inertial sensors, while ours is the sound signal from the microphones. In other words, information taken from a microphone, including sampling and windowing processes, might similarly be extracted and processed from an accelerometer or other HAR device [146–148]. Another distinction between this type of system and the one described in this section is that HAR systems usually focus on a person performing an activity. In contrast, the core reference point in our scenario and the element carrying the sensors is a robot. Another significant difference is the lack of a large sample corpus to generate automatic learning models [149]. Because of this detail, the system presented in this chapter explicitly includes the dataset generation phase.

This chapter describes the different elements composing the system. The first section covers the touch gesture recognition block of the system. Next, we detail the touch localisation modules. We proposed two approaches to solve the touch localisation problem. On the one hand, we introduced machine learning algorithms, and on the other hand, we proposed sound analysis techniques to provide a more precise localisation. Finally, the last section details how we integrated the system to run online on the robot.

4.1. Touch Gesture Recognition System

The Touch Gesture Recognition system constitutes the core of the ATR since the system's primary purpose is to recognise how a surface is touched. This section discusses the many stages of our touch gesture recognition system, including Sound Signal Acquisition (SA), Touch Activity Detection (TAD), Feature Extraction (FE), Dataset Creation (DC), and Touch Gesture Classification (TC). Figure 4.1 depicts a summary of the operation flow. Touches done on the robot's exterior shell are received by all of the robot's contact microphones, implying a parallel analysis of the sounds collected by each pickup. Once the system perceives a touch gesture due to a sudden signal change, it computes the significant values of the audio features until the gesture ends. Finally, all the characteristics generated in each pipeline are merged in instances. This process provides a dataset that can subsequently be used to train multiple families of classifiers

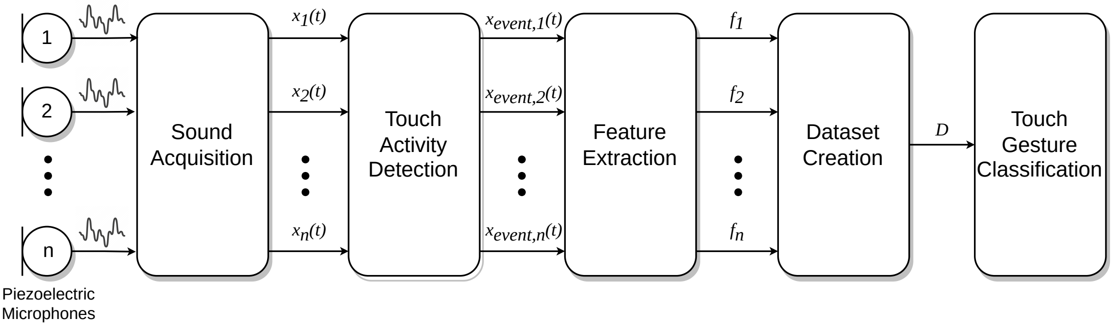


Figure 4.1: Pipeline of the Touch Gesture Recognition system. Touches done on the robot's exterior shell are received by all of the robot's contact microphones, requiring a parallel analysis of the sounds collected by each pickup ($x_i(t)$). Once the system perceives a touch gesture ($x_{event,i}(t)$), it computes the significant values of the audio features (f_i) until the gesture ends. Finally, all the characteristics generated in each pipeline are merged in instances. This process provides a dataset D that can subsequently be used to train multiple families of classifiers to recognise touches.

to recognise and locate touches in the Touch Gesture Classification and Localisation (TCL) phase (see Subsection 4.2.1).

4.1.1. Sound Acquisition

The first phase of the system corresponds to *sound acquisition*. In this phase, the system must extract the signals coming from the piezoelectric microphones. Each microphone will be connected to the robot via an external soundcard. Because the system requires a set of microphones, the first stage is for the system to be able to identify each microphone uniquely. Since the robotic platforms employed in this work run Linux-based operating systems, these events will be configured using Userspace '/dev' (UDEV) rules. The UDEV is the Linux subsystem that provides a computer with device events, allowing it to detect when a device connects and associate events with those connections. Labelling the soundcards allows unique configuration profiles to identify the different devices, which allows tuning parameters such as each soundcard's gain. This information is saved in a configuration file written in YAML Ain't Markup Language™ (YAML), a human-readable data-serialization language.

Advanced Linux Sound Architecture (ALSA)[12] is a software framework that provides an Application Programming Interface (API) for sound card device drivers. Due to ALSA limitations concerning audio routing, the system is also compatible with PulseAudio[13], a general-

[12] https://www.alsa-project.org/wiki/Main_Page

[13] https://www.freedesktop.org/wiki/Software/PulseAudio/

purpose sound server designed to run as a middleware between applications and hardware devices. The compatibility with both sound systems makes the interface adaptable to practically any Linux/UNIX-based operating system since ALSA is present in almost all of them. The interface has to implement the i*asound* and *pulse* C libraries to allow the system, through communication with either ALSA or PulseAudio APIs, to adjust the various settings of the sound system. Because these libraries are written in C, the interface has been designed in C++ to take advantage of its more modern libraries, backwards compatibility with C and its implementation of the Object-Oriented Programming (OOP) paradigm.

Before implementing our audio processing system, responsible for extracting and computing the sound features of our interest, we were looking for a sound analysis framework having two basic requirements in mind: i) we wanted a tool able to work in **real-time** having as short a delay as possible, and ii) the audio processing task needed to work in **three audio domains**: time, frequency, and time-frequency. In the **time domain**, the amplitude of a signal is measured as a function of time. It is a straightforward representation of a sound as a continuous vibration. On the other hand, the **frequency domain** displays how much of the signal exists within a given frequency band concerning a range of frequencies, providing extra information about the signal. Lastly, we can look at audio's spectral (frequency-based) content over time when combining both domains. This combination is the signal's **time-frequency** domain, which also supplies valuable data regarding the sound signal. Previous work has already explored the performance of systems that extract features from these audio domains in the context of voice recognition [150–152].

Software like *Praat*[14] or *CSound*[15] 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*[16] or *Octave*[17], but in both cases, they are oriented towards prototyping, and not for designing self-contained, high-performance applications. For all these reasons, we finally chose *ChucK*[18], a versatile audio processing programming language traditionally used by musicians and digital artists oriented towards real-time sound processing.

[14] http://www.fon.hum.uva.nl/praat

[15] https://csound.com

[16] https://www.mathworks.com/products/matlab.html

[17] https://octave.org/

[18] http://chuck.cs.princeton.edu

4.1.2. Touch Activity Detection

This phase runs independently for each microphone. The TAD software consists of a series of scripts or nodes developed in ChucK. The node performs a series of tasks for each microphone to identify when the contact has occurred. First, the acoustic signal samples are stored in a buffer in real-time using sliding windows, which is a popular way for signal processing jobs. Sliding windows are particularly useful for determining transient events and averaging frequency spectra over time. Each application's requirements, such as those for time and frequency resolution, determine the length of the segments. But that method also changes the signal's frequency content by an effect called spectral leakage. Spectral leakage is a smearing of power across a frequency spectrum that occurs when the signal being measured is not periodic in the sample interval. Depending on the application's requirements, window functions allow the distribution of spectral leakage in various ways. Therefore, we decided to implement a Hann window function, which, in addition to being widely used, is appropriate for sampling signals containing vibrations, as in this case [153].

After obtaining real-time sound samples, the system evaluates whether the signal has changed abnormally from one instant to the next to identify physical contacts. There are multiple ways to evaluate such spontaneous transitions, and they are closely related to Voice Activity Detection (VAD) techniques [151, 154]. These techniques include, among others, *spectral subtraction* [155] or the evaluation of variations in certain characteristics of the signal [151]. For this work, we opted for the second option as it is considered less computationally expensive in a real-time scenario. More specifically, this phase is based on the VAD phase from the Gender and Emotion Voice Analysis (GEVA) system [150, 151]. The feature of choice for this proposal is one of the most commonly used in these cases: the Signal-to-Noise Ratio (SNR) of the signal. This ratio compares the level of the desired signal to the level of background noise. SNR could be defined in terms of the Root Mean Square (RMS) power of the signal (see Eq. 4.1).

$$SNR = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}}\right)^2 \tag{4.1}$$

In this case, A represents the RMS of the signal. To incorporate SNR into the pipeline while in 'silence', the system has to calculate the cumulative average RMS and compare it window-by-window with the instantaneous RMS of the current window. When the system perceives an audio window with a particularly high SNR, i.e., above a certain threshold (in our work estimated empirically), as shown in Eq. 4.2, it considers that a contact has started.

Algorithm 1 Computation of the *SNR* before the start of a touch gesture.

```
$silence\_frames \leftarrow 1$
$silence\_frames_{\max} \equiv 10$ seconds
$touch\_started \leftarrow$ False
while True do
    $window \leftarrow x_{signal}$
    $A_{window} \leftarrow RMS(window)$
    $SNR_{window} \leftarrow (A_{window}/A_{background})^2$
    if $T_a$ then
        $touch\_started \leftarrow$ True
    else if low noise then
        $A_{sum} \leftarrow A_{sum} + A_{window}$
        $A_{background} \leftarrow (A_{sum} + A_{window})/silence\_frames$
        $silence\_frames \leftarrow silence\_frames + 1$
        if $silence\_frames = silence\_frames_{max}$ then
            $silence\_frames \leftarrow 2$
            $A_{sum} \leftarrow A_{background}$
        end if
    end if
end while
```

$$T_a = \begin{cases} TRUE, & if\ SNR_c > SNR_\tau \\ FALSE, & otherwise \end{cases} \tag{4.2}$$

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 threshold, respectively. Algorithm 1 summarizes how the system computes the SNR before the start of a contact. When the event has finished, and thus the SNR returns to a value below the threshold the contact is then considered to be terminated. The resulting contact is represented by a set of windows composed of sound samples. 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 at consecutive times. Therefore, to achieve a more stable output (e.g. several tickles grouped together), the acquisition window remains open for a 500 ms extra when SNR_c drops below SNR_τ to consider the end the gesture (see Fig. 4.2).

It is necessary to point out that using only SNR in the thresholds may not be enough in all cases. As an example of this limitation, we have had to deal with the sound generated by

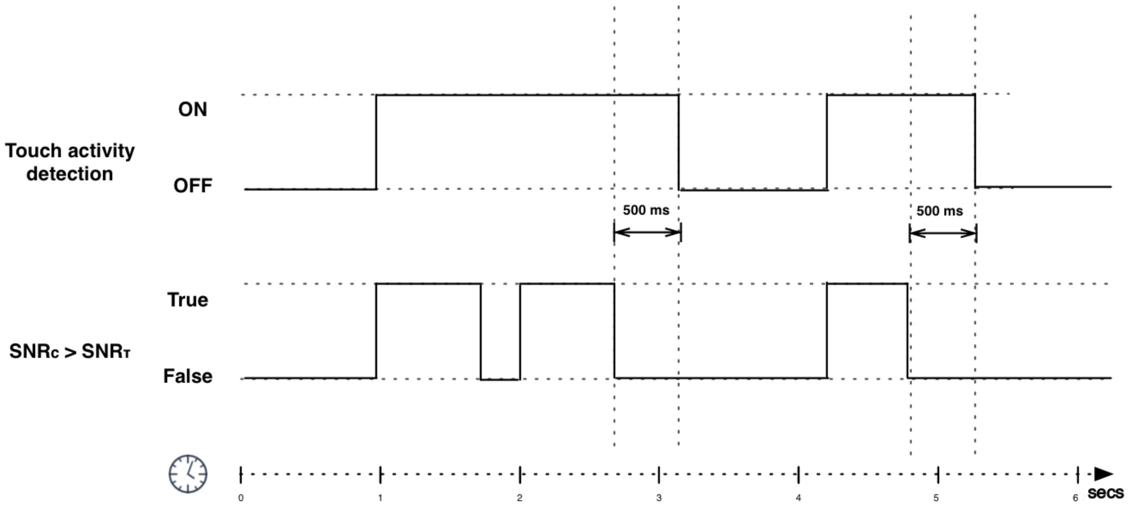


Figure 4.2: Illustration of Touch Activity Detection based on analysis of some features, such as SNR, particularly in the figure using the relation between the SNR current (SNR_c) and a SNR threshold (SNR_τ). The beginning of the gesture is detected when the SNR_c is greater than SNR_τ and the end of the gesture is detected when SNR_c is lower than SNR_τ during a time span.

the electric motors of each robot's joint. They could move at any time during the interaction, and the sound and vibrations generated by them should not significantly impact the system's accuracy. Furthermore, bumps on the surfaces with which the robot is in contact (e.g. the floor or the table on which the robot is placed) can create false positives. The decision rule mentioned above is utilised, with the addition of new circumstances, to reduce the impact of those events. To address this issue, it was determined to combine the information from the microphones with that from other conventional touch systems that may be found on the robot: capacitive sensors, which are one of the most frequent tactile sensors used in robotics. Equation 4.3 shows the updated logic.

$$T_a = \begin{cases} TRUE, & if\ SNR_c > SNR_\tau \\ & AND\ (C_1 \ldots\ OR\ C_n) \\ FALSE, & otherwise \end{cases} \tag{4.3}$$

where C_i represents the extra conditions besides the SNR, in this case, the capacitive sensors. The information these sensors provide must then be delivered to the ChucK nodes that process the soundwaves from the microphones to connect the capacitive sensor with the decision rule described above. As a result, a problem arises: transmitting the information from the capacitive sensors to the ChucK nodes. From the nodes' point of view, the first step is finding a compatible communication protocol supported by ChucK. ChucK supports the Musical Instrument Digital Interface (MIDI) and Open Sound Control (OSC) communication protocols. Because of its high compatibility, flexibility, resolution, and rich parameter space, we chose the OSC protocol to transfer the data. The next step is establishing asynchronous communication with the processes so that data acquisition is not influenced by information from the OSC channel. To address this, ChucK can execute multiple processes concurrently (as though they were running in parallel) using what the language developers called 'shreds'.

Concerning the capacitive sensors, the system must be capable of receiving information from them and transmitting it via OSC protocol to the ChucK nodes. The laboratory's robotic platforms already integrate an open-source communications middleware focused on robotic systems called ROS. In this sense, the capacitive sensors communicate their state asynchronously using ROS topics, ROS's primary communication vehicle. In summary, the idea is to translate the information from the ROS topic carrying the information from the capacitive sensors to an OSC protocol message. To tackle this task, we improve the multi-purpose control interface mentioned in Section 4.1.1. In addition to reading the YAML file and communicating with the ALSA and PulseAudio APIs, the interface incorporates the *oscpack*[19] C++ library to fulfil the 'translator' role mentioned before. Besides this, the interface will launch, pause and resume the ChucK nodes as necessary by sending OSC messages. The detailed schematic of the SA and TAD phases is shown in Figure 4.3. Since the control interface fulfils more duties concerning ChucK, we decided to include it in the TAD phase of the figure.

4.1.3. Feature Extraction

When the touch starts, the system analyses the sound signal to extract a series of features besides the SNR. This operation will be performed in the same ChucK node the TAD phase takes place. The whole set of features considered is described on Table 4.1. The table's first column indicates the name of the feature, the second one presents a brief description of it and the third one is the domain in which has been calculated. Each ChucK node starts computing and storing the instantaneous values of these sound features. Afterwards, once the touch contact is considered finished, the most relevant values of the features (average, maximum, minimum, and the difference between the maximum and the minimum, or range) are computed according to the duration of the gesture, except for the duration itself and the number of contacts per minute. The features related to the time domain are directly obtained from the sampled analogue signal acquired from the microphone. In the case of features belonging to the frequency domain, the Fast Fourier Transform (FFT) is applied to the time-domain signal [156]. Finally, features re-

[19] https://ccrma.stanford.edu/groups/osc/implementations/oscpack.html

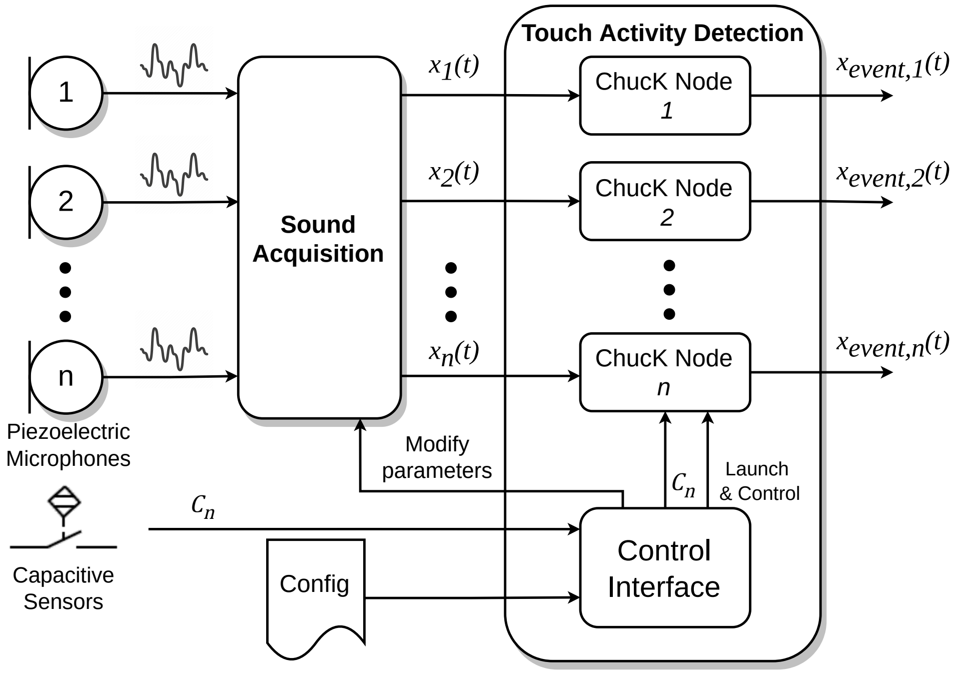


Figure 4.3: System pipeline with only the Sound Acquisition and Touch Activity Detection phases. The schematic reflects the addition of the YAML configuration file, the capacitive sensors that send the signal C_n and the control interface that sends OSC messages to the ChucK nodes.

lated to the time-frequency domain signal are obtained by applying the Discrete Wavelet Transform (DWT) [157] (see Figure 4.4).

In total, a touch gesture is composed of 33 features per microphone. Once the gesture finishes, as explained in Section 4.1.2, each of the ChucK nodes active during the current sound gesture will send their set of sound features to the control interface through OSC protocol. At this point, the DC phase starts, and all these features will be merged into a single instance representing the touch gesture.

4.1.4. Dataset Creation

Implementing and managing multiple microphones simultaneously raises some challenges. One of them is related to the detection of the touch gesture; that is, establishing its beginning and end. This issue is partially solved in the touch activity detection phase because we have this information for each microphone individually. The next challenge would be to determine what microphones were activated during the tactile gesture and for how long, from the moment the first microphone was activated until the last microphone stopped detecting sound. It needs to

Table 4.1: The set of audio features computed.

Feature	Description	Domain
Pitch	Frequency perceived by human ear.	Time, Frequency, 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 in 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 crosses the abscissa.	Time
Root Mean Square (RMS)	Amplitude of the signal volume.	Time
Signal to Noise Ratio (SNR)	Relates the touch signal to 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

be 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 4.1.1. We expected these combinations would bring diversity to the samples, allowing the classification to be more accurate.

This phase gathers the sound signal features from all of the active receivers under the same touch gesture event. For this reason, the DC phase needs to be able to coordinate and synchronise the responses from all of the microphones. When one script associated with a sound receiver establishes the beginning of a gesture, the DC node checks how many receivers have perceived the gesture within the same time period, and the system starts recording data from each micro-

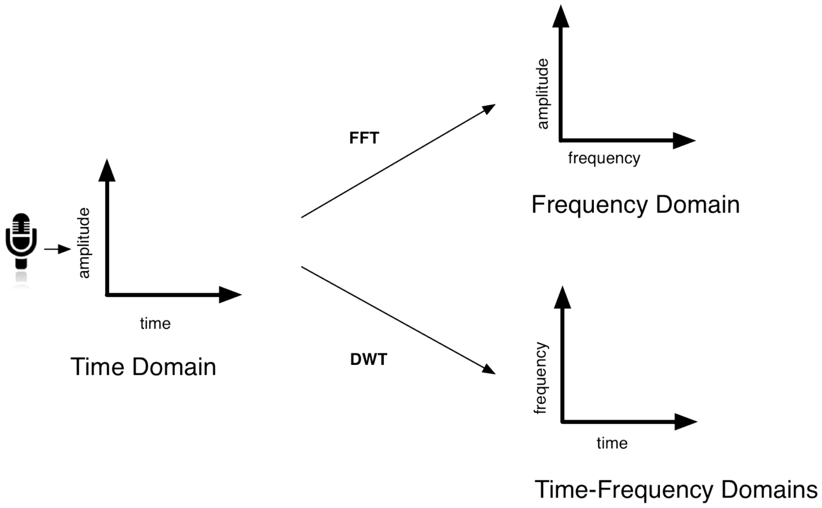


Figure 4.4: The acoustic signal is analysed in three domains: time, frequency (FFT), and time-frequency (DWT).

phone detecting the contact (a delay of milliseconds may appear due to the sound transmission). This is done by initiating a time window —a timespan of milliseconds— to record which microphones and how many of them detected the contact. The system then waits until each and every microphone involved in the interaction reports that the contact has ended. Once the current gesture ends, the DC node creates an instance with the data gathered by each ChucK node. This instance will represent the touch gesture event, and it will be composed of 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 zeros. An instance, I, will follow this pattern within the classification file: $I = (f_1, f_2, \ldots, f_n)$ where n is the number of contact microphones. Each sub-set per microphone f_i is defined by $f_i = (feature_1, feature_2, \ldots, feature_m)$ where m is the number of features computed (see Fig. 4.5.)

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

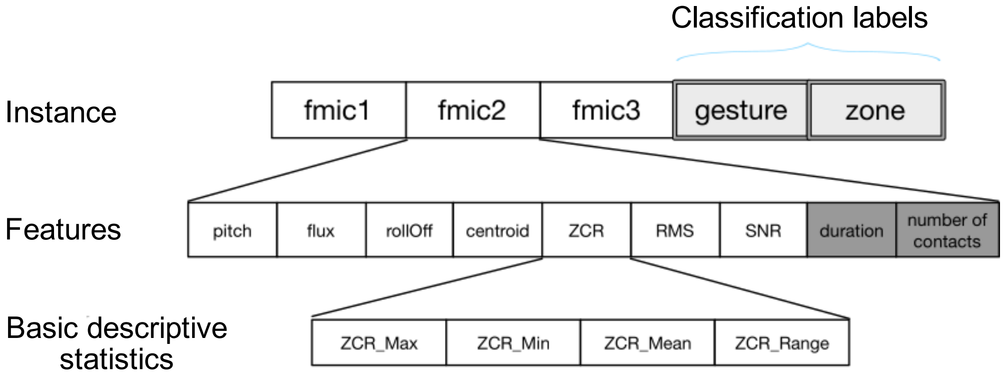


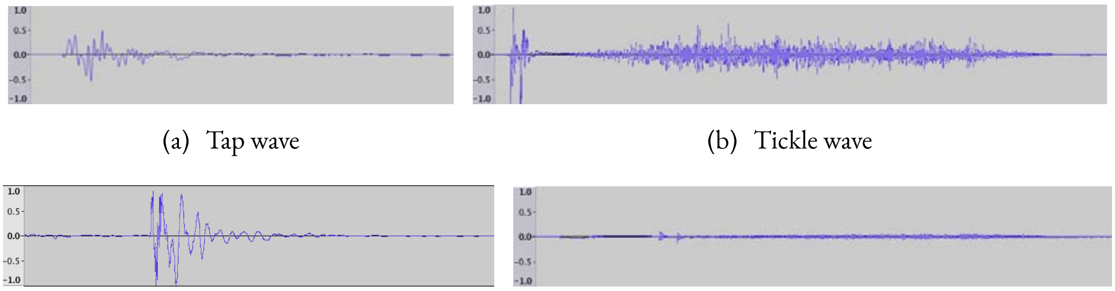
Figure 4.5: One touch gesture is stored as a dataset instance. An instance is composed of several input features corresponding to each microphone, besides the classification labels (kind of gesture and zone). Additionally, every feature is composed of four statistics: max, min, average, and range.

$$D = \{I_1, \dots, I_m\} \tag{4.4}$$

where m is the number of training instances of the dataset. The datasets are structured using a plain text format, known as Attribute-Relation File Format (ARFF), compatible with multiple machine learning frameworks. This format is particularly known for its capacity to store metadata while preserving readability.

4.1.5. Touch Gesture Classification

After the feature extraction, it is necessary to ascertain the kind of contact produced by running a touch classification process. Each kind of touch-gesture generates characteristic sound vibration patterns (*acoustic signatures*) that could be automatically differentiated using machine learning techniques. These different patterns of duration, intensity and waveform can be seen with the naked eye in a time-domain representation of the wave signatures. Figure 4.6 shows an example of the distinctive signatures for some of the touch gestures considered in this work. Using the features extracted from the sound signal, as explained in Section 4.1.3, it is necessary to determine the most appropriate algorithm for classifying those touch patterns through their main extracted features. In our case, due to the characteristics of our problem, we have decided to use *multi-class* algorithms. This is because we intend to assign the features extracted from the audio signal to a specific label or class from a group of more than two possibilities —which differentiates *multi-class* algorithms from *binary* algorithms—. More specifically, in this case, the class represents the type of contact on the robot's surface.



(a) Tap wave

(b) Tickle wave

(c) Slap wave

(d) Stroke wave

Figure 4.6: Acoustic signatures for the touch-gestures as acquired by the contact microphone in the time domain. The vertical axis represents the amplitude normalized between 0 and 1 by the highest amplitude detected among them. The horizontal axis represents the duration of the sound. All touch gestures are on the same scale, meaning they have different durations.

To proceed with the implementation of this family of machine learning algorithms, we have used the WEKA [158] framework that integrates by default 82 classifiers apart from allowing the incorporation of new ones. In this first study, we have compared all algorithms included in WEKA as well as 44 WEKA-based classifiers developed by the community (see the complete list of classifiers added to WEKA in Appendix A) making a total of 126 classification techniques. 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 networks, some deep learning implementations, bayesian algorithms, nearest neighbour algorithms, and support vector machines.

Other machine learning tools, such as *scikit-learn* [159], have been considered for use in this phase of the system. Although Python is the most widely used programming language in this field, we chose a tool that is easier to use and can be easily experimented with for this first implementation, which will be performed offline. WEKA provides a user-friendly Graphical User Interface (GUI) and tools, such as the 'Experimenter', that allow quick changes to both the algorithms and the dataset. Finally, WEKA also provides the user with flexibility by accepting multiple input formats from multiple sources when reading data. All these features make WEKA an appropriate choice for this phase. However, due to its ease of implementation and similar performance concerning WEKA, we acknowledge the possibilities of scikit-learn in online scenarios. For this reason, the system implements this library in the online phase, explained later. The work carried out in this section was reflected in the following journal publications.

Publications

Alonso-Martín, F., Castillo, J. C., Gamboa-Montero, J. J., & Salichs, M. Á. (2017). "Acoustic Sensing for Touch Recognition in a Social Robot". *In Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (HRI '17)*. Association for Computing Machinery, pp. 65–66.

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. (Q2)

Gamboa-Montero, J. J., Alonso-Martin, F., Castillo, J. C., Malfaz, M., & Salichs, M. A. (2020). "Detecting, locating and recognising human touches in social robots with contact microphones". *Engineering Applications of Artificial Intelligence*, 92, 103670. (Q1)

The first two contain a proof of concept of the system, with only one microphone, while the last publication includes the management of multiple microphones simultaneously. Both works did not include capacitive sensors at that moment.

4.2. Touch Gesture Localisation System

Within the field of robot audition, Sound Source Localisation (SSL) in robotic platforms has proven its importance [160–162] since these systems may allow a robot to improve its situational awareness and to complement other perceptual systems. Source localisation systems are currently applied in rescue environments in which visual contact is not available [163], in mapping tasks [164], or even in tasks focused on locating a human speaker in robot-assistant scenarios [165].

The purpose of the sound source localisation, in this case, is to detect where a touch has occurred on the robot's surface, i.e. the robot has a perception of not only what type of touch has occurred on its surface but also where it has occurred. Knight et al. and Chang et al. [166, 167] propose systems to localise contacts on artificial skins constructed with capacitive and force sensors, respectively. These authors emphasise the relationship between certain types of touches and their localisation. Our approach follows the same idea, but instead of focusing on acoustic sensors, we intend to apply SSL to improve HRI. In the context of robotics, touch localisation might serve multiple purposes. Some examples include the use of tactile commands to control the movement of a robot or attempting to endow a robot with the ability to understand human emotional states [168]. Knowing where the contact is made could help enhance the touch gesture detection task and the touch interaction in the scenario given in this work. As shown in Chapter 6, this might lead to the development of games and unique forms of interaction in which the user is instructed to touch a certain area of the robot in a specific way.

Existing acoustic sensing methods for locating human touch interactions on surfaces can be divided into classification systems and sound analysis systems. Classification-based systems propose the use of classification methods to distinguish the locations and shapes of various gestures. Previous works [169, 170], stated that these kinds of deployments had some flaws regarding touch localisation. Many of these limitations are related to the classification task itself, such as the requirement for a significant amount of training data and the restriction of these systems to gestures found in the training database. Despite this, such systems have several advantages, most of which are related to execution time and computational cost. Above all, they offer greater versatility on uneven or non-continuous surfaces, such as a robot's outer shell, where the surface is often made up of a set of elements that are not linked together.

In the second category, there are works that employ signal analysis techniques to pinpoint the origin of physical contacts. The taxonomy of sound source localisation algorithms revolves around the nature of sensor information used to estimate their positions. The system may therefore calculate the locations of the sources based on energy readings, Time of Arrival (TOA) measurements, TDOA measurements, and Direction of Arrival (DOA) data, or by utilizing the SRP function [171]. The signal analysis approach poses many limitations and challenges that must be considered when designing a system that implements this technology to facilitate its application in real-world scenarios. Such challenges include coping with computationally expensive operations that might imply delays in real-time operations. Moreover, if the nodes assigned to each sound receiver operate individually, the resulting audio signals might not be synchronized.

In this section, we propose an implementation for each system type, classification-based and analysis-based, because we believe the system can benefit from the advantages of each of these approaches depending on the context or robotic platform in which they are used. One criterion, for example, could be the size of the social robot. As illustrated in Chapter 2, a social robot can adopt different shapes. Thus, on a larger social platform, a more accurate approach (analysis-based systems) may be required, whereas classification-based techniques may be more appropriate on a smaller platform. Another criterion is computational cost because the social platforms shown in the literature have a wide range of computational capacities. Because we intend to implement our system on various platforms regardless of their shape or computational capacity, it makes sense to investigate both approaches.

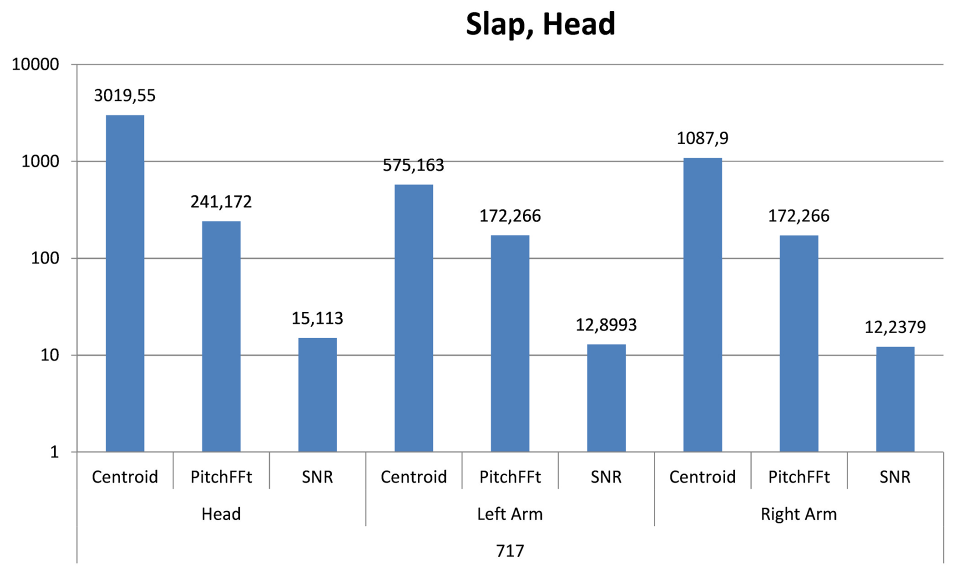
4.2.1. First approach: Machine Learning Classification Techniques

When a 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 [172], it would be expected then that those receivers located closer to the point of touch contact acquire a stronger signal than those that are located farther away. But in practice, some social robotic platforms have outer shells composed of a set of elements that are not linked together (i.e. Pepper [173], Nao [174] or Mini), and this expected behaviour does not always appear. In fact, there are cases in which strong touches are registered by different sensors with similar intensities, regardless of the distance. 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.

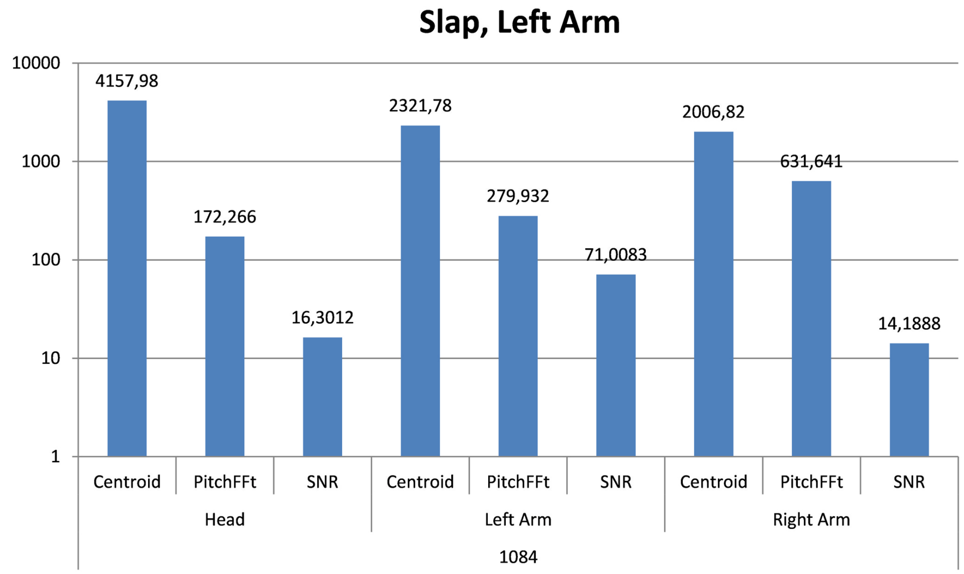
The sound propagation phenomenon is illustrated in Figure 4.7. The figure shows a graphical representation of some relevant features extracted from two instances in the dataset created in Section 5.2, although this situation repeats throughout the data collected. Each chart shows how the detection of the relevant features may not correspond to the values expected when slapping the robot in different places. That is when slapping the robot on the head (see Fig. 4.7-a)), the microphone in the head detects the highest values for all of the features. We expect the following: when a user touches a certain part of the robot with a microphone, that sensor is expected to provide higher feature values for some of the features. In contrast, as shown in Figure 4.7-b), we can see how a slap on the left arm does not provide the highest centroid (the median of the signal spectrum in the frequency domain, as explained before) 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. Therefore, this section studies how machine learning can help to mitigate this problem.

We decided to approach the issue caused by the sound propagation in a robot shell presented before as a multi-dimensional machine learning problem. We will assign one class label to the type of gesture, as described in Section 4.1.5, and the other label will be the contact location, more specifically, the area where the contact was detected.

Multi-dimensional classification problems appear in many application domains. For example, a text document or semantic scene may be assigned to multiple topics; a gene may have various biological functions; a patient may have numerous diseases or develop drug resistance to multiple HIV treatments; a physical device may malfunction due to various components failing; and so on. As Bielza et al. point out, multi-dimensional classification is a more difficult task



(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

Figure 4.7: Different instances represented in bar charts in logarithmic scale. The most significant features were selected to show the variations depending on the contact location. The *pitchFFT* and *centroid* features are measured in *Hertz*, and the *SNR* is dimensionless.

than single-class classification. The main issue is the significant number of possible class label combinations and the scarcity of currently available data [175]. The multi-dimensional classification problem can be divided into two categories: *multi-label* and *multi-target* problems.

1. Multi-label classification: In this approach, proposed by Schapire and Singer [176, 177], a data instance is composed by various binary class labels that can be classified simultaneously by one machine learning algorithm. This contrasts with the traditional task of single-label (binary or multi-class) classification, where each instance only contains one class label. Currently, this approach is being applied to various domains, including music, text processing, video, image, and even bioinformatics.

2. Multi-target classification: In multi-target learning (also known as multi-objective or multi-class multi-output learning), a single classifier is able to simultaneously recognise multiple labels that, in contrast to the multi-label approach, can take multiple values [175, 178]. 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. Since our use case includes two multi-class problems, one for the gesture and one for the location (more than two different gestures and more than two different zones), this is the approach selected in this system stage.

4.2.1.1. Design and Implementation

To test multi-dimensional classification algorithms, we have used the third-party framework, known as Multi-dimensional Environment for Knowledge Analysis (MEKA) [179], an extension of WEKA specially designed for multi-label and multi-target scenarios. MEKA integrates all of the basic problem transformation methods, advanced methods and multiple examples of classifier chains. The context of multiple target variables has important implications for classification (how to model dependencies between variables) and evaluation (how to score multiple target classifications for each instance) that traditional single-label frameworks do not deal with. MEKA has been designed specifically for this context.

MEKA was created to perform and evaluate multi-dimensional classification using the popular and effective family of problem transformation methods, which make use of existing off-the-shelf single-label (binary or multi-class) methods as 'base classifiers'. By default, MEKA integrates several kinds of meta-classifiers[20], such as Binary Relevance (BR), Classifier Chains (CC), Classifier Trellis (CT), Label Combination (LC), Ensembles of Pruned Sets (EPS), Ensembles of Classifier Chains (ECC), Nearest Set Replacement (NSR) and Bayesian Chain Classifier (BCC). In the same way as WEKA, MEKA can be extended with third-party algorithms. We have reflected this framework's inclusion in the system's partial pipeline in Figure 4.8. As the reader might perceive, we decided not to drop the multi-class approach, and in this case,

[20] An updated list of classifiers available in MEKA can be found here: http://meka.sourceforge.net/methods.html.

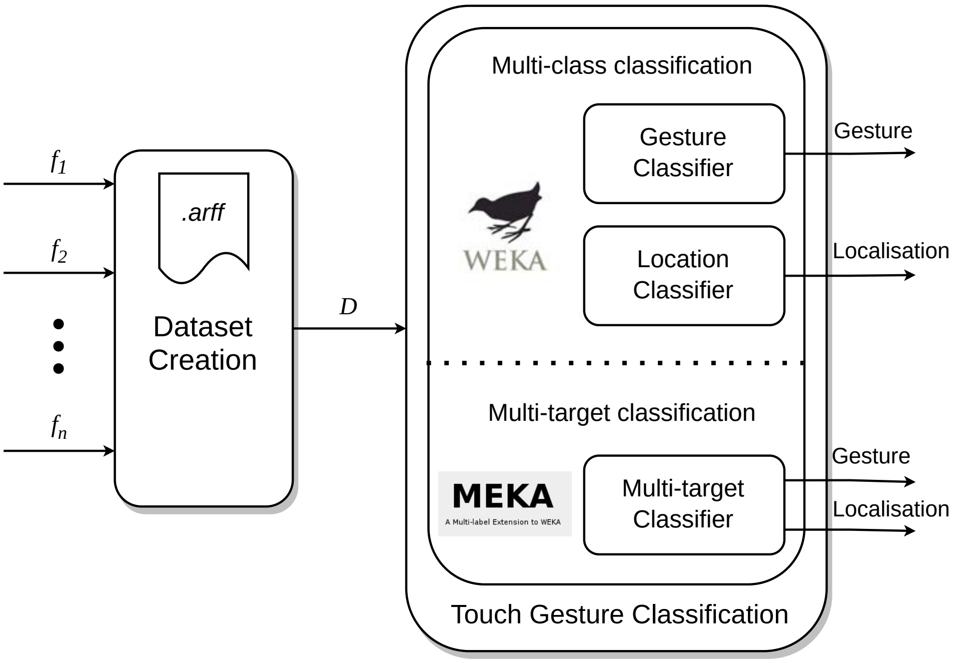


Figure 4.8: Data flow scheme of the DC and TCL phases of the ATR system, where the classifiers considered take as input the instances in the dataset and outcome the kind of touch and its approximate location.

the problem has been divided into two non-related multi-class problems. Both multi-class and multi-target approaches will be tested in section 5.2.

Other frameworks for multidimensional classification were considered besides MEKA, as in Section 4.1.5. *MULAN* [180] and *scikit-multilearn* [181] were some of the most relevant frameworks discovered in the matter. Again, the main advantage of the framework selected at this stage of the system is that it provides a very comprehensible GUI. More specifically, MEKA outperforms MULAN in terms of performance [179]. And with respect to scikit-multilearn, MEKA does support multi-target learning while scikit-multilearn only provides a wrapper to MEKA to offer this functionality. The tools described in this subsection and their impact on the ATR system were included in the following journal publication.

Publication

Gamboa-Montero, J. J., Alonso-Martin, F., Castillo, J. C., Malfaz, M., & Salichs, M. A. (2020). “Detecting, locating and recognising human touches in social robots with contact microphones”. *Engineering Applications of Artificial Intelligence*, 92, 103670. (Q1)

4.2.2. Second approach: Sound Analysis Techniques

Signal analysis, as opposed to other methods that rely on machine learning algorithms, allows for a higher resolution of the contact point. As a result, we envision that the system can be implemented on larger robotic platforms, with larger surfaces that offer more possibilities in terms of interaction, where higher resolution in touch localisation could imply integrating novel applications in social robotics. An example could be using tactile commands to control the robot's movement.

When modelling an SSL problem, the first element to define is the propagation model that the system follows. Depending on how far away an observer is from a sound-emitting object, the acoustic energy produced by the sound source will behave quite differently. A sound propagation model defines this behaviour and it is determined by: (i) the positioning of the microphones to properly acquire sound sources between them; (ii) the robotic application, as the sound generated by the user's touch may be close or far away from the microphone array; and (iii) the environment characteristics, as they define how sound propagates in the medium. When all these elements are specified, the propagation model defines the type of features to be used, and the information that can be obtained from the sound source, and limits the number of methods available to obtain this information.

In the *far field* propagation model, the source is far enough to as a point in the distance, with no discernable dimension or size. At this distance, the spherical shape of the sound waves can be approximated to a plane-wave, with no curvature. When you get close to a sound-emitting object, the sound waves behave much more complexly. The *near field* refers to this complex region. Because of the mix of circulating and propagating waves, there is no fixed relationship between distance and sound pressure in the near field, and measuring with a single microphone can be difficult and unreliable. Two elements of a sound source position can be computed as part of a SSL task, depending on the propagation model [182]: If the source is in the far field, there is a *direction-of-arrival estimation* [183], and if it is in the near field, there is a *distance estimation* [43, 45, 184].

The next element to define when modelling the SSL analysis problem is the source localisation method. The most common approaches for source localisation focus on different types of acoustic features, namely, the energy of the incoming signals, their TOA or TDOA, DOA, and SRP, resulting from combining multiple microphone signals [171]. Next, a short description of the features is provided:

- *Energy-based localisation*. This feature relies on the averaged energy readings computed over windows of signal samples acquired by the microphones [185]. Compared to TDOA and DOA methods, energy-based approaches are interesting because they do require the use of fewer microphones and they are free of synchronisation issues unlike those methods based on the TOA. These methods are mostly used for wireless acoustic sensory networks (WASNs) because of the low variation of acoustic power.

- *Time-Difference-Of-Arrival*. The TDOA is related to the difference in Time of Flight (TOF) of the wavefront produced by the source in a pair of microphones at the same node. TDOAs can be estimated at a moderate computational cost through the Generalised Cross Correlation (GCC) [186] of the signals acquired by microphones in the pair. The TDOA measurement constrains the source to a branch of a hyperbola with vertices in microphone positions and an aperture determined by the TDOA value. When two (or three in 3D) measurements from different pairs are available, the source can be located by intersecting hyperbolas. However, the resulting cost function is strongly non-linear, making minimization complex and error-prone.

- *Time-Of-Arrival*. TOA measurements are obtained by detecting the time instant at which the source signal arrives at the microphones present in the network. TOA uses the method of trilateration by forming equations for the anchors representing the circle having a radius equal to the distance from the source [187]. The solution to these equations gives the intersection point which is the location of the source. TDOA localisation methods are usually chosen instead of TOA methods since the latter requires precise timing hardware and synchronization mechanisms.

- *Direction-Of-Arrival*. The objective is obtaining the direction from which a propagating wave arrives at a point, where usually a set of sensors are located [188]. This approach requires a set of arranged sensor arrays. For this reason, this method is usually found in far-field environments. In these approaches, which involve multiple sensor arrays, each set of sensors estimates the DOA of the sources and transmits its estimate to a fusion centre. This method's main drawback is that it requires high computational power and multiple microphones at each node. However, they can reach very low bandwidth usage since only the estimates need to be sent. Also, since the DOA estimation is carried out in each node individually, the audio signals at different nodes do not need synchronisation.

- *Steered Response Power*. As a step forward from TDOA-based approaches, SRP approaches are beamforming-based techniques that compute the output power of a filter-and-sum beamformer steered to a set of candidate source locations defined by a predefined spatial grid [189]. This technique is considered as an extension of TDOA-based approaches because the computation of SRP requires the accumulation of GCC functions from several pairs of microphones. The SRP power map is made up of the collection of SRPs that were collected at various points in the grid, with the estimated location of the source represented by the point that accumulated the highest value.

Due to its robustness in noisy and reverberant environments, SRP-based approaches have attracted the interest of numerous researchers. In the particular case presented for these works, since social robots are usually composed of curved surfaces, the idea of modelling the surfaces as a grid seems a reasonable approach to make the system know the properties of the environment in which the contact is being performed beforehand. For this reason, SRP is the sound source localisation method of choice for our system.

The SRP algorithm relies on the following principles [190]. First of all, the signal received at the n^{th} sensor of a microphone array can be modeled as

$$x_n(t) = a_s(t) * h_n\left(\theta_s, t\right) + v_n(t) \tag{4.5}$$

where $a_s(t)$ is the signal generated by the source, θ_s is the position of the source, $h_n\left(\theta_s, t\right)$ is the impulse response from θ_s to the n^{th} sensor, and $v_n(t)$ is the sensor noise, which is generally supposed to be white, Gaussian and not correlated with the source signal and the noise of other sensors. It is worth mentioning that θ_s is written in bold because it can represent an angle, two spherical coordinates, or even a point in 3D Cartesian coordinates depending on the geometry of the array.

One of the most classic and popular approaches to DOA estimation is finding the direction that maximises the SRP that we would obtain using a filter-and-sum beamformer:

$$\hat{\theta}_s = \text{argmax}_\theta\, P(\theta) \tag{4.6}$$

$$P(\theta) = \int_{-\infty}^{+\infty} \left| \sum_{n=0}^{N-1} G_n(\omega) X_n(\omega) e^{-j\omega\tau_n(\theta)} \right|^2 d\omega \tag{4.7}$$

where N is the number of sensors of the array, $X_n(\omega)$ is the Fourier Transform of $x_n(t)$, $G_n(\omega)$ is the frequency response of the filter for the channel n, and $\tau_n(\theta)$ is the time delay occurring from the position or direction θ to the n^{th} sensor.

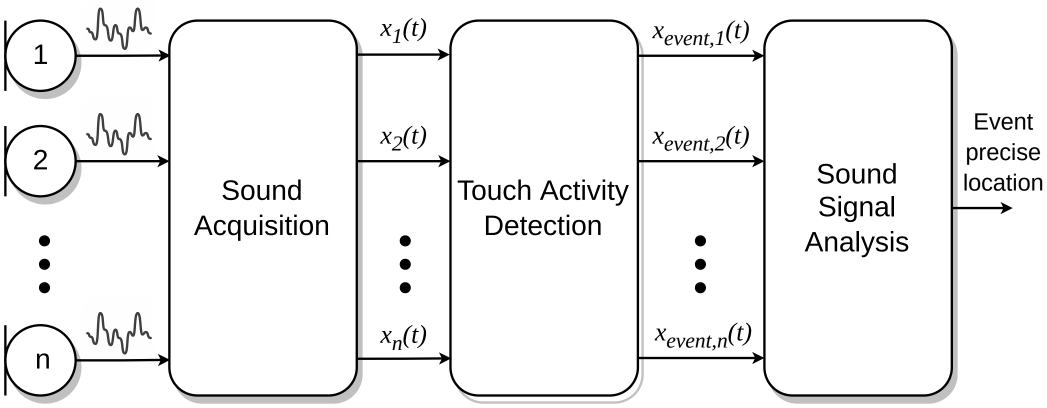


Figure 4.9: Pipeline of the ATR with the Sound Signal Analysis module.

4.2.2.1. Design and Implementation

At this point, the system is able to identify whether a contact has occurred on the robot and if so, it has extracted and stored the contact in the form of sound windows, this is represented by the SA and TAD phases in the Figure 4.1 from the Section 4.1. The objective here is to analyse the signal obtained and to find the location of the contact. As Figure 4.9 depicts, this phase of the system will replace the FE phase and beyond.

Figure 4.10 shows a detailed pipeline of the Signal Analysis system phase. The objective consists of the computation of the candidate sound source location or 'event location', as the Figure shows at the end of the pipeline. This is defined by a spatial grid (shown in Figure 4.11) that represents the environment in which the touch interaction has taken place. After setting up the grid, as Figure 4.10a shows, the system computes the Short-time Fourier Transform (STFT) [191] for the entire set of samples composing the event (Fig. 4.10b). This method of calculating the Fourier transform divides the signal into shorter fragments and obtains the Fourier spectrum information in each of them. The size of these fragments or windows can be adjusted and is considered a parameter of the system. One important feature of computing the STFT is that frequency bins conveniently separate the returned information. To take advantage of this feature, a frequency filter (Fig. 4.10c) has been implemented, which narrows down a suitable frequency range where the sound source might be present. The user can adjust this frequency range, and for this reason, is also considered a parameter of the system.

After filtering the signal, we implement a variation of the SRP algorithm that includes the Phase Transform (PHAT) weighting function [192, 193] (Fig. 4.10d) to whiten the cross-spectrum of the sound signal, giving an equal contribution to all frequencies, to rely only on phase information of the signal leading to much sharper correlation peaks. The main drawback

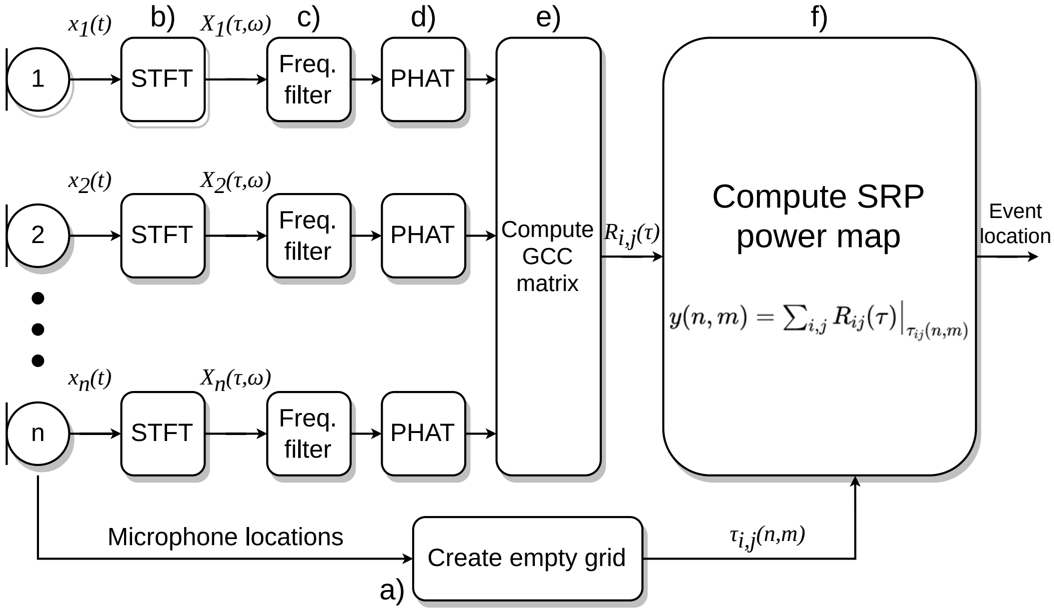


Figure 4.10: Pipeline of the Sound Signal Analysis module: (a) grid creation, (b) STFT computation, (c) frequency filter, (d) PHAT weighting, (e) computation of the GCC matrix, and (f) the resulting power map with the candidate location

of this method is that frequencies dominated by noise are also considered equally. To try to overcome this problem, and based on the work of Basiri et al. [194], the system also implements an exponential coefficient α for PHAT, which, ranging from 0 to 1, will control the 'amount of PHAT weighting' applied during the SRP algorithm (0 meaning *no-PHAT* and 1 *full-PHAT*).

The next step is to calculate the set of SRPs obtained at the different points of the grid by using the spectral information provided by the STFT. This information helps create a "Cross-Correlation Matrix", similar to the one used in the GCC algorithm [186], which provides the TDOA information extracted from the microphone pairs (Fig. 4.10e). Lastly, the system combines the GCC matrix with the spatial grid that represents the environment (represented in Fig. 4.10a) and assigns a value to each point of the grid depending on its position concerning the microphones, thus creating the *SRP power map*. In summary, this map represents how likely each point in the grid could be the sound source, where the point accumulating the highest value corresponds to the estimated source location (Fig. 4.10f). Fig. 4.11 shows an instance of the power map after contact on the surface. The work presented in this section was carried out as an international collaboration with the Instituto Superior Técnico (IST) in Lisbon and resulted in the following conference publication.

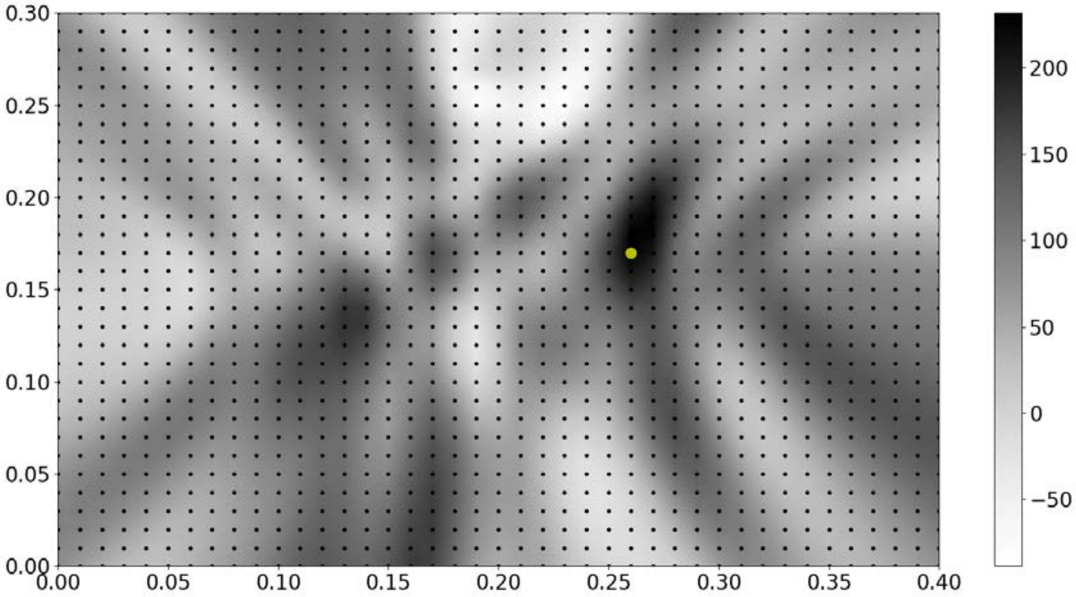


Figure 4.11: The SRP power map. Higher power values correspond to a darker colour. The yellow dot indicates the peak value.

Publication

Gamboa-Montero, J. J., Basiri, M., Marques-Villarroya, S., Castillo, J. C., & Salichs, M. Á. (2022), "Real-Time Acoustic Touch Localization in Human-Robot Interaction based on Steered Response Power", *In Proceedings of the 2022 IEEE International Conference on Development and Learning (ICDL).*

4.3. Online System Integration

In Chapter 2, we presented some of the limitations the STS systems in the literature had. Two of the most relevant, in our opinion, were having a system that (i) works online and (ii) did not require external sensors or an environment specially adapted for the task. In other words, the main objective has been to have an online, portable STS system. With that idea in mind, up to this point, we designed a system able to detect, recognise and localise touch gestures by creating a dataset with the information. After that, as shown in Sections 4.1 and 4.2, we use a framework to train a series of machine learning techniques offline. At this point, the system can only create datasets containing information about the type of contact and its location. Therefore, the next step is integrating an online version of the system to use this data to predict the contact type and location. This stage aims at using the information and machine learning data extracted

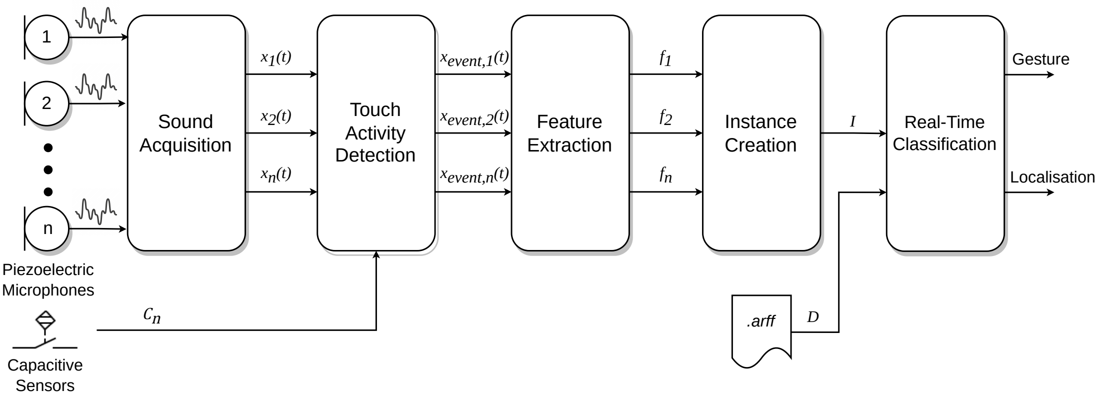


Figure 4.12: The pipeline of the online ATR system. The main difference relies on the last two phases: the DC phase now turns into the IC phase, which sends unlabelled instances to the OC phase, designed to predict the type and location online of the touch gesture.

from the system, described in Sections 4.1 and 4.2.1 to create a classifier that the robot can use to determine what types of gestures are performed on various body parts.

The online version of the system requires modifications on the operation pipeline described in Section 4.1. The first update involves the DC phase of the system. As described in Section 4.1.4, in this phase, the system gathers the information from multiple microphones f_i that belong to the same gesture. The pipeline explained before involves a scenario where we know beforehand how and where the user performed the contact, and we can call this scenario the 'train' scenario. The system labels and stores the touch gesture with the information known, creating a data instance I_i that will be stored in a dataset D. In this section, we present the 'predict' scenario, where the system does not have this information, so it would not be possible to label the instances to create a dataset. But, despite this, we would still need the unlabelled instance since it would be the input for the next phase of the system. In consequence, the DC phase turns into an Instance Creation (IC) phase that provides unlabelled instances to the next phase, the Online Classification (OC). These changes in the pipeline are reflected in Figure 4.12. This schematic reflects how the dataset is still present, not as a part of the IC phase but as an input for the OC phase, which, as it will be explained in detail later, will use the labelled instances to train the machine learning models.

The current section reports these changes in the pipeline, focusing on the last stage, the OC phase. We will describe this phase's requirements and the libraries implemented in the resulting classification module. This section also explains how the module functions and how it is completely integrated into the platform to make the system portable. We have to clarify that at this point, the system in its current iteration does not have the localisation system described in

Section 4.2.2 fully integrated, and the online setting is implemented with the system described in Section 4.2.1, being focused on approaching the localisation problem as a machine learning one.

4.3.1. Design and Implementation

This subsection describes the design of the OC module. In order to integrate the element in the architecture of the ATR system and the architecture of the robots present in the laboratory, it is designed as a ROS node. The first element to define in the OC phase is the set of machine learning tools that will compose its core. As explained in Section 4.2.1, the library that best adapted to our needs in the offline setting was MEKA due to its GUI and the amount of multi-target algorithms at its disposal. Despite this, moving to the online classification domain presented challenges that made us rethink our decisions concerning the machine learning framework chosen for the classification task.

On the one hand, as mentioned in Section 4.1.5, WEKA and MEKA are machine learning frameworks designed to offer a robust testbed for machine learning problems. Still, its application in real-time systems is not that frequent. In our case, to integrate with the architecture of the robots in our laboratory, the system needs to be compatible with ROS. That implies constraints in terms of libraries and programming languages our node can implement. More specifically, ROS uses Python and C++ as their primary programming languages. On their side, WEKA and MEKA have their respective APIs to design applications, but they have to be programmed in JAVA, which, at first glance, implies an incompatibility issue.

On the other hand, scikit-learn is an online classification solution to most machine learning applications since it is easy to implement, well-documented, and its API is developed in Python, the most widespread programming language across the ROS community. However, the library has a set of limitations mostly related to its application to multi-target settings. In this situation, both approaches, MEKA and scikit-learn, have their advantages and shortcomings, and both would require designing a way to adapt them to our machine-learning problem. Since the pipeline for our machine learning node would be essentially the same regardless of the framework we implement, we decided to integrate both and compare their performance in our online setting:

1. **MEKA approach:** ROS presents solutions for JAVA integration. *ROSJAVA* is a set of ROS libraries to enable the integration of JAVA and Android with ROS-compatible robots. Despite its active development, ROSJAVA's downsides include a lack of thorough documentation, instability (the bundle is still in alpha) and lack of compatibility with

some of the ROS's most essential communication libraries, like *actionlib*. Another option to enable compatibility between ROS and MEKA involves the design of a wrapper. Python is a language that provides many tools to allow the design of interfaces for libraries written in different programming languages. One of the most powerful solutions for JAVA is the *JPype*[21] library. JPype offers complete JAVA access from Python. It enables Python to explore and visualise Java structures, create and test Java libraries, and use Java-specific libraries. JPype also provides a potent engineering and code development environment using Python for rapid prototyping and Java for strongly typed production code. By integrating JPype in a Python ROS node, it can access all the tools provided by the MEKA library as if the node was written in JAVA. We opted for choosing Jpype in order to integrate MEKA into our online classification module.

2. **Scikit-learn approach:** By default, the scikit-learn set of tools destined for multi-target classifiers is very limited, having only a couple of estimators implemented for these tasks, like a multi-class multi-output classifier or the their Multilayer Perceptron (MLP) estimator[22]. Furthermore, native libraries based on scikit-learn also present a series of limitations. For example, scikit-multilearn relies on a rudimentary MEKA wrapper to perform multi-target classification. Therefore, we decided to manually implement a CC to classify both touch gesture and localisation labels. Deploying this ensemble algorithm requires two multi-class classifiers: one predicts the gesture, and the other predicts the body part touched. A chain of classifiers propagates the label prediction from one classifier to another by first classifying one label and then using the classified label as an extra attribute to classify the next label. According to Read et al. [195], the accuracy classifying the label can be chosen as a criterion to establish the chain order. For this reason, we choose the location as the first label in the chain and the touch gesture as the second. As the next chapter shows, the location is a label that is classified more accurately than the touch gesture.

After defining the set of tools that will be included in the OC node, we describe the different processes involved in this stage of the system. The complete flowchart is detailed in Figure 4.13. The operation is divided into two consecutive phases: a *build stage*, where the node prepares the data and trains the machine learning models, and the *classify stage*, where the system predicts the requested information and publishes it as a ROS topic.

[21] https://jpype.readthedocs.io/en/latest/

[22] More information about this topic in https://scikit-learn.org/stable/modules/multiclass.html

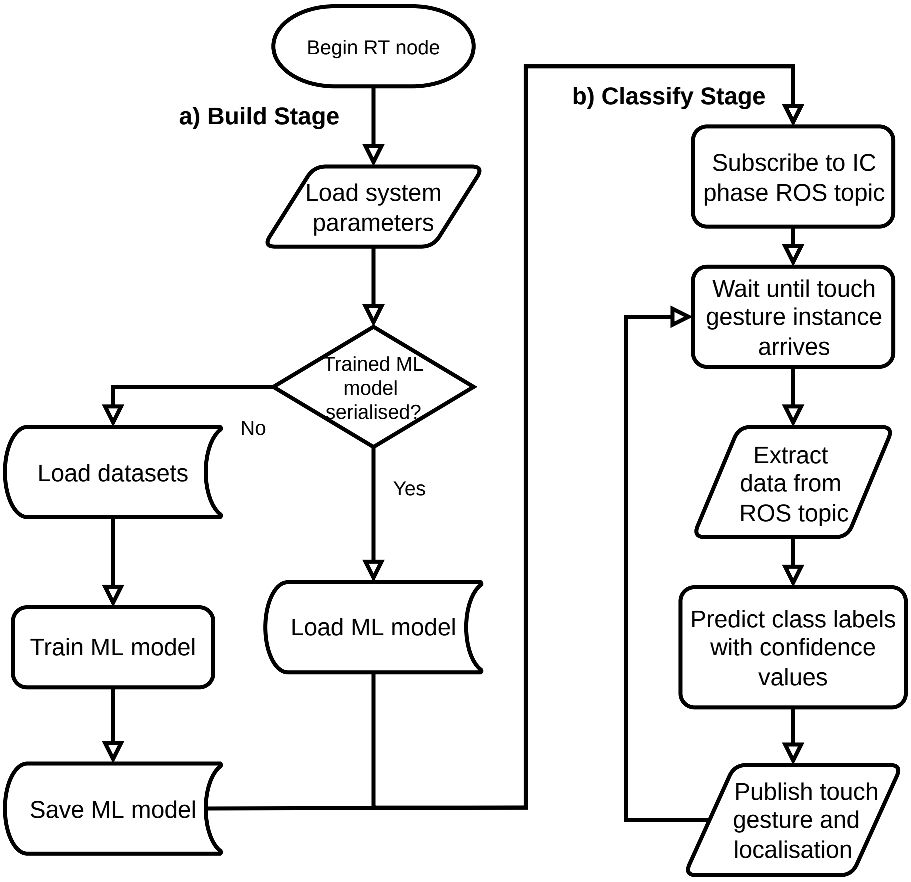

Begin RT node
a) Build Stage
Load system parameters
Trained ML model serialised?
No
Yes
Load datasets
Train ML model
Save ML model
Load ML model
b) Classify Stage
Subscribe to IC phase ROS topic
Wait until touch gesture instance arrives
Extract data from ROS topic
Predict class labels with confidence values
Publish touch gesture and localisation

Figure 4.13: Flowchart representing the OC module. a) The system loads the data and builds the machine learning models, and afterwards, b) it waits for a new instance to arrive in order to predict the touch gesture performed and its location.

4.3.1.1. Build stage

The scikit-learn and MEKA libraries allow preprocessing and applying transformations to datasets. Both libraries and Python tools to handle datasets create an optimal environment for designing machine learning applications. In this project, we implemented a supervised learning algorithm to classify gestures and the parts of the body on which they are performed. The steps the node has to perform to obtain the trained machine-learning models (shown in Figure 4.13a) are as follows:

1. First, the system loads data from the robotic platform, such as its name and installed microphones, to properly find and load the corresponding dataset and label the resulting trained model.

2. At this point, the system evaluated whether there is already a model for the datasets loaded. If not, it proceeds to train the model using the data available. First we explain the steps carried out by the OC in case the latter happens.

3. As mentioned earlier, before training models with the dataset, the samples were stored in an *arff* format file. To load the data in the OC node, we use a data structure called *dataframe* contained in the *Pandas* library, a Python module designed for data analysis and manipulation. A *dataframe* is a two-dimensional data structure, similar to a table or spreadsheet, that allows data to be stored in identified rows and columns and accessed for the program online, optimising resources.

4. Then, the system trains a machine learning estimator in order to classify online the incoming instances from the IC ATR system phase. We selected the estimator better suited to our needs in the offline tests shown later, in Section 5.4. This is the step where the first difference between the machine learning tool used appears since the machine learning model trained here would be either from MEKA or scikit-learn.

5. Once the model is ready, it is saved in a file using the *pickle* library, which allows serialising objects and data in files and retrieving the serialised data from the generated file. Introducing serialisation avoids retraining the system when the datasets are the same, significantly reducing the system's startup time.

6. In case there was already a trained ML model from previous training, by using The *pickle* library, the system loads the model and prepares goes to the *classify stage*.

At this point, the node is ready to move forward to the classification stage.

4.3.1.2. Classify stage

Once the system has the models built, saved, and ready to predict the gestures, the node proceeds to the next phase. Online sample classification is possible thanks to the implementation in ROS. The middleware makes the ATR capable of performing the following: collecting the information from the touch sensors when detecting a contact on the robot (SA and TAD stages), pre-processing the features from the interaction in a format that the machine learning model can use (FE and IC stages), and finally, to perform the classification for each group of classes (*touch gesture* and *contact location*). This last step is what is covered in this OC module. More specifically, between the last two steps, the classification results are then sent from the IC node through a ROS *topic*, which sends the predicted gesture and body part labels and the confidence values for each label.

The actions carried out by the OC module at this point correspond to the *classify stage* and can be broken down into the following steps, beginning with gathering data from the touch gesture and ending with the prediction of the type of contact and its location. The steps described here are shown in Figure 4.13b.

1. The node subscribes to the *topic* that sends the unlabelled instance. The raw values of the attributes extracted from the touch interaction are received through this communication channel.

2. Afterwards, the node waits for the incoming data from the IC stage.

3. After a touch gesture has finished, and the resulting instance from the IC phase representing the gesture is sent, the OC module receives the message and extracts the attributes of the touch gesture from the ROS topic. Then, the node transforms the list of attributes into a *dataframe*, which contains the names of the different attributes and their respective values. As already mentioned, a *dataframe* allows handling incoming data and is one of the primary input sources for the machine learning models in this prediction stage.

4. At this point, another difference between the MEKA and scikit-learn approaches appears. With MEKA, once the data is ready to predict, the information is fed directly to the model, which returns the predicted labels with their corresponding confidence values. In the case of the scikit-classifier approach, since the implementation of the CC is done manually, first, the above *dataframe*, which contains the attributes of the detected touch instance, is fed into the classifier model of the contact location. As a result, the prediction of the contact location and the confidence of the predicted class are obtained. We use the robot's predicted body part as another attribute to predict the gesture. For that, we add the coded prediction of the contact location into the *dataframe* above. As with the body part, the classifier model of the gesture performs the prediction on the *dataframe*, which now includes the predicted robot part. This way, we obtain the gesture prediction with its confidence value.

5. Finally, the node publishes a ROS message in a topic containing the predictions of the gesture and the contact location and their respective confidence values. Once this operation is finished, the node checks again if there are new touch instances to classify.

In Section 5.4, we test this module on the robot using a set of *touch gesture-contact location* predictions. These were analysed, obtaining confusion matrices and machine learning classification metrics for the labels. In addition, we also explore the confidence values obtained for the correct predictions to check the degree of confidence of the classifier in predicting the different labels. We conduct these analyses for both online classifiers, the one implemented with MEKA and the one using scikit-learn. Then, we compare the results from the two systems, mentioning the advantages and disadvantages of each of them. This version of the system is afterwards put to test in a research environment for the experiments conducted in Chapter 6.

4.3.2. System Integration

At this point, the system is completely developed, so what remains to be done? One of the challenges we discussed in Chapter 2 regarding STS systems was the need for full integration of some of those systems and how this poses problems to be solved. Among the issues we could highlight would be those concerning the software designed for these systems and, more specifically, the goal of developing modular and portable STS systems at the hardware and software levels. These goals imply that the system should be easy to deploy on the robotic platform, it also should be self-contained, and despite being isolated from the rest of the modules that compose the robotic platform, it should be able to communicate with the platform's subsystems. Likewise, the system has to cope with eventualities that may arise, such as device disconnections or unexpected restarts.

This section covers the modifications we have integrated into the system to meet the above challenges. First, we specify how our system works with the software architecture explained in Section 3.3. Secondly, we describe our approach regarding portability, introducing the concept of containerisation into our system and the tools that allow this. Afterwards, we explain how we can improve the system's robustness at a software level by handling possible receivers' disconnections and implementing a controlled ATR system restart.

As we mentioned in Section 3.3, our architecture consists of three 'managers', Perception manager, HRI manager and Expression manager; a Liveliness module, a memory that stores data about the robot and the user (the Context module) and a Decision-Making System (DMS) with skills the robot can perform. The perception and actuation modules contain detectors connected to the sensors and drivers connected to the actuators.

In our case, The Perception manager bridged the information provided and the rest of the architecture to integrate the system. The Perception manager's primary function is to receive data from each robot's sensor and produce a unified message that the other system modules can understand; more specifically, it filters, formats and packages the information according to different criteria (type of data, connections, time window, etc.). The Perception manager has three abstraction levels. More specifically, the ATR will be connected to the Perception manager's Level 0, which comprises modules that take information from specific sensors or percep-

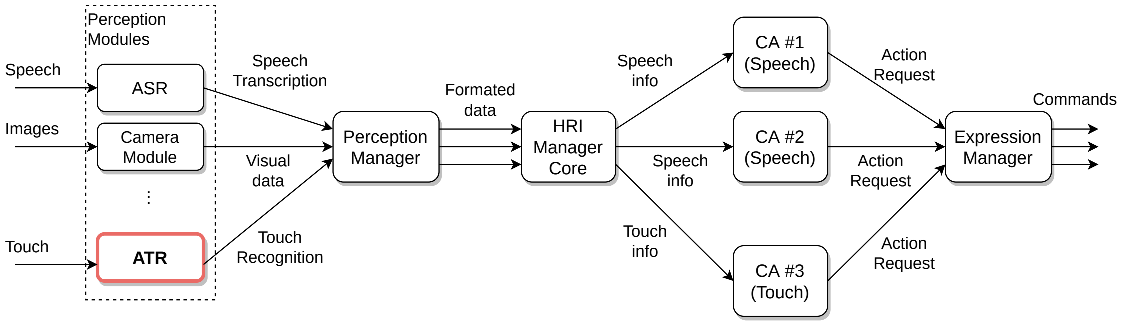


Figure 4.14: Schematic of the ATR integration in the system's architecture. Our system (highlighted in red) is a Perception module that sends touch-related information to the Perception manager.

tion units and convert it into a standardized format. For that, we modified the ROS message to be an array of key-value pairs, where the keys are tags that identify the information being sent, in this case, the type of touch gesture, contact location and their respective confidence values. Figure 4.14 shows where our system fits in the architecture regarding the three main 'managers'.

4.3.2.1. Improving the system's portability

System portability is a crucial criterion for improving STS systems, as its software is often developed with specific criteria for each task. Improving the system's portability facilitates switching between different devices, a highly desirable feature from a design point of view, and simplifies the system's deployment on a platform. To achieve this, we have implemented in the ATR system a virtualisation technique at the operating system level known as containerisation, which isolates an application or group of applications together with their libraries and configuration files. By doing so, they do not modify the operating system in which they are installed. Another advantage of containerisation is control. Most containerisation software includes tools to evaluate and monitor a container's performance and state, both within and outside the container's platform, increasing the system's robustness.

The main alternative to containerisation is a virtual machine. However, resource usage is the main advantage of containerisation compared with a virtual machine, especially in cases of standalone systems such as the ATR. Containerisation packages applications as portable container images to run in any environment consistently. Among the current software that allows creating and managing containers, we have implemented *Docker* [196] since its one of the most widely used containerisation tools. It also offers state-of-the-art performance and compatibility with almost every Operating System (OS). As mentioned before, Docker allows deploying applications in separate containers independently and in different languages. It also reduces the risk of conflict between languages, libraries or frameworks.

To implement the solution, we have designed a *Dockerfile*, a script that includes all the necessary steps to install the system in the virtual environment that will constitute the container. Essentially, a Docker image is a read-only template used to build containers, thus allowing storage and shipping applications. This image is also configured to interface with the I/O elements of the system. In our case, the system must interact with the sound system architecture of the OS to access the sound signal from the microphones. Another feature regarding portability is how the image can be built. A Docker image can be built locally or remotely. We opted for the latter option to significantly reduce the deployment time of the system since Docker also contains tools to pull images built remotely. With the image downloaded in the platform, the last element to discuss is the container itself, which will represent the ATR application. A Docker container is a standardised, encapsulated environment that runs the application and is managed using the Docker API or Command Line Interface (CLI). This container will represent an active instance of the application itself. To manage the containers, we have designed a group of scripts that use the Docker CLI, initialising the container with specific attributes such as the ID of the robotic platform, stopping a container and also updating the Docker image. However, despite having converted our system into a containerised and self-contained application, one element, the UDEV rules, is still needed in the target system. This element is discussed below as a fundamental tool to improve the system's robustness.

4.3.2.2. Improving the system's robustness

After addressing the system's portability, the last feature of the system to address is its robustness. To do that, we must delineate what situations our design would face that test this characteristic. Following the design proposal shown in Chapter 3, one of the situations that the system might face is a disconnection of the soundcard placed in the detachable arm of the robot. Another possible scenario might be an unexpected restart of the robotic platform. In those cases, the system should be capable of restarting at the same time as the rest of the subsystems present in the robotic platform. With these concepts in mind, it is possible to define more concretely what we want to accomplish: deeper device control, which allows one to determine what state the system's devices are in, whether they are disconnected, whether they have been reconnected, or whether the robot is identifying them while rebooting. Some GNU/Linux-based operating systems employ UDEV rules for this purpose. Section 4.1.1 demonstrated how these rules could assign devices a unique identifier, allowing them to be consequently assigned to their corresponding ChucK node. In addition to this functionality, UDEV rules enable the execution of scripts and processes when a device is connected or disconnected. Thanks to this feature, we take the first step to increase the system's robustness.

Despite this functionality, there is an added problem, as the programs that the UDEV rules can launch have to be short-lived. These processes cannot be kept active for an indefinite period, as this would prevent the rest of the rules from loading. In those cases, the processes to be launched are usually firmware updates and scripts that perform a brief tuning of the connected device. However, an option is to keep the process in the background: make it a *systemd* process. Systemd[23] is a set of system management daemons, libraries, and tools designed as a central management and configuration platform for interacting with the GNU/Linux operating system kernel. It provides the following three general functions: (i) A system and service manager, which manages both the system, such as by applying various configurations and its services; (ii) a software development platform, which serves as a basis for the development of other programs, and (iii) a connection between applications and the Linux kernel, which provides various interfaces that expose functionality provided by the kernel. We create a unit configuration file called *service* that encodes information about how the ATR process is controlled and supervised by systemd. And at this point, having converted the system into a docker container poses an extra advantage, since a container is easier to diagnose and launch through systemd than a set of ROS nodes. Through this file, the programmer cannot only define what to do when the service starts (in this case, launch the container) but what to do when it's stopped, enabling stopping the service safely. And finally, through a service file, you can define the conditions to be met to launch the service. In our case, the process could only be started when the OS detects all the connected microphones. We have control of this by storing the IDs of the disconnected microphones in a plain file. This way, if multiple microphones disconnect, but just one is connected again, the service would not trigger the ATR. In conclusion, turning the ATR into a service gives the user and the operating system more control over launching, stopping, and diagnosing if any service-related problems have arisen since systemd stores all the information related to the service in logs.

4.4. Summary

Throughout this chapter, we have described the technical details regarding the software of the ATR system. First, Section 4.1 detailed the different phases that comprise the gesture recognition system, listing the various stages that compose it. The first stage involved acquiring sound from the microphones (SA). Afterwards, we proceeded to describe the phase in which the system detected the touch gesture. In this stage (TAD), the SNR and the capacitive sensors intervened, so the system could minimise false positives, i.e. sounds that are not considered physical contact. Next, we described the feature extraction phase (FE), in which the touch sys-

[23] https://www.freedesktop.org/wiki/Software/systemd/

tem delimited the duration of the physical contact and proceeded to extract the main features of the sound signal. Then, we explained how the system generates an instance representing the physical contact (DC), which was then fed into a dataset for the next step, the classification through machine learning (TC). In this last phase, a machine learning framework was used to evaluate whether the information from the audio signal could be employed to differentiate between different types of touch gestures.

Next, Section 4.2 details how we tackled the problem of localising physical contact. Two different approaches were explored. The first was through, again, machine learning techniques. We proposed to convert the contact location into another label to be classified by the system. For that, we offered two approaches: by keeping the two labels independent as two different multi-class problems and employing machine learning algorithms that establish a relationship between both labels. The latter approach is called multi-target. In addition to using machine learning techniques to solve the problem, we proposed introducing signal analysis techniques to find the contact location with more resolution. For this purpose, we used features such as the time difference between the audio signals perceived between microphones. Among the options available, the technique we proposed was the SRP-PHAT algorithm, based on modelling the surface where contact occurs as a probability map.

The last block of this chapter, Section 4.3, consisted of the complete integration of the system into a robotic platform. First, we described the modifications made to the system to adapt it to an online setting. For this purpose, a new module has been created that integrates several machine learning frameworks to predict instances of unlabelled tactile gestures. Finally, we describe how the system is integrated into the software architecture of the robots and the modifications made to the touch system to increase its robustness and portability, two features we consider fundamental in a STS system.

Chapter 5

Acoustic Touch Recognition System Performance Experiments

This chapter describes the various tests designed to assess the performance of the proposed touch recognition system. Each section reflects the evolution of the ATR, and every experiment contained in them is designed to evaluate the different elements that compose the system. Section 5.1 starts with a proof of concept of the touch gesture classification system presented in Section 4.1 using only one microphone. Afterwards, the next sections contain the performance tests of the touch localisation techniques described in Section 4.2. The first, Section 5.2, consists of the machine learning approach and is tested in conjunction with the touch classification system. The second one, Section 5.3, evaluates the sound analysis approach alone. Finally, in Section 5.4, we conducted a more thorough evaluation in order to prepare the system for the human-robot interaction experiments conducted in Chapter 6. This last section is connected to the integration described in Section 4.3.

5.1. Touch Gesture Classification

This section covers the tests performed on the system's first iteration, presented in Section 4.1. We deployed a single microphone version of the system on a robotic platform. **As a proof of concept for the potential of acoustic sensing technology when applied to touch interaction in social robotics, the goal was to test the ATR system's capacity to recognise**

various types of tactile gestures. The proposed system must distinguish different tactile gestures while offering similar accuracy results to the techniques presented in the literature.

5.1.1. Methods

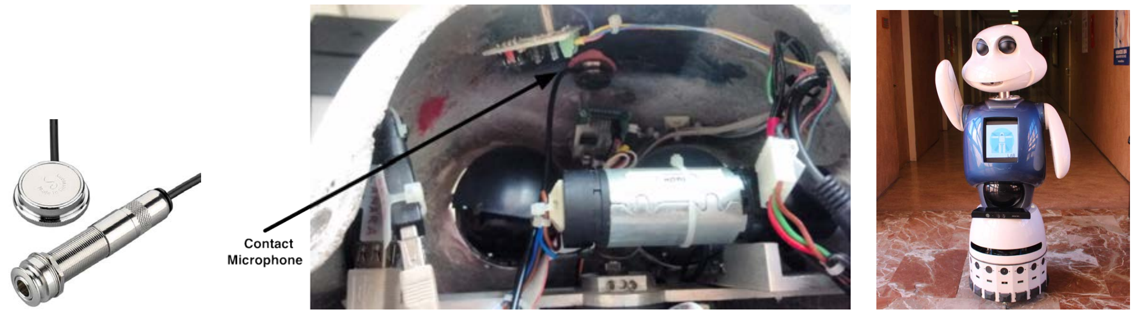
The experiments involved different users performing a series of tactile gestures on a robotic platform, in this case, the Maggie robot [139]. The experiments essentially consisted of the following: First, each user was given information on the definition of the gesture and a demonstration of how to do it on the platform, which was presented on the robot's integrated tablet. Finally, the user performed each of the gestures several times. It is important to note that the experiment was carried out individually by each user without them having previous information about what they were going to do. After this brief preview, this subsection details the touch gestures chosen for the experiment and explores the method of creating the dataset in more depth.

5.1.1.1. Experimental setup

The social robot used for this work was the Maggie robot (see Figure 5.1-c)). Described more in detail in Section 3.1.1, the Social Robotics Group developed it at Carlos III University (Madrid) [139, 197] as a research platform aimed at research on HRI. Regarding the microphone setup for this experiment, only one contact microphone was placed inside the robot's head, beneath the fibreglass shell (see Figure 5.1-b). Due to the physical properties of this part of the shell —concave and rough—, it was necessary to use clay to smooth and homogenise the surface, thus improving the contact between the microphones and the shell. The setup was described in detail in Section 3.2.1. Due to its height (135cm), in order to interact with the robot, each participant stood close to the sensed area, in this case, the shell covering the robot's head, so that they could easily touch it. Furthermore, participants faced the robot so that they could see the screen on its chest, which displayed video demonstrations of the touch gestures the user had to perform.

5.1.1.2. Set of touch gestures

Previous studies proposed different sets of touch gestures recognized during HRI, emphasizing the relevance of non-verbal communication in this kind of interaction [198, 199]. Among these works, the one proposed by Yohanan et al. [17] stands out by presenting a complete dictionary of touch gestures composed of 30 items extracted from human-animal interaction and



(a) Contact microphone.

(b) Contact microphone inside the robot's head.

(c) Maggie robot.

Figure 5.1: Robotic platform and the integrated contact sensors.

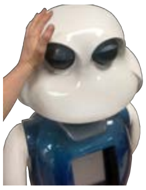
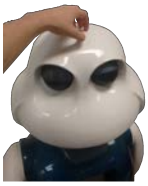
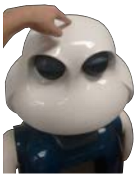
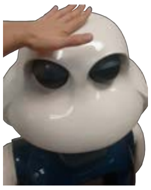
social psychology literature works. Derived from this line of work, Altun et al. [7] devised a set of 26 gestures. Earlier, Silvera et al. [53] proposed using a set of 6 gestures to achieve more atomic expressions distinguishable by their EIT-based artificial skin.

For this experiment, we adopted a set of touch gestures based on the one presented in Silvera's work, considering those more apt for interaction with the social robotic platform used for this experiment. We chose to discard the 'push' and 'pat' gestures, the former for having a very similar sound fingerprint with respect to the 'tap' and 'slap' gestures [200] and the latter to avoid damaging the platform. The rest of the gestures were as follows. A 'stroke' is used to convey empathy, 'tickle' is associated with fun or joy, 'tap' could transmit warning or advice, and finally, 'slap' might be associated with discipline [54]. Table 5.1 shows a thorough classification of the touch gestures according to their contact area, perceived intensity, duration and user intention. Additionally, despite using a set of four gestures, the system was designed to easily adapt and incorporate new gestures in case new applications require it. This could be achieved by training the classifiers with new sets of touch gestures, as is shown later in this chapter, in Section 5.4.

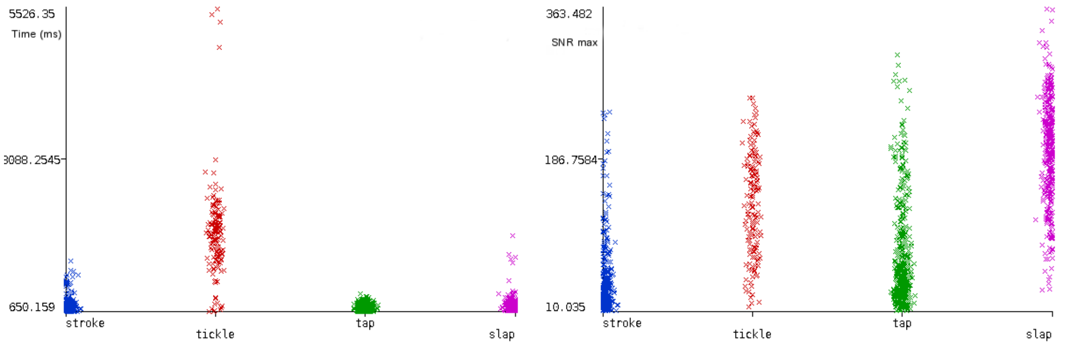
5.1.1.3. Procedure

The dataset designed for this experiment consisted of 1981 touch gesture instances collected from 25 different users. The dataset was divided into a train set that contained 1347 touch gesture instances from 10 users. More specifically, this set is composed by 360 'strokes', 153 'tickles', 463 'taps', and 371 'slaps', representing a 70% of the total amount of instances. On the other hand, the test set was composed of 634 new touch instances performed by 15 users, different from those in the validation dataset.

Table 5.1: Characterization of the touch gestures employed over the Maggie robot [139]. The last column shows an example of how each gesture can be performed.

Gesture	Contact Area	Intensity	Duration	Intention	Example
Stroke	med-large	low	med-long	empathy, compassion	
Tickle	med	med	med-long	fun, joy	
Tap	small	low	short	advise, warn	
Slap	small	high	short	discipline, punishment, sanction	

The interaction between the robot Maggie and the users occurred as explained below. First, it is important to note that a supervisor conducted the experiment and the users participating in the data collection interacted one at a time. Then, the supervisor gave instructions to each user related to the zones —in this case, any point of the robot's head surface— and the kind of touches to be performed over the robot's surface. Afterwards, the robot Maggie showed a video tutorial using the tablet built into its chest. The video display how to perform one of the gestures mentioned by the supervisor. We chose to show the videos as a way to standardize how users should perform the gesture, as people from different cultures could perform gestures in different manners. Once the video finished, the user performed that gesture on the robot as often as he/she wanted. As explained in Chapter 4, the sound features obtained during the interactions are labelled with the name of the gesture performed and stored in a dataset. Finally, this process was repeated for each of the remaining touch gestures. The order of the touch gestures was randomised per user.



(a) Gesture (*x* axis) *vs* Duration in ms (*y* axis).

(b) Gesture (*x* axis) *vs* Signal To Noise Ratio maximum (*y* axis).

Figure 5.2: Visual interpretation of some of the features of the training set.

After finishing the data gathering, we preliminarily analysed the sound features from the instances to evaluate if there were differences between the touch gestures collected. In Figure 5.2-a) we show first the duration of the different gestures, and in Figure 5.2-b) the relationship between each kind of gesture and the maximum SNR reached. Regarding the duration, we can observe how the 'tickle' gestures span a longer time than the others. Regarding the SNR, this feature is related to the signal amplitude for each gesture, in other words, how strong a touch gesture is with respect to the noise. As shown in Figure 5.2-b), there appeared to be differences among gestures, indicating that this feature was suitable for the proposed machine learning classification task.

5.1.1.4. Evaluation metrics for data analysis

In traditional classification problems, *precision*, *recall* and *F-score* are the most frequent evaluation criteria employed. The *F-score*, also known as *F-measure* or *F-score*, is a single value metric designed to measure the accuracy of a learning system [201]. This metric is derived from the precision and recall measurements [202]. Each measure is computed as shown in Equations 5.1, 5.2, and 5.3.

$$Precision,\, P = \frac{TP}{TP + FP} = \frac{Y \cap Z}{Z} \tag{5.1}$$

$$Recall,\, R = \frac{TP}{TP + FN} = \frac{Y \cap Z}{Y} \tag{5.2}$$

$$F\text{-}score = \frac{2 \times P \times R}{P + R} \tag{5.3}$$

where Y is the *true values*, composed of *true positives* (TP) and *true negatives* (TN), and Z represents the *predicted values*, that contains *true positives* (TP) and *false positives* (FP). Therefore, the *precision* (P) is the fraction of correctly predicted instances among all the predicted instances. On the other hand, the *recall* (R) is the fraction of correctly predicted instances that have been predicted with respect to the number of true values. Since both measures are important metrics regarding the accuracy of a classification system, it is usual to use another metric, the *F-score*, calculated as the harmonic mean of recall and precision.

The previous metrics define only the binary classification problem (true or false classification), but in our case, we had a multi-class setting, as we described in Section 4.1.5. Therefore, applied to a multi-class problem, we would have one of these metrics per class value (e.g. P_{tap}, R_{tap}, $F\text{-}score_{tap}$, etc.). To generalise and obtain a single metric representing our system's accuracy, we used a different version of the F-score measure, the *weighted F-score*. Equation 5.4 shows how it is computed.

$$Weighted\ F\text{-}score = \frac{1}{n}\sum_{i=0}^{q}(F\text{-}score_i \times n_i) \tag{5.4}$$

where n is the total number of instances and q represents the total number of classes. As it can be deduced, this metric not only considers the number of classes but also the number of instances per class and the total number of instances.

5.1.2. Results

The machine learning task was divided into two phases that will be performed consecutively: (i) The first one involves *cross-validation* over the training dataset; alternatively, (ii) a different dataset (the test set mentioned before) was used to assess the performance of the system (typically known as the testing phase). The test dataset contained new interactions and involved a different group of users from those who participated earlier in the creation of the training dataset. Usually, the second approach performs worse than the first, but the resulting accuracy is much closer to the one achieved in an online system version. Since this was the first system test, we thought these test results could give a better insight into the performance of the acoustic technology applied to touch gesture classification.

Table 5.2: Classifiers with the best performance using the training set and cross-validation.

#	Classifier	Description	F-score
1	Random Forest	A set of many individual learners (trees). The random forest combines multiple random trees that vote on a particular outcome.	1
2	Neural Network	Neural Network implementation based on Multilayer Perceptron	0.93
3	LMT	Classification trees with logistic regression functions at the leaves.	0.82
4	CNN	Convolutional Neural Network implementation for WEKA	0.81
5	SMO (SVM)	Implements John Platt's sequential minimal optimisation algorithm for training a Support Vector Machine (SVM)	0.80
6	DL4J	Deep Convolutional Network implemented in Java and Weka	0.76

Finding the best-performing classifier on the test set also gives a preliminary idea about how the classifiers might perform on non-trained data. Usually, a training set is created with $70-80\%$ of the samples destined for the training set and a test set of about $20-30\%$ of the total amount of samples, following the Pareto Principle[24]. With this idea in mind, this dataset contains 70% of train samples and 30% of test samples.

5.1.2.1. Validation results

This step involved finding the most appropriate classifier for the newly created dataset. This is done by applying ten-fold cross-validation to the training set, a validation technique designed to assess how the results of a statistical analysis would generalize to an independent dataset. Usually, five or ten-fold cross-validation is recommended to achieve a good compromise between variance and bias when estimating the error [203, 204]. These first results (see Table 5.2) show a perfect classification using a Random Forest (RF) classifier [205], being this estimator a technique that usually offers good performance.

[24] Details about the Pareto Principle:
https://www.thebalance.com/pareto-s-principle-the-80-20-rule-2275148

As the table shows, the second best estimator was the MLP, which obtained an F-score of 0.93. Logistic Model Trees (LMT) also achieved good performance, but lower than the estimators mentioned before, with an F-score of 0.82. This approach implements classification trees with logistic regression functions at the leaves. A similar performance was achieved by an SVM-based algorithm, providing an F-score of 0.80 using Sequential Minimal Optimization (SMO) [206]. As explained in Section 4.1.5, WEKA also allows the implementation of deep learning-based estimators. In our case, a CNN[25] classifier and a Deep Learning for Java (DL4J)[26] estimator were tested, showing competitive but lower performances (0.81 and 0.762 F-score, respectively).

5.1.2.2. Test results

To check the ability of the classifiers to generalize, we also tested the best-performing classifiers from the previous subsection using an untrained part of our dataset, the test set. Since this dataset was created using different users with respect to the cross-validation set, the classifiers now face an independent set of information. The test results show how the performance of the LMT classifier is the same as in the validation stage, with an F-score of 0.81. This is displayed in Table 5.3. We can also observe how the performance of the RF estimator dropped, now being placed as the second best result, with an F-score of 0.79, closely followed by a Decision Table/Naive Bayes hybrid (DTNB) approach with a value of 0.78. Despite this, the results were promising enough to continue developing this technology.

5.1.3. Discussion

Analysing the results in detail, we found that the top-scored classifiers among the 126 tested coincided with those showing good performance in traditional machine learning works [205, 207]. In our case, RF reached the highest accuracy in validation. However, its accuracy dropped when dealing with the test set. In contrast, LMT provided a comparable accuracy both with the validation and the test sets. Table 5.4 presents the confusion matrix for this estimator, also showing the classification errors obtained in the recognition of each gesture. In most cases, the classifier can distinguish successfully between the four gestures. However, there is still a chance of confusion between some of them, resulting in misclassification. For example, *stroke* gestures tend to be confused with *tickles*, this might be because of their similar duration. Moreover, *strokes* are also confused with *taps* because both gestures present low intensity.

[25] Johannes Amtén's CNN implementation: `https://github.com/amten/NeuralNetwork`

[26] DL4J website: `https://deeplearning4j.org`

Table 5.3: Classifiers with the best performance using the test set.

#	Classifier	Description	F-score
1	LMT	Classification trees with logistic regression functions at the leaves.	0.81
2	Random Forest	A set of many individual learners (trees). The random forest combines multiple random trees that vote on a particular outcome.	0.79
3	Neural Network	Neural Network implementation based on Multilayer Perceptron	0.75
4	CNN	Convolutional Neural Network implementation for WEKA	0.74
5	DL4J	Deep Convolutional Network implemented in Java and WEKA	0.73
6	SMO (SVM)	Implements John Platt's sequential minimal optimisation algorithm for training a Support Vector Machine (SVM)	0.72

Table 5.4: Logistic Model Trees confusion matrix using the test set composed of 634 new touch instances.

Predicted gesture **True gesture**	**Stroke**	**Tickle**	**Tap**	**Slap**
Stroke	**94**	21	33	15
Tickle	6	**122**	5	11
Tap	8	0	**146**	7
Slap	7	0	4	**155**

Concerning the deep learning-based algorithms (CNN and DL4J), they performed acceptably in both sets, entering in the 'top-6' best performers. Despite this, they do not improve the results achieved by traditional classifiers such as RF or LMT. This made sense since most of the works in the literature indicate that deep learning algorithms achieve better results when applied to high-dimensional problems (using raw data as the input) that present thousands of instances [208]. For this reason, CNNs are especially suitable for raw speech analysis [209] and raw image classification problems [210, 211]. Finally, the number of users involved (25) and touch instances (1981) may not seem very large. However, the number of instances is still similar to some of the works in the literature [54, 57]. The following journal publication contains the content from this section.

Publication

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. (Q2)

5.2. Touch Gesture Localisation. Machine Learning Classification Approach

The following experiments investigated the system's implementation in a broader context. While in the previous experiment, the system was only deployed using a single microphone, as a proof of concept, in the following tests multiple microphones were used to cover the rigid robot parts that we expected would be touched most frequently and could contain a microphone. **The objective was to evaluate the system's performance with multiple microphones classifying both the type of gesture performed on the robot and the contact location.** We used the previous approach, multi-class, and we also introduced multi-target machine learning algorithms. The tests from this section are connected to the implementation described in Section 4.2.1.

5.2.1. Methods

Throughout this subsection, we will first discuss how the system was set up on the robotic platforms employed for this experiment, two in this case. Afterwards, we explained the selection of tactile gestures and the method used to obtain the set of samples for this experiment. Finally, we include the multi-target metrics used to evaluate the system's performance.

5.2.1.1. Experimental setup

The integration of the microphones followed a similar process in both robotic platforms, Maggie and Mini (described in Sections 3.1.1 and 3.1.3, respectively). In both robots, we integrated three receivers beneath each robot's shell to ensure they would not hinder interaction. In the robot Maggie, the first pickup was located on the inner side of the shell that forms the robot's head, as in the previous experiment. The other two contact microphones were placed on the robot's left and right shoulders, again on the inner side of the surface, to avoid disrupting the interaction. We again used modelling clay to create a smooth and homogeneous layer

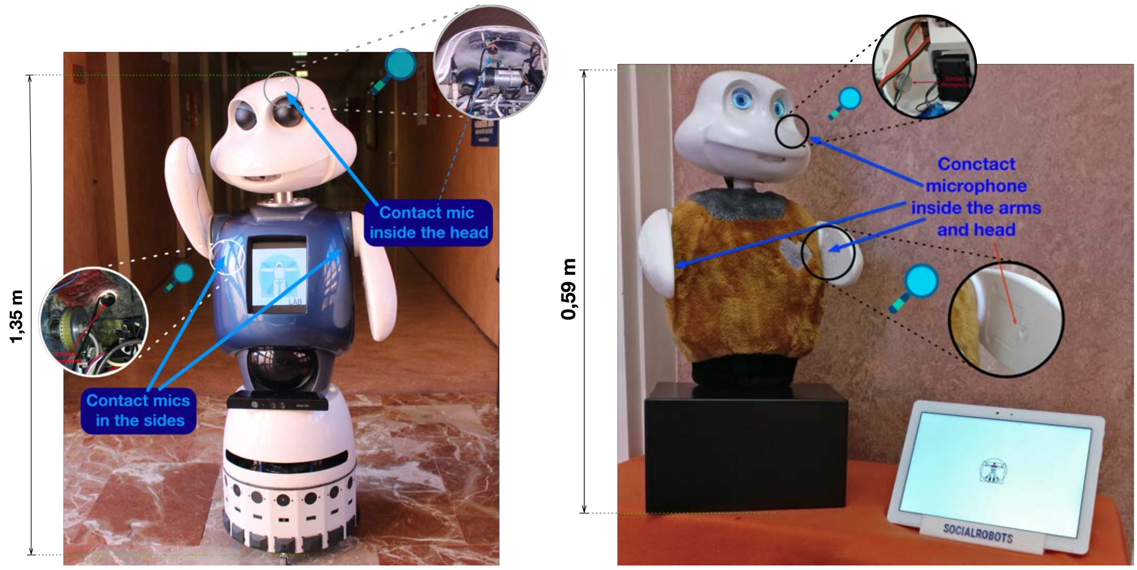

1,35 m
Contact mic inside the head
Contact mics in the sides
0,59 m
Conctact microphone inside the arms and head
SOCIALROBOTS

Figure 5.3: Robotic platforms used in the touch localisation experiment with the location of the acoustic sensors on their surfaces. Left: The social robot Maggie. Right: The social robot Mini.

between the newly installed microphones and the shell since the latter is rough and concave (see Fig.5.3, left). This setup was described in Section 3.2.1. As in the previous experiment, each participant had to interact with the robot standing up due to the robot's height.

Concerning the robot Mini, the contact microphones were located on its rigid surfaces: the head and both arms. The main reason to discard Mini's shoulders was that Mini's torso is made of foam. For this experiment, we wanted to evaluate the performance using one type of material per robot. While Maggie is made from fibreglass, Mini is mostly made from PLA, one of the most used materials in 3D printing. Mini's head, arms, and internal structure were made using this material. As Figure 5.3 right shows, in the head, the microphone was placed on the robot's left cheek. To install the remaining two microphones, we designed and built compartments in the forearm area of the arms. In this case, the robot Mini was placed on a table, and the volunteer was seated in front of the platform, close enough to access all the sensing surfaces easily. This setup and its components were described in Section 3.2.3.

5.2.1.2. Set of touch gestures

These experiments shared the same set of gestures as in the previous tests. Since the main goal was to apply the system to different platforms and evaluate the system's performance using a more significant number of microphones, we decided to maintain the same number of gestures. This set consisted of the touch gestures 'tap', 'slap', 'stroke', and 'tickle'. The experiments shown later, in Section 5.4 explored introducing more touch gestures and how the users interpreted them.

5.2.1.3. Procedure

In this experiment, we used our learning strategy to locate and identify touch gestures. Forty distinct users completed the training part of this learning process. The users were divided into two groups because we used two social robots (20 for each robot). We created two different new datasets since increasing the microphones implied more attributes in the instance; therefore, we could not use the one designed for the previous test with one microphone. Each robot underwent a separate training session with different volunteers. The participants interacted with the robots one at a time, accompanied by a supervisor. The process for both robots during those sessions was as follows: first, each participant entered the testing area. A supervisor gave clear directions regarding the robot's sections to be touched and the motions made during the studies (tables 5.5 and 5.6). The participants were also made aware that before each test, the robots would offer guidance on how to execute each touch gesture by displaying a video explanation for each of the gestures on its tablet (Maggie has one built into its chest, and Mini has an external tablet next to it). The users could then make that move on the robot as often as they like, as in the previous experiment. After the participant performed the gestures, the experiment ended. The video tutorials were designed to standardise how the users perform each gesture because touch gestures may differ between cultures or manners.

Because the system proposed in this work has been integrated into two different social robots, we collected two independent datasets, one per platform. All of the audio signals generated in these interactions were collected by the contact microphones, their relevant features were extracted and these features were then stored in the dataset. As the previous results showed, gathering a sizable amount of data is essential to improve the system's capacity to generalise since learning to recognise tends to be directly correlated with the size of a dataset. We have gathered a total of 3572 instances for the Maggie dataset, and for the Mini dataset, 2777. A summary of the number of instances per gesture and place acquired in the datasets is shown in Tables 5.5 and 5.6.

5.2.1.4. Evaluation metrics for data analysis

We conducted two training stages for the multi-class setting to determine which classifier performs better with gesture localisation and recognition. As a result, we had two different classifiers, one per label —touch gesture and touch location—, each trained with a different

Table 5.5: Touch instances divided into kind of gesture and location in the Maggie dataset.

Maggie	**Head**	**Left shoulder**	**Right shoulder**	**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 5.6: Touch instances divided into kind of gesture and location in the Mini dataset.

Mini	**Head**	**Left Arm**	**Right Arm**	**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

version of the same dataset. To create these versions, we removed either the gesture or location data from the instances in the main dataset. Therefore, each classifier performed distinct, independent tasks, and the accuracy of the entire system depended on how well each of them did on its own. Since we were categorising two different occurrences, the likelihood of such events intersecting is equivalent to the product of their probabilities. To determine the *combined F-score* of our multi-class system, we multiplied the F-scores we received from each label's classifier, as Eq. 5.5 shows.

$$Weighted\ F\text{-}score_{combined} = Weighted\ F\text{-}score_{gesture} \times Weighted\ F\text{-}score_{location} \tag{5.5}$$

Ten-fold cross-validation was used to determine the F-score from each classifier. As in the previous experiments, we used cross-validation since it typically offers a decent trade-off between variance and bias when estimating the error [203, 204].

Because our system had two different tasks —detecting the kind of touch gesture and localising it—, we decided to explore another family of classification techniques: multi-target algorithms. It is important to note that the primary distinction between single-target and multi-target classification is that the prediction can be entirely accurate, partially accurate (with varying degrees of accuracy), or completely inaccurate. Because none of the evaluation metrics designed for single-label classification can fully capture this idea, evaluating a multi-target classifier is more complex than assessing a single-label classifier. In this case, the *Hamming score*, also called *multi-label accuracy*, is one of the most used metrics in this kind of machine learning setting [212]. Furthermore, this metric is one of the main metrics in MEKA, the tool used in this section. As shown in Equation 5.6, this score is defined as the proportion of the predicted correct labels ($Y_i \cap Z_i$) to the total number of labels (predicted Y_i and actual Z_i) for that instance [213]. The overall *Hamming score* is the average across all instances.

$$\textit{Hamming score} = \frac{1}{n}\sum_{i=1}^{n}\frac{Y_i \cap Z_i}{Y_i \cup Z_i} \tag{5.6}$$

5.2.2. Results

This section provides the results for the two possible classification approaches per robotic platform. The results are comprised of six tables, four for the multi-class results and two for the multi-target approach. Despite only showing the top six classifiers, more than a hundred classifiers were trained and evaluated with different configurations. Each classifier was trained ten times, using different configuration parameters to find the best scores. Despite this, we acknowledged that finding the best configuration parameters constitutes a problem on its own, known as Combined Algorithm Selection and Hyperparameter (CASH), outside the scope of our work [142].

5.2.2.1. Results for Maggie robot

This section covers the testing of both classification approaches in Maggie since it was the first robot in which the system was installed. We first cover the multi-class approach followed by the multi-target results.

- *Multi-class algorithms.* In this first testing stage, we compared the performance (F-score) of the different classifiers with the two multi-class versions of the Maggie dataset. For the training phase of the first classifier, the touch gesture classifier, we considered all input features, then omitted the location, and finally added the touch gesture performed as the class label. With this dataset, the *Random Forest* classifier was the best-performing classifier, achieving an *F-score* of 0.858. The results of the top six best classifiers are summarised in Table 5.7. We repeated these tests, considering the location as the class label instead of the touch gesture. In this case, most of the classifiers achieved high performance, as Table 5.8 shows. Given that both classifiers were trained independently and as we mentioned,

Table 5.7: F-score in gesture recognition in Maggie robot (multi-class).

#	Classifier	Description	F-score
1	Random Forest	A set of many individual learners (trees). The random forest combines multiple random trees that vote 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 (SVM)	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

Table 5.8: 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 (SVM)	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	A set of many individual learners (trees). The random forest combines multiple random trees that vote on a particular outcome.	1.000
6	IBK	K-nearest neighbors classifier	0.995

the *combined F-score* is the result of multiplying the F-scores of each classifier, considering the best classifier found for both tasks. In the case of the Maggie robot, this is 0.858 multiplied by 1, which results in 0.858.

Table 5.9: Hamming score in gesture recognition and localisation together in Maggie robot (multi-target)

#	**multi-target classifier (metaclassifier and its associated classifier)**	**Hamming score**
1	BCC based on Random Forest	0.904
2	BCC based on Logistic	0.890
3	BCC based on LogitBoost	0.882
4	BCC based on SMO (SVM)	0.882
5	BCC based on Neural Networks (MLP)	0.875
6	BCC based on J48	0.869

- *Multi-target algorithms.* In this approach, we evaluated the seven different multi-target classifiers present in MEKA using the main Maggie dataset containing both labels. As explained in Section 4.2.1, most of these approaches consist of a meta-classifier that incorporates a multi-class estimator in its core. These multi-class estimators were chosen from those that performed better in the multi-class setting. After testing all the combinations, the best multi-target classifier was the BCC. To improve the clarity of the results, we decided to centre the table on the best-performing multi-target classifier, in this case, BCC, and show its performance with the best multi-class estimators. In this case, the BCC performed best with the multi-class estimator Random Forest, obtaining a Hamming score of 0.904. Table 5.9 shows the six best performing BCC-based multi-class classifiers.

5.2.2.2. Results for Mini robot

This section shows the performance results of the ATR system implemented in the second robotic platform, the social robot Mini. As in the previous section, this one displays the results of the multi-class and multi-target algorithms, respectively.

- *Multi-class algorithms.* The results from the multi-class approach were similar to the ones obtained for Maggie. More specifically, the *Logistic regression* classifier achieved the best F-score, 0.870 (see Table 5.10). Furthermore, *Logistic Model Trees* and *Random Forest* still obtained a competitive F-score, achieving 0.851 and 0.844, respectively. Concerning touch gesture localisation, the system achieved the same F-score in both robots, 1.0 (see Table 5.11). Consequently, the *combined F-score* of the system with this approach is 0.870 for the robot Mini.

Table 5.10: F-score in gesture recognition in Mini robot (multi-class)

#	Classifier	Description	F-score
1	Logistic	Multinomial logistic regression model with a ridge estimator	0.870
2	LMT	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 vote 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 (SVM)	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

Table 5.11: 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 (SVM)	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 vote 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

- *Multi-target algorithms.* We repeated the process to obtain the best multi-target algorithm. As with Maggie, the BCC algorithm performed the best. The results of this meta-estimator

Table 5.12: Hamming score in gesture recognition and localisation together in Mini robot (multi-target)

#	multi-target classifier (metaclassifier and its associated classifier)	Hamming score
1	BCC based on Random Forest	0.912
2	BCC based on Logistic	0.900
3	BCC based on LogitBoost	0.897
4	BCC based on Neural Networks (MLP)	0.893
5	BCC based on PART	0.890
6	BCC based on JRIP	0.889

with the different multi-class classifiers are in Table 5.12. In this case, BCC based on Random Forest achieved the highest Hamming score, 0.912.

5.2.3. Discussion

The results above show that the system still provides high scores compared to the previous tests in the previous section. However, we have to remark that the results have suffered from increasing the number of microphones due to increased features per instance. As explained in Section 4.1.4, these features are associated with each microphone, increasing linearly as each microphone is included in the setup. Focusing on the numbers, the first learning approach has been multi-class algorithms, which obtained a high F-score for both robots. More specifically, for the robot Maggie, what we have called *combined F-score* —the product of both localisation and touch gesture recognition F-scores— was 0.858. In the robot Mini, the system also performed well, achieving competitive results: F-score of 0.870. For the second approximation, we employed multi-target algorithms. Their 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, F-score for the multi-class approach and Hamming score for multi-target algorithms, are not exactly equivalent. Despite this, the Hamming score is similar to the 'accuracy' of a multi-target system according to the literature [212]. For that, we computed the accuracy of the best-performing multi-class algorithms in this section and compared them to the multi-target approach results. The formula for the accuracy is shown in Eq. 5.7.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.7}$$

The computed accuracy in Maggie was 0.851 with a Random Forest multi-class classifier, significantly lower than the best accuracy using BCC, which was 0.9. In Mini, the multi-target approach also performed better, obtaining an accuracy of 0.912 compared to the Logistic Boost algorithm, which reached an accuracy of 0.870 (similar to its F-score). Apart from this, for the problem presented in this work, we argue it is more appropriate to use multi-target classification algorithms (the second approximation) because these algorithms take advantage of the possible influence between the two labels to classify, which leads to better results overall. Besides, this approach avoids training one classifier for each label (location and type of gesture), effectively reducing the computational cost of implementing the system in its online-classification iteration.

Finally, Table 5.13 adds our proposal to the works reviewed in Section 2.1.3 that perform touch gesture recognition. The table shows the platforms used, the technologies implemented, the number of gestures the system is able to distinguish and the results of the techniques using cross-validation and accuracy as a metric. Since it is the metric used in the works from the literature, here we use accuracy again instead of F-score as a metric to be able to compare the performances.

In our work, we computed our system's accuracy by averaging the best-performing multi-class algorithm, Random Forest, in robot Maggie (0.851, as mentioned before) and Logistic Boost in the robot Mini (accuracy of 0.870). In this table, we can observe that Silvera's and Albawi's approaches using sensitive skin achieved lower accuracy. However, in Albawi's approach, we have to indicate that the number of gestures is significantly larger, obtaining the samples from the Corpus of Social Touch (CoST) dataset [16], but having only one contact location. Hughes, Muller and Zhou's proposals also achieved lower accuracy with their systems, in this case, implementing Deep Learning techniques and with a similar number of gestures compared to our approach (four, five and five, respectively). The proposal from Flagg et al., using a conductive fur, reports a significantly higher accuracy, closer to our proposal's, but with one less gesture and only one contact location. Finally, the work proposed by Cooney et al. achieves better results with a larger number of gestures. Nevertheless, we must make two clarifications in this aspect. First, their approach involves a combination of embedded optical sensors and external cameras, thus requiring an external setup. Secondly, some of their gestures come from a combination of gesture and location, so to make a fair comparison, our amount of gestures should be counted as 12 (four touch gestures times three touch locations) instead of four. Summarising the comparison with the literature, despite distinguishing a lower number of gestures, our proposal of an acoustic-based STS system offers competitive results compared to similar works presented in the literature.

Table 5.13: Comparison of gesture recognition using several different techniques including our proposal. The works are ordered according to their accuracy.

Study	**Platform**	**Technologies**	**Num. of gestures**	**Accuracy**
Hughes et al. [59]	Human-animal affective robot	Pressure-sensitive robotic skins	4	0.613
Albawi et al. [57]	Artificial robotic arm	Pressure-sensitive robotic skins	14	0.637
Silvera et al. [54]	Artificial robot arm	Pressure-sensitive robotic skins	6	0.740
Muller et al. [58]	Socially Assistive Robot	Capacitive and pressure-array touch sensors	5	0.740
Zhou et al. [60]	Human-animal affective robot	Pressure-sensitive robotic skins	5	0.761
Flagg et al. [56]	Human-animal affective robot	Conductive fur	3	0.820
Our proposal	Two social robots (irregular and rigid surfaces)	Built-into contact microphones	4	**0.861**
Cooney et al. [61]	Humanoid robot mock-up (foam-covered mannequin)	External cameras, built-into optical sensors	20	0.905

In conclusion, in this test, we proved the system's scalability by integrating three contact microphones in two robots, allowing us to classify the contact location. This required a study of the sound propagation phenomenon in connected rigid parts (the inner structure and shell of the robots). This phenomenon caused different sensors to detect a touch gesture. Ideally, it was expected that the closest sensor registered the highest sound intensity, but this did not always happen, as explained in Section 4.2.1. Another interesting effect was the influence of ambient noises and how the contact microphones register those. During this experiment, this rarely happened as the intensity of sounds propagating in the air was not enough to be captured by the contact microphones. These experiments also presented some limitations, such as that the recognition and localization of touch gestures were currently limited to the robot's rigid parts, making the results unpredictable if the users touch other areas, such as the foam covered by a layer of soft fabric in Mini's torso. The performance shown in this test and this limitation motivated us to develop the module tested in the last section of this chapter, Section 5.4 and test the system in Mini's foam. Another drawback was related to the precision of this touch localisation. As mentioned in Section 4.2, touch localisation systems based on machine learning classification tend to have low precision in locating the source since a label represents it. In the case of Mini, this should not be a problem due to its size. Still, in larger-size robots, like Maggie or Mbot (described in Sections 3.1.1 and 3.1.2), its size offered an opportunity to test other approaches, sound analysis methods, as we show next, in Section 5.3. The following indexed publication includes all the content described in this section.

Publication

Gamboa-Montero, J. J., Alonso-Martin, F., Castillo, J. C., Malfaz, M., & Salichs, M. A. (2020). "Detecting, locating and recognising human touches in social robots with contact microphones". *Engineering Applications of Artificial Intelligence*, 92, 103670. (Q1)

5.3. Touch Gesture Localisation. Sound Analysis Approach

As explained in Section 4.2.2, this approach revolves around achieving higher resolutions of the origin of the contact point. As a result, it is envisioned to be implemented on robotic platforms with larger surfaces (e.g. the platform Mbot, explained in Section 3.1.2), where the information about just the contact area might not satisfy the platform's needs. **The objective of this experiment was to localize touch contacts on a rigid, non-planar surface of a robot.** More specifically, this part of the system was tested on the fibreglass surface of the MOnarCH project robotic platform [140]. The SRP sound localization method is used in this Section as it allows modelling complex and non-flat shells robots and is proven to provide a high measurement rate suitable for real-time estimation [214].

The tests in this section used the localisation analysis module described in Section 4.2.2, connected to the Touch Activity Detection phase. In this section, the system was not evaluated along with the gesture recognition system.

5.3.1. Experimental Setup

The system proposed uses an array of microphones attached to the outer shell of a robot to detect sound caused by the human touch on this surface. More specifically, we propose a case study where the aim is to locate touches on the head of the robot. In this first implementation of the system, a professional audio interface was used to provide the system with an increased sampling rate and a higher audio quality than commonly available sound cards. This section offers insights regarding the hardware platform and describes how the sensors were mounted

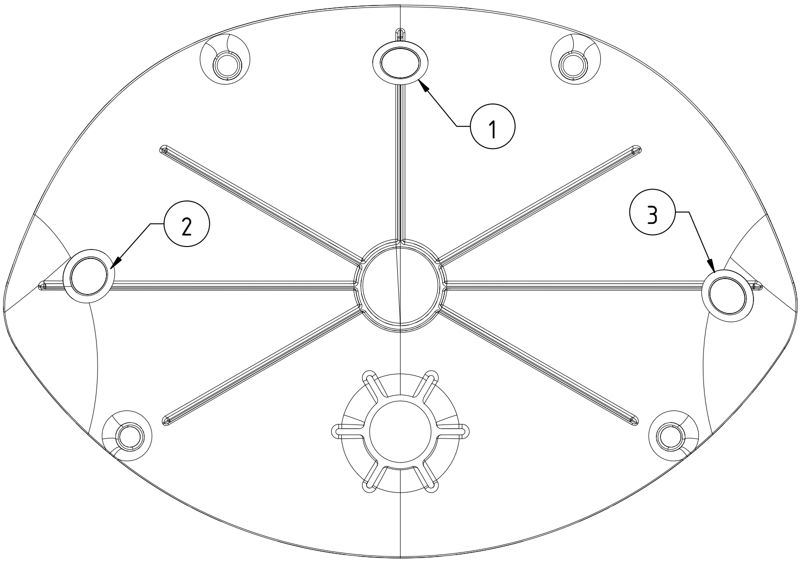


Figure 5.4: Experimental setup of the sound analysis approach with the piezo microphones placed in Mbot's head cover.

on the platform's previously mentioned rigid surface. In this case, the platform was the robot developed in the FP7 MOnarCH project [140], specifically the SO model. The top cover of the robot's head was used for this experiment, as it was a slightly curved area and was considered prone to physical contact. The measurements of this surface were 40 × $30cm$, with a thickness of $0.03cm$. For this scenario, the piezo microphones are placed on the robot surface with a putty-like pressure-sensitive adhesive to preserve the high-frequency signals for experimental purposes. The arrangement of the microphones is shown in Fig. 5.4. For reference, this setup, including its components and its specifications, was explained in Section 3.2.2.

The experiment consists of two phases. First, a calibration was conducted before the data gathering in order to find suitable parameters to test the system. It is important to note that we cannot assure that some system parameters have the optimal values (e.g. the speed of sound on the surface). This is mainly because the robot shell, besides the fibreglass material, has coatings on the inside and outside. After the calibration, we proceeded to test the system. For the final data gathering, we used the fingertips to perform 15 sets of taps over the surface every five centimetres in a straight line between microphones two to three, achieving a total of $15 \times 7 = 105$ contacts.

Table 5.14: Main system parameters.

Parameter	Value	Description

Microphone 1 position
Microphone 2 position
Microphone 3 position
	x = 15.1cm, y = 27 cm
	x = 4.2 cm, y = 15cm
	x = 34.1cm, y = 14cm
		Microphones location (with respect to the grid in Fig. 4.11)

Window size	4098 samples	Amount of samples in each window in the Sound Acquisition.
NFFT size	4098 samples	Amount of samples in each window of the STFT.
Speed of sound	480 m/s	Speed of sound in the material.
Frequency range	0-8000 Hz	Range of frequencies that contain the event.
PHAT coefficient	0.8	Controls the weight of the frequencies in the sound signal.
SRP grid resolution	1 cm	Balances the algorithm resolution.

5.3.2. Parameters

As explained in the previous section, our system had a set of parameters that needed to be tuned to locate the sound source properly. In the current iteration of the system, the calibration of the system was carried out empirically, adjusting the parameters by trial and error. The system's parameters, empirically set values, and descriptions are shown in Table 5.14. They were calibrated taking into account the performance and delay of the system (window and NFFT sizes, and the SRP grid resolution) and the observed frequency spectrum of the signal (frequency range and PHAT coefficient). Finally, the speed of sound was adjusted by observing the time differences of arrival between microphone signals (TDOAs).

5.3.3. Results

The results are shown in Table 5.15. They are separated into two axes, x and y, and the magnitude of the error vector. The first piece of information that could be extracted was that the maximum and minimum average error values were lower concerning the y-axis. The minimum error value in x was at the positions $P = (25, 15)cm$, being $0.93cm$. The y-axis error presented similar minimums to its x counterpart ($P = (5, 15)cm$ had an error of $1.13cm$), but it has higher maximums, as in the case of $P = (20, 15)cm$, where the average error reaches $4.47cm$. This difference in accuracy between both axes might be due to the arrangement of the microphones and the smaller difference in distance of the microphone pairs 1-2 and 1-3 compared to pairs 2-3.

Table 5.15: Localization error e average and standard deviation after performing 15 contacts from mic 2 to 3 every $5cm$ in a straight line (a total of 105 contacts). The last row contains the values for all the contacts at once, regardless of their position. Maximum and minimum values per column are highlighted. The first four columns represent the error in the x and y axes, respectively, while the last two represent the error module. All data is expressed in centimetres.

Position	$\bar{e}_x$	σ_x	$\bar{e}_y$	σ_y	$\bar{e}$	σ
$(5, 15)$	2.80	2.34	**1.13**	1.25	3.25	**2.34**
$(10, 15)$	1.33	**0.49**	2.07	1.71	2.64	1.47
$(15, 15)$	2.33	1.54	2.40	**0.74**	3.48	1.38
$(20, 15)$	2.53	**2.39**	**4.47**	**2.50**	**5.77**	2.13
$(25, 15)$	**0.93**	1.03	4.40	1.06	4.58	1.17
$(30, 15)$	1.93	1.71	1.47	1.36	**2.66**	1.87
$(35, 15)$	**2.87**	0.99	1.33	0.98	3.24	**1.17**
All contacts	2.10	1.73	2.47	1.94	3.63	1.97

Despite the difference in mean error, the system also showed high standard deviation error values on the x-axis. An example of this can be seen in the $P = (20, 15)$ cm, where the standard deviation reaches a value of $\sigma = 2.39cm$. We suspected the coating on both sides of the surface might cause these values. The coating, in addition to other properties of the robot surface, such as its thickness ($3.1mm$), could affect sound propagation and might be the cause of such errors and standard deviation values.

5.3.4. Discussion

In this work, we proposed a system to localize contacts performed on the rigid, non-planar shell of a service robot in real-time, using a set of spatially separated piezo transducers attached to the inner shell of a robot and the Steered Response Power sound source localization algorithm. The system has been tested on the fibreglass surface of a real robotic platform. Table 5.16 compares the different sound-based touch localization systems mentioned in the literature (Section 2.1.2). We decided to establish the comparison with only the systems that did not contain any active transducers and that had values regarding the accuracy, omitting the ones that did not specify them or the ones that relied on touch classification to do the localisation.

Even though these results do not improve the performances from all the works presented in the literature, they would allow a contact localization resolution of $4.5cm$ in the worst case, showing the potential capability of using this algorithm to convert almost any surface of a service robot into a real-time touch-sensing surface. In this sense, it needs to be noted that the surface

Table 5.16: Comparison between passive touch localization systems shown in the literature.

System	Sensors	Surface	Method	$\bar{e}_{x,y}$	$\bar{e}$
Paradiso et al. [24]	Piezo mics	$2.24m^2$ Glass	TDOA	-	2-4cm
Toffee [43]	Piezo mics	$72 \times 72cm$ Wood	TDOA	-	$10.20cm$
ALTo [45]	Piezo mics	50×50 Wood	TDOA	$x = 1.45cm$, $y = 2.72cm$	-
Our proposal	**Piezo mics**	**40 × 30cm Fibreglass**	**SRP-PHAT**	$x = 2.10cm$, $y = 2.47cm$	$3.63cm$

tested for this work is round-shaped, smaller and notably thinner than the surfaces presented in the literature. This proves that using SRP in this kind of surface provides competitive results. Regarding the technique implemented, the SRP algorithm allows the modelling of the surface properties in which the contact is being performed. Another advantage is the capability of live sound source localization due to the fact that SRP provides high-rate measurements.

Regarding the system's limitations, the optimal set of parameters is an open problem known as hyperparameter optimization. We've designed a system that depends on multiple parameters, and even though this allows more possibilities in terms of adaptability to different environments and platforms, this also means that this phase of the system requires a calibration phase that must be carried out each time the environmental conditions change. As a solution and for future work, we propose the implementation of machine learning, more specifically, hyperparameter optimization techniques, that could help with this task. Another limitation was that the system was installed on one surface of the robot, which could be modelled in 2D as a grid despite being curved. In this sense, we plan to extend the system to more complex-shaped flat surfaces present in the robot, trying to cover the platform's whole shell.

Also, it is important to note that the localisation module tested in this section was connected only to the TAD system phase and was not tested along with the gesture recognition system since this development originated as a proof of concept for the technology employed. For this reason, this part of the system cannot be considered fully integrated into the ATR architecture. Nevertheless, the results obtained are considered a successful approximation of this localisation technique to the field of social robotics and therefore a possible addition to the current ATR architecture. At a software level, this system is fully compatible with the touch gesture classification system. Nevertheless, to do this, we would have to address the space that the sound interface occupies since one of the design considerations of the ATR system has been achieving a low deployment complexity. The contents from this section were included in the following publication.

Publication

Gamboa-Montero, J. J., Basiri, M., Marques-Villarroya, S., Castillo, J. C., & Salichs, M. Á. (2022), "Real-Time Acoustic Touch Localization in Human-Robot Interaction based on Steered Response Power", *In Proceedings of the 2022 IEEE International Conference on Development and Learning (ICDL).*

5.4. Online Integration

This section concludes with the experiments testing the performance of the ATR system online. This experiment is linked to the component described in Section 4.3. We devised a more complex experiment for these tests based on the methodology demonstrated in Sections 5.1 and 5.2. But, in this case, only the robot Mini will be involved. This section pursues the following. First, we planned to enable the system to differentiate a more significant number of tactile gestures. More specifically, we explored more thoroughly the possible touch gestures that match the platform. In this experiment, we also explored more in-depth the ability of our system to distinguish between these gestures. After gathering the dataset, we performed a series of alterations on the dataset in order to enhance the system's performance and, this way, achieve an optimal training dataset. Then, **we tested the system's prediction performance online, comparing the approaches we proposed in Section 4.3.1, based on the MEKA and scikit-learn machine learning tools.** This evaluation consists of a preliminary test to verify the system was prepared for the experiments presented in Chapter 6.

Besides these objectives, in this section, we propose two more novelties. In the first place, the robot will conduct the data-gathering process autonomously. The platform will guide the participant during the process, giving the appropriate indications. Despite this, the process will still be supervised by an experimenter to intervene in case of malfunction. The last novelty is related to the surface materials involved where the system was installed. In previous Sections, we conducted our experiments on rigid surfaces due to their ability to transmit sound. More specifically, Section 5.2 implied leaving one relevant surface in the social robot Mini without sensors: its torso made of foam. We understood the importance of this surface and this material since it is softer and is located in a zone more prone to be touched by a user. For this reason, in this experiment, we included this area in addition to both arms, with different materials in our robot setup: plastic and foam.

5.4.1. Methods

The experiments on human-robot tactile interaction that were presented in Sections 5.1 and 5.2 formed the theoretical and practical basis for this experiment. Although it was built on the foundations of both previous experiments, this study had some significant differences from the previous ones regarding data gathering. First, a larger number of samples and a larger set of gestures. The second difference was in the process of gathering these samples since it was the robot managing the complete process, making the data gathering semi-autonomous.

5.4.1.1. Experimental setup

The experiment was conducted inside a closed office with a desk and two chairs, one in front of the desk (user's chair) and the other behind it (experiment supervisor's chair). For this experiment, we used the social robot Mini. The robot, a tablet, a computer screen and a keyboard were on both sides of the desk. The robot and tablet were placed in front of the user. At the same time, the experiment supervisor used the screen and keyboard on the other side of the table to verify the correct functioning of the touch system during the interactions. This setup is shown in Figure 5.5. During the experiment, using voice instructions, the robot indicated to the participants the gestures to be performed, the parts of the body on which they should perform them, and the number of repetitions of each gesture-location combination. The experimenter was there only to answer minor questions at the beginning and restart the system if something unexpected happens. Still, he/she could interfere in the data-gathering process. The different touch gesture/contact location combinations between subjects were done randomly, but we tried to ensure no significant disparity between the total number of instances per combination. For this reason, and knowing the number of participants beforehand, we assigned each participant the touch gesture/contact location combinations.

Concerning the sensitive areas of the robot, there was also a significant change. In the experiments shown up to this point, the microphones were located only in the rigid areas of the robot. In the case of the Mini robot, that left its torso made of foam without any sensors installed. This area is of great interest because it is prone to physical contact. It is also made of foam and covered by a cloth. Furthermore, **foam is a material area where contact microphones have not yet been tested**, as far as we know. Therefore, for this experiment, the three areas of the robot to be equipped with the piezo microphones were both arms and the front of the robot's torso (its 'belly'). The microphone was inserted in a small pocket in the foam made with a sharp instrument. We decided to remove the head from this experiment due to the fragility of the structure of the robot's neck. Because of this change in the setup involving new contact locations and new materials, we could not reuse the dataset from previous tests.

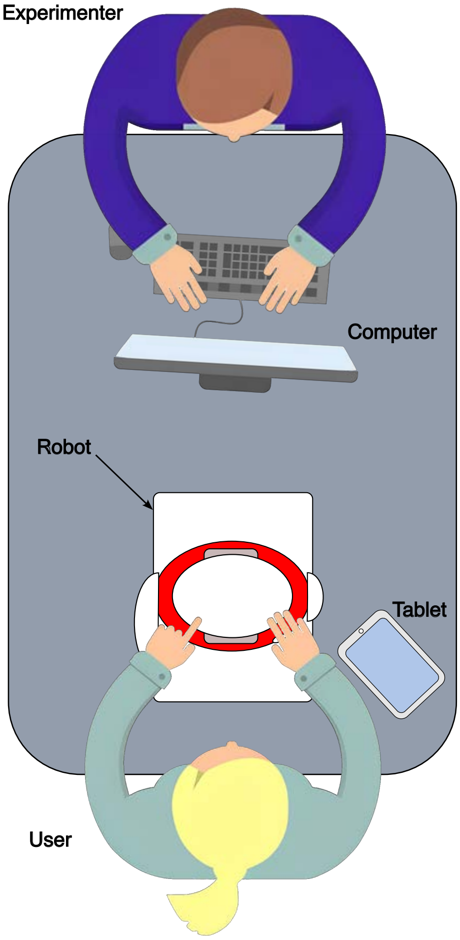
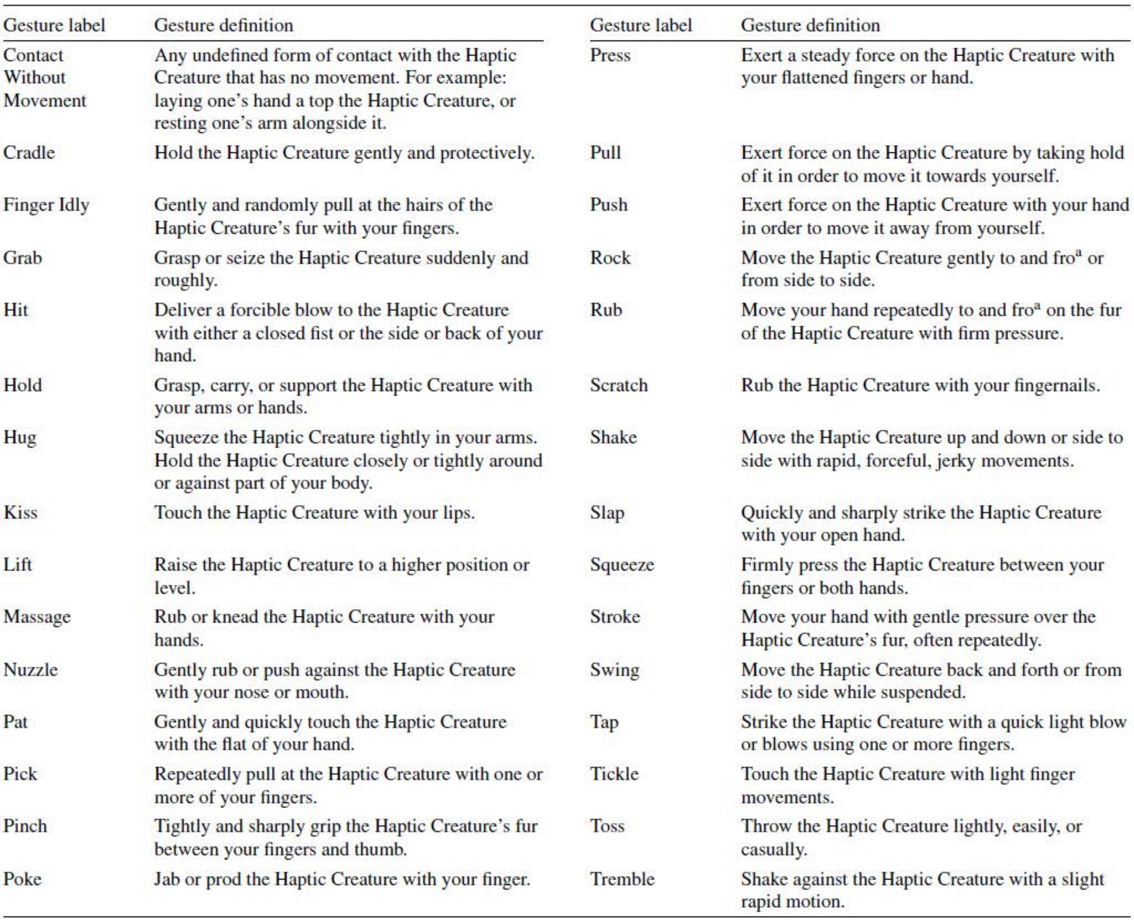


Figure 5.5: Setup of the online experiments with the Mini robot.

5.4.1.2. Set of touch gestures

For this experiment, we decided to increase the number of touch gestures the system can recognise to test the limits of touch recognition. So, we tried to find a compromise before our current set and the one presented by Yohanan. We defined the criteria to discard some of the gestures from the 30-item dictionary shown in Figure 5.6. The first condition to contemplate is the ability of the robot to move. In contrast to the Huggable robot presented in the literature, Mini is a desktop robot, so gestures that imply moving the robot like 'cradle', 'grab', 'hold', 'hug', 'lift', 'press', 'pull', 'push', 'rock', 'shake', 'squeeze', 'swing', 'toss', 'tremble', 'massage' and 'hit' could not be done. Secondly, we tried to focus on gestures done using just the hands, so 'kiss' and 'nuzzle' were also discarded. Then, we discarded 'pick' and 'pinch' because the robot does not have a long fur to perform these gestures easily. Lastly, we considered ruling out 'pat' and 'poke' for having a similar sound fingerprint to the 'tap' gesture (as mentioned in



Gesture label	Gesture definition	Gesture label	Gesture definition
Contact Without Movement	Any undefined form of contact with the Haptic Creature that has no movement. For example: laying one's hand a top the Haptic Creature, or resting one's arm alongside it.	Press	Exert a steady force on the Haptic Creature with your flattened fingers or hand.
Cradle	Hold the Haptic Creature gently and protectively.	Pull	Exert force on the Haptic Creature by taking hold of it in order to move it towards yourself.
Finger Idly	Gently and randomly pull at the hairs of the Haptic Creature's fur with your fingers.	Push	Exert force on the Haptic Creature with your hand in order to move it away from yourself.
Grab	Grasp or seize the Haptic Creature suddenly and roughly.	Rock	Move the Haptic Creature gently to and fro[a] or from side to side.
Hit	Deliver a forcible blow to the Haptic Creature with either a closed fist or the side or back of your hand.	Rub	Move your hand repeatedly to and fro[a] on the fur of the Haptic Creature with firm pressure.
Hold	Grasp, carry, or support the Haptic Creature with your arms or hands.	Scratch	Rub the Haptic Creature with your fingernails.
Hug	Squeeze the Haptic Creature tightly in your arms. Hold the Haptic Creature closely or tightly around or against part of your body.	Shake	Move the Haptic Creature up and down or side to side with rapid, forceful, jerky movements.
Kiss	Touch the Haptic Creature with your lips.	Slap	Quickly and sharply strike the Haptic Creature with your open hand.
Lift	Raise the Haptic Creature to a higher position or level.	Squeeze	Firmly press the Haptic Creature between your fingers or both hands.
Massage	Rub or knead the Haptic Creature with your hands.	Stroke	Move your hand with gentle pressure over the Haptic Creature's fur, often repeatedly.
Nuzzle	Gently rub or push against the Haptic Creature with your nose or mouth.	Swing	Move the Haptic Creature back and forth or from side to side while suspended.
Pat	Gently and quickly touch the Haptic Creature with the flat of your hand.	Tap	Strike the Haptic Creature with a quick light blow or blows using one or more fingers.
Pick	Repeatedly pull at the Haptic Creature with one or more of your fingers.	Tickle	Touch the Haptic Creature with light finger movements.
Pinch	Tightly and sharply grip the Haptic Creature's fur between your fingers and thumb.	Toss	Throw the Haptic Creature lightly, easily, or casually.
Poke	Jab or prod the Haptic Creature with your finger.	Tremble	Shake against the Haptic Creature with a slight rapid motion.

Figure 5.6: Yohanan's touch gesture dictionary [17].

Section 5.1), although, we ended up keeping the first, 'pat', since it appeared in Silvera's set and to verify if the system was able to differentiate it.

We used the *DeepL Translator*[27] to translate the gesture definitions into Spanish. These definitions are derived from the common points between Yohanan's English definitions (see figure 5.6), as translated into Spanish by DeepL Translator, and the definitions of the gestures in the Real Academia Española's dictionary. The names of the gestures used in the experiment are listed below, along with their descriptions.

- Pat: Gently and quickly touch the surface with the flat of your hand.
- Stroke: Move your hand with gentle pressure over the surface.
- Tap: Strike the surface with a quick light blow or blows using one or more fingers.

[27] Translator URL *DeepL Translator*: https://www.deepl.com/es/translator

- Tickle: Touch the surface with light finger movements.
- Scratch: Rub the surface with the fingernails.
- Slap: Quickly and sharply strike the surface.
- Rub: Move your hand repeatedly to and fro over the surface.

5.4.1.3. Procedure

The process of instance extraction is described below. Much of the process had many elements in common with the ones described in Sections 5.1 and 5.2. The main difference is that the robot guided the experiment autonomously. In this case, the experimenter only supervised in case of malfunction. These steps went as follows:

1. Before initiating the experiment, the subjects received a brief explanation about the experiment's aim, what was going to be collected during the process, how they would interact with the robot, and how the robot would interact back at them. In addition, they were informed about the pseudonymised nature of the trial and the responsible use of their data for academic purposes only. Their consent was requested by signing a data protection document.

2. Then, the experiment started, and the robot indicated to the user through voice commands the gesture and the part of his body where it should be performed. Before the user could touch the robot, the platform defined the touch gesture showing the definition on its tablet and through voice using its text-to-speech system. The definition was followed by a video demonstration of how to perform the touch gesture using its tablet. This step helped the participant understand the nature of each gesture.

3. Afterwards, the participant performed this touch gesture-contact location combination repeatedly. After the volunteer finished with this particular combination, the robot gave the next one, and the previous step was repeated. These two steps are repeated for each touch gesture-contact location combination assigned to the participant. We planned to obtain 126 samples per participant (21 different touch gesture-contact location combinations and at least six repetitions of each combination).

4. After finishing interacting with the robot, the participants were asked to complete a post-experiment questionnaire. This questionnaire was composed of eight items: one item, ' Understanding the touch gesture X was difficult', per touch gesture (seven in total), and 'The videos and definitions of the touch gestures were useful to understand them'. Both

Touch gesture	**Number of samples**
Tap	516
Slap	515
Stroke	393
Tickle	489
Pat	496
Rub	424
Scratch	447
Total	3280

Table 5.17: Number of samples per touch gesture.

Body zone	**Number of samples**
Left-arm	1150
Right-arm	1077
Belly	1053
Total	3280

Table 5.18: Number of samples per robot's body zone location.

questions were evaluated using a 5-point Likert scale ranging from 1-'completely disagree' to 5-'completely agree'.

After completing the experiment, a total of 3280 touch samples were obtained from 28 subjects, translating into an average of approximately 117 samples per participant. This average is lower than the 126 samples planned in the experimental phase because five participants could not finish the procedure due to a malfunction during their experiment.

Table 5.17 displays the number of samples for each gesture class collected in the experiment. The class with the highest number of samples was the 'tap', while the 'stroke' gesture had the lowest number of samples. Respectively, Table 5.18 presents the number of samples obtained for each part of the robot's body. The class with the highest number of samples was the 'left arm' zone, while the zone with the lowest number of samples was the 'belly'.

5.4.1.4. Evaluation metrics for data analysis

In addition to the parameters used in Sections 5.1 and 5.2 —F-score for multi-class tests and Hamming Score for multi-target results—, for this section, we will make use of one more multi-target metric: the *Exact Match Ratio*. This metric indicates the percentage of samples correctly classified in all their labels. Its formula is the one shown in the equation 5.8:

$$\textit{Exact Match Ratio} = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i) \tag{5.8}$$

where n is the number of classified instances, and the term $I(Y_i = Z_i)$ represents the instances where all their labels have been correctly predicted. As it happens with the F-score, this metric is a value between 0 to 1. Even though this metric does not distinguish between completely incorrect and partially correct predictions, we used it to provide more information when two or more multi-target classifiers displayed similar performances.

5.4.1.5. Dataset alterations

We proposed for this experiment to apply different processing techniques to the resulting dataset to improve gesture prediction. We employed once again WEKA for this purpose, which contains different filters and data processing techniques. By applying these techniques, we planned to obtain new datasets that were evaluated and compared with each other afterwards. The different filters and methods that have been tested on the global dataset are:

- Principal Component Analysis: the algorithm known as PCA is a dimensionality reduction method. Its function is to reduce the dataset's number of attributes, reducing its dimensionality. This is achieved by linearly transforming the data into a new coordinate system in which (most of) the variance in the data can be expressed with fewer dimensions than the original data. This process results in eigenvectors or principal components [215].

  The first principal components usually contain the most relevant information regarding the data, while the other components could be ignored. This technique also has its drawbacks. In the first place, the original features are more readable than the resulting Principal Components. Also, Principal Component Analysis (PCA) requires standardising the data; otherwise, the technique cannot find the optimal Principal Components. Finally, although Principal Components attempt to cover the most variance among features in a dataset, if the number of Principal Components is not carefully chosen, the dimensionality reduction might translate into losses in accuracy.

- Normalisation: this technique rescales the values of the different attributes to fall within a defined range. We applied this technique to ensure all the numeric attributes of the dataset have their values within the range [0,1]. The normalisation formula applied to the values of the different attributes of a sample is shown in the equation 5.9:

$$x_{normalised} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5.9}$$

Where x is the attribute's value for each instance, x_{min} is the minimum value of that attribute, and x_{max} is the maximum value of the attribute. Changing the values of numeric columns in the dataset to use a common scale avoids distorting differences in the ranges of values. In consequence, it might improve the performance of the machine learning estimator.

- Standardisation: data is scaled to fit a conventional normal distribution during standardisation. A standard normal distribution has a mean of 0 and a standard deviation of 1. Equation 5.10 shows the standardisation formula:

  $$z_{standardised} = \frac{x - \bar{X}}{\sigma} \tag{5.10}$$

  Where $z_{standardised}$ is the standardised value of the instance attribute, x is the value of the instance attribute, $\bar{X}$ is the mean value of the attribute, and *sigma* is the standard deviation of the attribute. In this case, the main motivation to standardise the data is to enable the application of other techniques, such as PCA.

- Synthetic Minority Over-sampling Technique: Synthetic Minority Over-sampling Technique (SMOTE) is mainly used when there is an imbalance between the instances of different classes. This method performs an oversampling of the chosen minority class, generating new synthetic instances from the real instances of that class using transformations and operations between neighbouring instances [216].

  This method makes it possible to create new synthetic instances for each class, thus increasing the number of total instances. It has been decided to apply this method to the trained classes to improve the classifier's prediction by increasing the number of training instances. For each class to which SMOTE has been applied, the number of total instances has been doubled. The main drawback of this technique is that the resulting dataset might cause overfitting in the machine learning estimator used for the classification, worsening the online performance of the system.

- Removing instances with a conflicting class attribute: sometimes, the classifier algorithm may not be able to classify certain classes based on the data gathered correctly. This causes the prediction metrics of that class to be low, but it can also pollute the classification of other classes, causing false positives and false negatives. In this situation, we could remove the conflicting class from the dataset to prevent the rest of the classes from being affected.

- Relabelling instances under the same class attribute: when two or more classes are similar, confusion between them will likely be observed in the classification metrics or the confusion matrices. Instead of classifying these classes separately, they can be merged into a single class by relabelling these instances under a single label. This technique will reduce the number of gestures the system can classify. However, in return, the resulting dataset might take advantage of gestures that might be naturally confusing to the user.

5.4.2. Dataset analysis

Following the methodology shown in sections 5.1 and 5.2, the samples obtained in this tactile interaction experiment were evaluated using various machine learning algorithms. This process was carried out with the tools WEKA and MEKA, obtaining a series of metrics that allowed us to evaluate the performance of various classifiers for the touch gestures and contact locations of the robot. Compared to previous tests, in this subsection, we aimed to obtain more information from the gathered dataset to make all the necessary transformations on the data. We strived to achieve the best performance in the subsequent online tests with this final dataset.

5.4.2.1. Preliminar multi-target evaluation

Among the seven multi-target algorithms tested using MEKA, Tables 5.19 and 5.20 present the best-performing algorithm, NSR, along with the best-performing algorithm from the tests in the previous Section 5.2, BCC. The tables display the metrics obtained for each classifier (Hamming score and *Exact Match ratio*) with the best six multi-class estimators. For both evaluations, we used ten-fold cross-validation. Both tables show that, except for the PART classifier, the metrics obtained in the rest of the classifiers when using the NSR method are equal or superior to those obtained with BCC. Regarding the overall performance of the classifiers, the classifiers that obtained the best metrics (highest values) were the *RandomForest* with NSR and the *AdaBoostM1 - RandomForest* with NSR. The metrics of the two classifiers are practically identical, so the use of *AdaBoostM1*, which is a meta-algorithm designed to improve the performance of other classifiers (in this case, of *RandomForest*), has failed to improve the performance of the default *RandomForest*.

A deeper look at the results retrieved by MEKA showed that for the separate class sets (*gesture* and *place*), the classifier *RandomForest* with NSR obtained a 68.7% touch gesture *predic-*

Table 5.19: Hamming score and Exact Match ratio in gesture recognition and localisation together in Mini robot (multi-target) using the NSR algorithm and ten-fold cross-validation.

#	multi-target classifier (metaclassifier and its associated classifier)	Hamming score	Exact Match ratio
1	NSR based on Random Forest	0.824	0.670
2	NSR based on AdaBoost+Random Forest	0.824	0.669
3	NSR based on LMT	0.778	0.596
4	NSR based on J48	0.741	0.540
5	NSR based on PART	0.736	0.536
6	NSR based on Neural Network (MLP)	0.736	0.525

Table 5.20: Hamming score and Exact Match ratio in gesture recognition and localisation together in Mini robot (multi-target) using the BCC algorithm and ten-fold cross-validation.

#	multi-target classifier (metaclassifier and its associated classifier)	Hamming score	Exact Match ratio
1	BCC based on AdaBoost+Random Forest	0.817	0.652
2	BCC based on Random Forest	0.817	0.650
3	BCC based on LMT	0.776	0.586
4	BCC based on PART	0.742	0.527
5	BCC based on J48	0.737	0.519
6	BCC based on Neural Network (MLP)	0.736	0.506

tion percentage[28]. In contrast, for the contact location, a 96.5% prediction percentage was obtained. These results show that the most complex and problematic group of classes to classify is the set of gestures, while the contact locations have been predicted with a prediction percentage close to 100%. For this reason, the next dataset analysis focused on the prediction of gestures, assuming high accuracy in classifying the robot's zones. This assumption is also seconded by the results presented in Section 5.2.2.

5.4.2.2. Preliminar multi-class evaluation

We observed from the evaluation's results using multi-target classifiers a decline in the system's performance, probably due to the increased number of gestures. The gesture's classification was the factor found to be the issue's root. We chose, following Section 5.1.1, to conduct tests with multi-class algorithms solely focused on the classification of the gesture to determine

[28] Among the metrics retrieved by MEKA in its evaluation report, the *prediction percentage* is a secondary metric that represents the percentage of labels correctly predicted by the classifier.

Table 5.21: F-score in gesture recognition in Mini robot using multi-class algorithms using ten-fold cross-validation.

#	Classifier	F-score
1	Random Forest	0.693
2	LMT	0.610
3	Logistic	0.593
4	FURIA	0.567
5	PART	0.562
6	J48	0.561

Table 5.22: Partial multi-class metrics obtained from the touch gesture version of the main dataset using the Random Forest classifier.

Touch gesture	Precision	Recall	F-score
Tap	0.568	0.589	0.578
Slap	0.669	0.738	0.702
Stroke	0.688	0.702	0.695
Tickle	0.797	0.681	0.734
Pat	0.532	0.476	0.502
Rub	0.733	0.785	0.759
Scratch	0.757	0.779	0.767
Weighted average	0.674	0.674	0.673

the root cause of the issue in more detail. The results gave us a better idea of the transformations we wanted to apply to our dataset. In order to perform the first analysis, we removed the contact location label from the instances in the main dataset as we did in the tests from Section 5.2. We used a set of multi-class WEKA classifiers for these tests to determine which ones provide the best performance. Table 5.21 displays the F-scores using ten-fold cross-validation. The results demonstrated that, as it was seen using multi-target classification, the Random forest classifier was the top performer.

Knowing that the *RandomForest* algorithm is the best performer, we decided to obtain more detailed partial metrics from its validation results. This resulted in the *precision*, *recall* and *F-score* values shown in Table 5.22. For each metric, the two highest scores are marked in green, while the two lowest scores are in red. We can observe that the gestures ‘tap’ and ‘pat’ present the lowest metrics. In addition, the classes ‘rub’ and ‘scratch’ have higher metrics than the rest of the classes, except for the metric precision, being the class ‘tickle’ the one that has reached the highest value in this metric.

Table 5.23: Confusion matrix of the Random Forest classifier obtained from the evaluation with the global dataset of samples.

Predicted gesture True gesture	A	B	C	D	E	F	G
A = Tap	**304**	88	13	6	102	2	1
B = Slap	74	**380**	1	3	56	1	0
C = Stroke	9	1	**276**	11	20	58	18
D = Tickle	5	0	20	**333**	19	37	75
E = Pat	123	97	18	17	**236**	4	1
F = Rub	6	1	53	10	4	**333**	17
G = Scratch	14	1	20	38	7	19	**348**

We also retrieved the confusion matrix of actual versus predicted gestures, shown in Table 5.23. In this table, the rows represent the actual gesture labels for the different samples. In contrast, the columns represent the predicted labels (abbreviated with alphabetical labels) for the samples in the global dataset.

In the confusion matrix, we have highlighted those false negatives for each predicted gesture that exceed S/n samples, where S = the total number of samples of the true gesture, and $n = \mathit{number\ of\ gestures} = 7$. We have considered this number a significant value of false negatives since if one row displays it in all its cells, that would mean that the classifier cannot distinguish between any of the gestures. Thus, we introduce a cut-off value that serves as a criterion to identify which gestures are most confused with each other. Therefore, after setting this threshold, we raised the following observations from Table 5.23:

- More than 1/7 of the 'taps' (74) had been classified as 'slaps' (88) or 'pats' (102).
- More than 1/7 of the 'tickles' (70) were predicted as 'scratch' (75).
- More than 1/7 of the 'pats' (71) were classified as 'taps' (123) or 'slaps' (97).
- Surprisingly, the 'stroke', a gesture that was conflictive in the experiments from Section 5.1, showed only one label, 'rub' (58 predictions), slightly above the 1/7 threshold (56).

5.4.2.3. Evaluation of the dataset alterations

After the previous analyses in WEKA, we could observe that the similarity between the classes 'touch', 'slap' and 'pat' stands out with respect to the other touch gestures. Moreover,

Table 5.24: Averages and standard deviations of the answers to the items: 'Understanding the touch gesture X was difficult'.

Touch gesture	$\bar{X}$	σ
Pat	1.857	1.025
Slap	1.714	0.958
Tap	1.643	0.972
Tickle	1.393	0.673
Stroke	1.286	0.589
Rub	1.286	0.589
Scratch	1.214	0.558

the metrics obtained in this section show that the classes 'tap' and 'slap' are the gestures with the worst results in the classification. These results match with the responses in the questionnaire to the items 'Understanding the touch gesture X was difficult'. As Table 5.24 shows, the class 'pat' was the most difficult to interpret and understand by the subjects, followed by the classes 'slap' and 'tap'. As a result of these observations, we have tried to eliminate the 'pat' instances from some of the generated datasets to evaluate the impact of the absence of this gesture.

Knowing this information, we decided to apply the different filters and techniques explained in the previous section to improve the performance of the offline system and therefore achieve a suitable dataset for the online evaluation. Here we present all the modifications performed in the main dataset. All the modifications have been applied and evaluated in WEKA on the main dataset without the contact location label. Since some of the changes we anticipated involved removing instances from a certain touch gesture, we decided to list them below in terms of the number of different touch gestures. Due to the multiclass results obtained in the previous subsections, we designated the Random Forest classifier as the estimator of choice for the evaluation, and we used ten-fold cross-validation as the validation technique:

- **Seven touch gestures (tap, slap, stroke, tickle, pat, rub, scratch)**: In this case, we did not remove any instances from the dataset. Table 5.25 shows the precision, recall, and the F-score obtained in the classification of the gesture set for the four datasets retrieved after applying PCA, normalization, standardization and SMOTE. These results are compared to those obtained from the global base dataset (without filters). The highest values for each metric are highlighted in green.

  We observed how using SMOTE has greatly improved the results in gesture prediction concerning the results obtained from the main dataset. However, we were suspicious regarding the improvement since, as we mentioned before, it could imply overfitting. We

Table 5.25: Precision, recall and F-score values obtained after performing ten-fold cross-validation with a Random Forest trained with the datasets with seven touch gestures in WEKA. The highest values for each metric are highlighted in bold.

Cross-validation with seven touch gestures (tap, slap, stroke, tickle, pat, rub, scratch)			
Dataset alteration technique	Precision	Recall	F-score
No changes	0.67	0.67	0.67
PCA	0.54	0.53	0.53
SMOTE	**0.83**	**0.85**	**0.84**
Normalisation	0.67	0.67	0.67
Standardisation	0.67	0.67	0.67

also have to highlight how applying PCA to the dataset significantly lowered the results compared to the rest of the datasets. Therefore, we can conclude that, in this case, dimensionality reduction does not positively change the classifier's performance. As for the rest of the filters, normalisation and standardisation, their application did not make a great difference concerning the performance of the main dataset without filters. We finally decided to omit them from the next tests for all these reasons.

- **Six touch gestures (tap, slap, stroke, tickle, rub, scratch)**: in this case, the cross-validation results are presented in Table 5.26. For this evaluation, we experimented with merging instances to the same label based on the prior multi-class evaluation and the results from the questionnaire. More specifically, this was referred to the gestures 'tap', 'slap' and 'pat'. Most of the changes in this aspect involve the 'pat' label and its instances. This was because, as mentioned above, it was the worst-performing class in the previous testing phase classification and the most confusing to identify by the users. Also, according to the previous multi-class tests, we estimated it might generate confusion in predicting the instances labelled 'slap' and 'tap'. Therefore, for the derived datasets tested here, instances from the classes 'pat' were merged and treated as 'slaps', these instances were instead relabelled as 'taps' or they were removed from the dataset.

  When applying the SMOTE algorithm to create a dataset that already suffered relabelling changes, we considered those classes that took advantage of the relabelling process, and therefore they were not augmented. From the results in the table, we observed that the dataset in which the instances labelled as 'pat' were removed and SMOTE was applied was the one that achieved the best results in gesture prediction.

- **Five touch gestures**: Finally, Table 5.27 shows the evaluation results for the modified datasets that contained five classes. Due to the similarity observed between the classes

Table 5.26: Precision, recall and F-score values obtained after performing ten-fold cross-validation with a Random Forest trained with the datasets with six touch gestures in WEKA. The highest values for each metric are highlighted in bold.

Cross-validation with six touch gestures (tap, slap, stroke, tickle, rub, scratch)			
Dataset alteration technique	Precision	Recall	F-score
Relabel (pat to slap)	0.67	0.38	0.48
Relabel (pat to slap) + SMOTE (except for slap)	0.78	0.86	0.82
Relabel (pat to tap)	0.73	0.86	0.79
Relabel (pat to tap) + SMOTE (except for tap)	0.79	0.78	0.78
Remove pat + SMOTE	**0.89**	**0.89**	**0.89**

Table 5.27: Precision, recall and F-score values obtained after performing ten-fold cross-validation with a Random Forest trained with the datasets with six touch gestures in WEKA. The highest values for each metric are highlighted in bold.

Cross-validation with five touch gestures (slap, stroke, tickle, rub, scratch)			
Dataset alteration technique	Precision	Recall	F-score
Relabel(pat, tap to slap)	0.91	0.97	0.94
Relabel (pat, tap to slap) + SMOTE (except slap)	0.93	0.96	0.95
Remove (pat) + Relabel (tap to slap) + SMOTE (no slap)	**0.94**	**0.97**	**0.95**

‘tap’, ‘slap’ and ‘pat’, and also because ‘slap’ was the label that was predicted better, we decided to test the effect of relabelling the ‘tap’ and ‘pat’ labels as slaps. In another combination, we removed the instances labelled as ‘pat’ (the label that shows the worst performance and the one the participants understood less, according to Table 5.24) and relabelled the taps as slaps. We followed the same criteria as in six-gesture datasets when applying SMOTE to a dataset that already had relabelled instances.

As the results show, the dataset in which the *pat* class has been removed, the taps were relabelled as slaps and with SMOTE, has the best cross-validation results, closely followed by datasets that relabelled their taps and pats as slaps and were modified by SMOTE. Despite their good performance, these datasets only improve the number of gestures achieved in previous sections by one (from 4 to 5, by adding ‘rub’ and ‘scratch’ and losing ‘tap’), which is a critical factor in order to finally use them as training datasets for the online module of the system.

5.4.3. Online module results

After pre-evaluating with MEKA and WEKA, and then altering the dataset using WEKA, we proceeded to test the performance of the online module designed in Section 4.3. For that, we started by conducting a brief evaluation of the performance of the scikit-learn node to check how this library performs with the altered datasets. By providing information regarding which label was predicted the best, touch gesture or contact localisation, these results helped us to **determine the order in which the scikit-learn Classifier Chain designed for the online module should classify both labels.** Afterwards, **we tested the system online, comparing the two libraries implemented for the online module: MEKA and scikit-learn.** The experiment tested both variations of the online module by making them simultaneously predict a series of contacts made by a user who did not participate in the previous data-gathering process.

5.4.3.1. Scikit-learn module evaluation

We selected the best four datasets for this evaluation based on the classification metrics collected in WEKA. These datasets, ordered according to the number of touch gestures, are the following:

1. **Main dataset** + **SMOTE**: Seven different touch gestures (*tap*, *slap*, *stroke*, *tickle*, *pat*, *rub*, *scratch*).

2. **Main dataset + remove (pat) + SMOTE**: Six different touch gestures (*tap*, *slap*, *stroke*, *tickle*, *rub*, *scratch*).

3. **Main dataset + Remove (pat) + Relabel (slap, tap = slap) + SMOTE (except for slap)**: Five touch gestures (*slap*, *stroke*, *tickle*, *pat*, *rub*, *scratch*).

From now on, to abbreviate the names of these datasets, we will refer to them according to the number by which they are ordered in the list above (e.g., the main dataset + SMOTE will be renamed 'dataset 1'). Since we are building our Classifier Chain manually, for this phase, we decided to split the evaluation into two multi-class problems, one for the touch gesture and the other for the contact location. Since scikit-learn does not provide tools similar to WEKA's 'experimenter' to test multiple algorithms at once, we opted for using the algorithm that offered the best performance in the previous tests, the Random Forest classifier. The evaluation tests were carried out using ten-fold cross-validation. Tables 5.28 (centred on the touch gestures) and 5.29 (centred on the contact location) show the results from the evaluations.

Table 5.28 contains a comparison of the *precision*, *recall* and *F-score* values per touch gesture, as well as weighted average metrics for each tested dataset. The highest values per column are highlighted in green, while the lowest numbers are marked in red. The following are the main observations gathered from the gesture-centred table:

- **Results from the model trained with dataset 1 (seven touch gestures)**:
  1. Presents the best partial results for the 'tickle' label.
  2. It is the worst classifier in terms of precision, recall and F-score for the 'rub', 'scratch', 'slap' and 'tap' gestures.
  3. As expected, the weighted average metrics (WA in the table) are the worst among the three datasets tested.
- **Results from the model trained with dataset 2 (six touch gestures)**:
  1. It has the best values for rubs, strokes and taps.
  2. It shows the best precision values classifying slaps.
  3. This model does not have any lowest value (in red).
  4. This dataset offers the best weighted metrics among all the models.
- **Results from the model trained with dataset 3 (five touch gestures)**:
  1. This model shows the highest precision, recall and F-score values for the 'scratch' gesture.
  2. This model shows the highest recall and F-score values for the 'slap' gesture.
  3. It has the lowest precision, recall and F-score values classifying strokes and tickles.
  4. The model has the second-best scores for weighted precision, recall and F-score.

Consequently, we conducted the same analysis for the contact locations for the touch gestures (see table 5.29). The table presents the best metrics highlighted in green and the worst metrics in red. The following are the observations raised from the table for each contact location classification model:

- **Results from the model trained with dataset 1 (seven touch gestures)**:
  1. The model has the highest precision and F-score values for the 'belly' contact location. However, it also shows this class's lowest recall value.

Table 5.28: Precision, recall and F-score values for multi-class touch gesture classification in scikit-learn. The results were obtained using ten-fold cross-validation with a Random Forest trained with the altered datasets. The highest values for each metric are highlighted in green, and the last column represents the weighted value for the metric.

Dataset 1 (7 gest.)	**Pat**	**Rub**	**Scratch**	**Slap**	**Stroke**	**Tap**	**Tickle**	**Weighted**
Precision	0.799	0.910	0.926	0.872	0.872	0.852	0.943	0.882
Recall	0.783	0.894	0.911	0.859	0.911	0.893	0.924	0.882
F-score	0.791	0.902	0.918	0.866	0.891	0.872	0.933	0.882
Dataset 2 (6 gest.)	**Pat**	**Rub**	**Scratch**	**Slap**	**Stroke**	**Tap**	**Tickle**	**Weighted**
Precision	-	0.934	0.937	0.939	0.890	0.869	0.937	0.918
Recall	-	0.912	0.916	0.903	0.924	0.932	0.913	0.917
F-score	-	0.923	0.927	0.921	0.906	0.899	0.925	0.917
Dataset 3 (5 gest.)	**Pat**	**Rub**	**Scratch**	**Slap**	**Stroke**	**Tap**	**Tickle**	**Weighted**
Precision	-	0.916	0.944	0.932	0.871	-	0.913	0.915
Recall	-	0.900	0.933	0.932	0.905	-	0.908	0.916
F-score	-	0.908	0.938	0.932	0.888	-	0.911	0.915

2. It also shows the lowest recall and F-score values for the 'left arm' contact location, but the best precision.
3. This model shows the lowest precision value for the right arm contact location. Despite this, the model also has the highest recall and F-score value for this contact location.
4. It has the highest precision, recall and F-score weighted values.

- **Results from the model trained with dataset 2 (six touch gestures)**:

  1. This model showed the lowest precision and F-score values for the 'belly' contact location.
  2. This model offers the highest F-sopre values for the 'left arm' contact location.
  3. This model offers the highest precision values for the 'right arm' contact location.
  4. It had the lowest weighted values, but with a difference of 0.008 with respect to the one that showed the highest scores.

- **Results from the model trained with dataset 3 (five touch gestures)**:

Table 5.29: Precision, recall and F-score values for multi-class contact location classification in scikit-learn. The results were obtained using ten-fold cross-validation with a Random Forest trained with the altered datasets. The highest values for each metric are highlighted in green, and the last column represents the metric's weighted value.

Dataset 1 (7 gestures)	**Belly**	**Left arm**	**Right arm**	**Weighted**
Precision	0.978	0.959	0.946	0.961
Recall	0.966	0.942	0.975	0.961
F-score	0.972	0.951	0.960	0.961
Dataset 2 (6 gestures)	**Belly**	**Left arm**	**Right arm**	**Weighted**
Precision	0.939	0.957	0.964	0.953
Recall	0.969	0.947	0.945	0.954
F-score	0.954	0.952	0.954	0.953
Dataset 3 (5 gestures)	**Belly**	**Left arm**	**Right arm**	**Weighted**
Precision	0.957	0.953	0.961	0.957
Recall	0.983	0.950	0.938	0.957
F-score	0.970	0.952	0.950	0.957

1. This model offers the highest recall values for the 'belly' contact location.
2. The *precision* values for the 'left arm' contact location is the lowest among the three models.
3. It shows the lowest recall and F-score values for the 'right arm' contact location.
4. This is the second-ranked model in terms of weighted average scores among the three models.

After studying the advantages and disadvantages of each classifier in comparison to the prediction of gestures and contact locations, we had to choose what model fitted the better for the online scikit-learn module. We finally opted for the model trained with the second dataset for the reasons listed below:

- This is the classifier that best predicts the set of gestures in scikit-learn. Furthermore, the classifier trained with dataset 3 only classifies five gestures, while the chosen model can classify six.
- We considered that the results obtained in the touch gesture prediction from this model are higher enough compared to those from the model with seven classes to sacrifice one trained class (gesture 'pat'). As we commented before, the instances we removed the touch gesture label that, according to the participants, was most difficult to understand.

- It stands out in the classification of up to three different gestures (*slap*, *stroke* and *tap*). As mentioned, these were the most conflictive touch gestures in terms of the confusion matrix shown in Table 5.23.
- Despite being the classifier with the worst overall metrics for body part prediction, there is only a 0.008 difference in weighted average F-score between this model and the best one.

After deciding the dataset for both models (touch gesture and contact location), we had to make a decision regarding the order of the classifier chain, i.e. which label is classified first, the gesture or the body area. Because the touch gesture prediction outperformed classifying the contact location, we concluded the risk of misclassifying the contact location was lower. As a result, we opted to start the classifier chain by predicting the contact location. Afterwards, the touch gesture classifier would predict the touch gesture using the contact location as an extra attribute.

5.4.3.2. Online classification evaluation

At this point of the Section, we have successfully determined the dataset we would use for the online evaluation. The next step was testing the performance of our module in an online setting. This test was carried out using one extra volunteer that had not previously performed the touch interaction experiment. The participant performed 122 touch gestures on the robot, a number close to the one gathered in the evaluation tests carried out before. During these touch interactions, we were gathering the information simultaneously from both the scikit-learn and the MEKA versions of the system. The data-gathering process proceeded in the same way as it was described in Section 5.4.1, being the only difference that the supervisor was gathering information regarding the gesture asked by the robot, the gesture predicted by the robot and the confidence values both classifiers provided.

We built the corresponding confusion matrices from these results, recorded the confidence values from each correct prediction, and computed the appropriate metrics for the set of collected instances to compare the performance of both libraries. For the comparison between libraries, we separated the analysis of the predictions from the touch gesture from the ones concerning the contact location.

- **Touch gesture predictions**: The left half of Table 5.30 shows the confusion matrix gathered from the online results. In this case, we could observe how almost all the instances belonging to the classes 'rub' and 'scratch' were correctly predicted. Taps, slaps

Table 5.30: Touch gesture confusion matrix results from the online ATR experiments. The left side of the table corresponds to the scikit-learn classifier, while the right side corresponds to the one implemented using MEKA.

	Scikit-learn						**MEKA**					
Predicted **True**	A	B	C	D	E	F	A	B	C	D	E	F
A = Tap	**23**	1	1	0	0	0	**23**	2	0	0	0	0
B = Slap	2	**19**	0	0	0	0	1	**20**	0	0	0	0
C = Stroke	1	0	**17**	0	2	0	2	0	**18**	0	0	0
D = Tickle	0	0	0	**18**	0	2	0	0	0	**19**	0	1
E = Rub	0	0	1	0	**17**	0	0	0	2	0	**16**	0
F = Scratch	0	0	0	1	0	**17**	0	0	0	1	1	**16**

and tickles showed two false positives each, being the stroke class the one the classifier missed the most, with up to three false positives. In summary, for the scikit-learn classifier, 90.98% of the samples were predicted correctly. On the right side of the table we displayed the predictions obtained from the MEKA model. As this matrix shows, compared to the results from the scikit-learn classifier, the classes 'slap' and 'tickle' are the ones that were predicted the best, with only one missed instance. while in the classes 'tap', 'stroke' and 'rub' and 'scratch' we can two misclassifications in each of them. In total, for the MEKA classifier, 91.8% of the contact were correctly predicted gesture-wise.

We obtained the accuracy, recall and F-score metrics from the confusion matrix for each gesture and the average of these three metrics. These values are presented in Table 5.31. As the confusion matrix also reflected, scores for the 'tickle' and 'slap' touch gestures stand out in the scikit-learn case, with more than 0.9 in all their metrics, meaning almost all their instances were classified correctly, and in terms of precision, that not many other touch gesture instances were misinterpreted as 'tickle' and 'slap' touch gestures. The lowest precision values are those of the class 'tap', and the worst recall values are found in the gesture 'stroke'. As for the F-scores, the lowest value is found for the 'stroke' class. Lastly, this classifier showed a weighted F-score of 0.91, in line with the value obtained in the previous cross-validation tests.

As with the confusion matrix, the right side of the metrics table is reserved for the MEKA results. From this side of the table, we can observe that the gesture 'tickle' scored 0.95 on all three metrics, being also the gesture that was classified the best. On the other hand, we obtained the lowest precision values for the 'tap' touch gesture —as it happened with the scikit-learn classifier—, while the lowest recall and F-score values were found for the

Table 5.31: Metrics extracted from the confusion matrix regarding touch gesture classification. The left side of the table corresponds to the scikit-learn approach, while the right side corresponds to MEKA.

	Scikit-learn			**MEKA**		
Touch Gesture	**Precision**	**Recall**	**Fscore**	**Precision**	**Recall**	**Fscore**
Tap	0,885	0,920	0,902	0,885	0,920	0,902
Slap	0,950	0,905	0,927	0,909	0,952	0,930
Stroke	0,895	0,850	0,872	0,900	0,900	0,900
Tickle	0,947	0,900	0,923	0,950	0,950	0,950
Rub	0,895	0,944	0,919	0,941	0,889	0,914
Scratch	0,895	0,944	0,919	0,941	0,889	0,914
Weighted	0,911	0,910	0,910	0,919	0,918	0,918

gestures 'rub' and 'scratch', the latter two having the same value for both metrics. As it can be seen in this table, compared to the scikit-learn model, for these two gestures, the precision and recall metrics are switched, meaning that the MEKA classifier had more issues classifying the true 'rub' and 'scratch' instances than in the previous case correctly. Lastly, the weighted average F-score for the whole set is 0.918.

Finally, Table 5.32 shows the average values ($\bar{X}$) and standard deviations (σ) of the confidence returned from the classifier each time it made a correct prediction. The table is divided into both classifiers, scikit-learn at the left and MEKA at the right. All the values are also divided by touch gesture, and the last row corresponds to the average of all values. The two highest average confidence values and the two lowest standard deviations for each classifier are highlighted in green. In contrast, the two lowest averages and the highest standard deviations are marked in red.

Regarding the scikit-learn approach, the two gestures with the highest average confidence values were the 'slap' and 'scratch' touch gestures. In contrast, the two gestures with the lowest average confidence were 'tap' and 'rub'. As for the standard deviations of the confidence values, the gestures with the lowest deviations in the confidence values were 'tap' and 'stroke', being the classes 'tickle' and 'rub' the ones with the highest values. In the case of the MEKA classifier, for the highest average confidence values, we also had the 'slap' and 'scratch' touch gestures ranking at the top. The lowest average confidence values were found for the 'tap', 'stroke' and 'rub' classes. Concerning the standard deviations, the classes 'tap' and 'stroke' held the lowest two values. In contrast, the 'rub' and 'scratch' touch gestures presented the highest standard deviation, meaning these confidence values from these classes varied the most among all the touch gestures.

Table 5.32: Averages and standard deviation of the confidences obtained from the touch gesture classification. The left side of the table corresponds to the scikit-learn approach, while the right side corresponds to MEKA.

	Scikit-learn		**MEKA**	
Touch Gesture	$\bar{X}$	σ	$\bar{X}$	σ
Tap	0.625	0.098	0.616	0.131
Slap	0.786	0.129	0.722	0.151
Stroke	0.640	0.115	0.616	0.129
Tickle	0.683	0.173	0.666	0.179
Rub	0.549	0.176	0.504	0.182
Scratch	0.751	0.145	0.752	0.238
Average	0.673	0.139	0.646	0.168

Table 5.33: Touch gesture confusion matrix results from the online ATR experiments regarding contact location classification. The left side of the table corresponds to the scikit-learn classifier, while the right side corresponds to the one implemented using MEKA.

	Scikit-learn			**MEKA**		
Predicted **True**	A	B	C	A	B	C
A = Left arm	**38**	2	0	**40**	0	0
B = Right arm	1	**36**	3	3	**36**	1
C = Belly	0	0	**42**	2	0	**40**

- **Contact location predictions**: In this case, the confusion matrix is shown in Table 5.33. In the scikit-learn approach all instances of the class 'belly' were correctly predicted, while in the class 'left arm', we can find two missed predictions and four in the case of the 'right arm' contact location. In total, 95.08% of the contact locations were correctly predicted by this model.

  For the MEKA classifier, the confusion matrix is presented as the right matrix in Table 5.33. In this case, the samples that belong to the class 'left arm' were all correctly classified. However, as it happened in the scikit-learn case, four instances from the 'right arm' contact location were classified incorrectly. Finally, we had one misprediction for the class 'belly'. In summary, as it happened in the scikit-learn case, 95.08% of the labels had been correctly classified for the contact location prediction.

  The multi-class metrics calculated from the confusion matrix above for each class and the overall average values are shown in Table 5.34. Regarding the scikit-learn approach, on

Table 5.34: Metrics extracted from the confusion matrix regarding contact location classification. The left side of the table corresponds to the scikit-learn approach, while the right side corresponds to MEKA

	Scikit-learn			**MEKA**		
Contact location	**Precision**	**Recall**	**Fscore**	**Precision**	**Recall**	**Fscore**
Left arm	0,974	0,950	0,962	0,889	1,000	0,941
Right arm	0,947	0,900	0,923	1,000	0,900	0,947
Belly	0,933	1,000	0,966	0,976	0,952	0,964
Weighted	0,951	0,951	0,951	0,955	0,951	0,953

the left side of the table, the robot's contact location with the best F-score was the 'belly' class, which also had a 1.0 recall. In contrast, the 'right arm' contact location reported the lowest recall and F-score of the set. The last contact location to mention, the left arm, reported the highest precision. An average F-score of 0.951 has been achieved in this case.

The right side of Table 5.34 reports the metrics for MEKA approach. The lowest precision was obtained for the class 'left arm', while the 'right arm' class presented the highest value of this metric. Despite this, this contact location had the lowest recall. Concerning the highest values, the 'left arm' and 'right arm' contact locations presented the highest recall and precision values, respectively. Finally, the 'belly' class reported the highest F-score. As for the average F-score, it reached a value of 0.953.

Table 5.35 shows the average confidence values and the standard deviations of the contact location for both classifiers. In the scikit-learn case, the highest average confidence is obtained for the class 'left arm', although it is also the one with the highest standard deviation value. Conversely, the 'right arm' class has the lowest average confidence and standard deviation. From the right side of the table, which corresponds to the MEKA classifier, we observed that the 'right arm' contact location presented the highest average confidence and also the lowest standard deviation values. Alternatively, the 'belly' class presented the lowest average confidence and the highest standard deviation.

Finally, we also computed the Hamming score using the confusion matrix and the formula from Section 5.2. Table 5.36 shows the results. In this case, as it happened with the F-scores, the MEKA approach shows superior performance in the online classification. However, the values are not that far away, proving that our custom CC is also a valid approach. Regarding the response times, once the classifier was trained (or the model loaded), the span between the user finishing the gesture and the robot responding was almost equal in all the interactions to the *end-of-gesture* time window. As the user might recall from Section 4.1, the system maintained a time window opened in order to consider that the gesture has finished. So, in this aspect,

Table 5.35: Averages and standard deviation of the confidences obtained from the contact classification. The left side of the table corresponds to the scikit-learn approach, while the right side corresponds to MEKA.

	Scikit-learn		**MEKA**	
Touch Gesture	$\bar{X}$	σ	$\bar{X}$	σ
Left arm	0.923	0.149	0.895	0.164
Right arm	0.862	0.135	0.901	0.102
Belly	0.896	0.146	0.871	0.167
Average	0.894	0.143	0.889	0.144

Table 5.36: Hamming scores of the online experiments using the MEKA and scikit-learn tools.

	Scikit-learn	**MEKA**
Hamming score	0.930	0.934

we have to emphasise that the classification was not adding any noticeable delay to this 500ms offset.

5.4.4. Discussion

In this section concludes with the experiments testing the performance of the ATR system online. Following the methodology used in the experiments from Section 5.1, we decided to analyse in more detail the performance of the classification for each touch gesture, trying to understand the difference and similarities between them. For that, in addition to analysing the confusion matrices of the classifiers, we also conducted a short post-experiment questionnaire in which we asked the volunteers whether they had trouble understanding the meaning of the gesture. Regarding the confusion matrices obtained for the main dataset, they revealed certain similarities between some gestures: the more relevant was the one that exists between 'tap' and 'pat'. This could also be observed from the partial precision, recall, and F-score obtained for these classes, where they showed the lowest values. According to the confusion matrices, taps and pats were mostly confused for slaps. It is worth mentioning that precisely these three gestures were the ones that obtained the highest average scores regarding to being understood by users. Moreover, all three interactions are of short duration, and while 'pat' and 'tap' have similar intensity, 'slap' and 'pat' are gestures that have to be performed with the palm, thus producing similar sound signatures.

For this experiment, we proposed introducing different transformations to the main dataset. These techniques included filtering outliers, normalisation, standardisation, PCA and a technique designed to balance unbalanced datasets (SMOTE). In addition, due to the results obtained from the preliminary multi-target and multi-class tests, we proposed removing instances from conflictive labels and relabelling them as similar touch gestures (indirectly removing them). The main motivation for making these changes to the main dataset is to achieve the best classification results possible for the online module. Some of these methods showed significant improvements in the offline results, especially after applying SMOTE, which increases the number of instances for the labels it is applied. Thanks to this method, we started from an *F-score* of 0.67, achieved with the main dataset, to 0.84 using SMOTE with cross-validation, in this case, without removing any label or instance from the dataset. Despite this, we decided to explore further alterations to the main dataset, such as merging or deleting classes and obtaining datasets with fewer total classes but better evaluation metrics. After combining some of these alterations, we could achieve an F-score of 0.89 with six gestures or a score of 0.95 with five gestures.

The second subsection started by exploring the performance of the scikit-learn library, implemented for the online module, on the altered dataset that showed better performance using WEKA. Besides evaluating the library's performance, our main objective was to decide the order of the classifier chain we implemented in this module and to choose a training dataset for our models, both for the one that implemented scikit-learn and the one that implemented MEKA. After evaluating using ten-fold cross-validation three scikit-learn models using the three highest-performing datasets in WEKA, it was observed that the results from the scikit-learn library were almost on pair or were even better than the results achieved using WEKA. For datasets 2 and 3, we obtained F-scores of 0.918 and 0.915, respectively (see table 5.28), while the first dataset, the one that preserved all seven gestures, scored 0.882. In the contact location prediction, the average metric metrics of the three datasets were between 0.95 and 0.96 (see table 5.29). Among the datasets obtained, we opted for the one with 6 touch gestures because of its F-scores in touch gesture classification. After the modifications and augmentation using SMOTE, this final dataset was composed of 5568 instances. Afterwards, we had to choose the appropriate order for the classifier chain in scikit-learn. In this case, based on the scores of the touch gesture and the contact location classification, we started by classifying the contact location without knowing the gesture since it showed more reliability.

The last part of this section tested the performance of online predictions to demonstrate compared both machine learning library implementations: scikit-learn and MEKA. The results obtained from this test were similar to the ones obtained in the cross-validation tests, proving that the trained models were not overfitted. For the online sets, the classifier with *scikit-learn* had an F-score of 0.910 for the touch gesture and 0.951 for the contact location classes, while for the MEKA-based classifier, the F-score for touch gesture recognition was 0.918 and for predicting the contact location 0.953. In terms of Hamming scores, the MEKA-based classifier achieved a score of 0.934 and 0.930 in the case of the scikit-learn approach. As the results showed, the average precision, recall, F-score and Hamming score of both classifiers were very similar, being the average metrics of the classifier with MEKA slightly higher than those of the classifier with scikit-learn, and for this reason, it was the preferred classifier for the experiments conducted in 6. Finally, regarding the confidence values, we observed that most of the average values of the scikit-learn classifier were slightly higher than those obtained from the version implemented using MEKA. Furthermore, most labels had a lower confidence standard deviation for the classifier using this library. As happened before, the difference in the performance of both approaches is not very high.

In summary, we consider that through the experiment conducted in this section, the ATR has been successfully tested to classify 6 different gestures in two different materials simultaneously, foam (soft skin) and plastic (hard skin). We acknowledge that the number of subjects in the online experiment is not high. Still, we must emphasise that in the Section 6.2 experiment, we show the system working in real-time in a real-world application. Although the results from this experiment did not show the system's performance in machine learning metrics, they could be considered a qualitative evaluation of our system.

5.5. Summary

In this chapter, we described the various tests designed to assess the performance of the proposed touch recognition system. Section 5.1 starts with a proof of concept of the touch gesture classification system presented in Section 4.1 using only one microphone. In those experiments, where we used Maggie as the robotic platform, 25 volunteers participated. And after gathering a dataset composed of 1981 samples, the first version of the system achieved an F-score of 0.81, using an LMT classifier in train/test evaluation focusing on touch gesture classification with one piezo microphone and 4 touch gestures ('tap', 'slap', 'stroke', 'tickle'). These results validated the initial proposal; therefore, we could expand the idea to a larger number of receivers.

Afterwards, the next sections contain the performance tests of the touch localisation techniques described in Section 4.2. The first, Section 5.2, consists of the machine learning approach and is tested in conjunction with the touch classification system. For this experiment, we opted to maintain the four initial touch gestures and test the system on two different platforms: Maggie and Mini. Even though Mini also had foam in its chest, for this first test with multiple microphones, we preferred to use rigid materials for both platforms. Each dataset involved 20 users, and in terms of samples, the Maggie dataset comprised 3572 instances and the Mini dataset 2777. With those datasets, we obtained in the multi-class tests an F-score of 0.858 for Maggie and 0.870 in the case of Mini. For the multi-target tests, we had to employ a different metric for the evaluation, the Hamming score, also called multi-label accuracy. In this case, we achieved a Hamming score of 0.904 in the case of Maggie and 0.912 in the case of Mini. Both approaches demonstrated that the system could be successfully scaled using more microphones to cover larger surfaces of a robotic platform. The second one, Section 5.3, evaluates the sound analysis approach alone. By implementing the second touch localisation approach, presented in Section 4.2.2, we explored the possibilities that the sound signal analysis techniques offered to the ATR. We achieved an average error of $3.63cm$ in the Mbot's thin, curved fibreglass surface.

Finally, in Section 5.4, we conducted a more thorough evaluation in order to prepare the system for the human-robot interaction experiments conducted in Chapter 6. This last section is connected to the integration described in Section 4.3. After validating our proposal with the offline evaluations from Sections 5.1 and 5.2, we decided to focus on Mini to create a suitable dataset for online classification. Through this test, we tried to expand further on some aspects that were not covered in enough detail in the previous tests. Some of these aspects were the maximum suitable touch gestures that the system could distinguish in a desktop platform and also whether the system could handle microphones placed on different materials on the same robot: plastic and foam. As a result, we could expand the number of gestures to 6, and we achieved, in multi-target offline cross-validation of the dataset, a Hamming score of 0.934 in online classification. The dataset gathered from this experiment was originally composed of 3280 samples from 28 users, and the final dataset, after the modifications and augmentation using SMOTE, was composed of 5568 instances.

Chapter 6

Human-Robot Touch Interaction Experiments

Chapter 4 described the design and the implementation of the Acoustic Touch Recognition system, and the previous one, Chapter 5, discussed a series of tests to evaluate its performance. This chapter will focus on two applications of the designed acoustic touch recognition system. In this case, we will use it for its true purpose: to enhance research regarding social touch. The two case studies covered in this chapter will revolve around active social touch, the first centred on how to use touch contact and computer vision to affect communication applications, and the second one on how active touching a social robot can affect the users' behaviour.

Section 6.1 combines the information from our touch gesture detection system with a set of face recognisers to make the robot capable of recognising the user's affect display. Before that, a previous experiment will be conducted to evaluate how humans interpret the combination of another person's facial expression and physical contact. The experiment's main objective will be designing an affect recognition system derived from this data to enhance the robot's perceptual capabilities.

The second experiment is presented in Section 6.2, which will test the impact touching the robot has on the user in terms of engagement, intrinsic motivation and fun. For that, we will make the users involved in the experiment play a memory game, and the peripherals to play the game will be the robot itself —meaning that the user will have to touch the robot to play and interact during the experiment—, or an external device based on buttons.

6.1. Affect Display Recognition through Tactile and Visual Stimuli in a Social Robot

The way humans communicate with robots has evolved in recent years with the emergence of new technologies. Using these new technologies, the interactions between these two elements has be enhanced in new ways. Providing a robotic platform with the ability to recognise and express emotions by analysing the perceived stimuli constitutes another step toward achieving a more natural interaction. While text, facial, and voice recognition have become increasingly fluid in recent years, thanks to the development of machine learning algorithms, recognising and expressing emotions via multimodal recognition is a field that the literature could further explore. As Beale and Peter studied, emotions are produced in interpersonal relationships after the first few interactions, implying that it is a gradual process that takes time [217]. Therefore, the ability of the devices with which the user will interact to perceive emotions is an added value because it may generate a sense of trust. This feature becomes essential in personal assistance or education applications, where trusting the 'caretaker' is key in order to follow his/her indications. In this sense, social robots stand out among those devices with educational or assistive care functions.

According to Henschel et al. [218], "a social robot must be able to interact bidirectionally, display thoughts and feelings, be socially aware of its surroundings, provide social support, and demonstrate autonomy". With these considerations in mind, to make a robot *socially aware of its surroundings* and thus *interact bidirectionally*, it appears necessary to equip such devices with the ability to recognise the user's *affect display*: the expression of the user's internal emotional estate[29]. Based on this drive, **the main goal of the work described in this section is to study how a combination of visual and tactile stimuli can influence people's perceptions of affect display and how to apply these findings to a social robot**. In the experiments performed, the subjects had to determine the perceived valence and arousal of simultaneously being exposed to the two stimuli mentioned above. Based on the results from the analysis, we also propose an application for the robot to determine the user's affect display at any given time.

[29] In this work, we will use the definition of affect display introduced by Yohanan et al. [17]. We must clarify that the authors acknowledge that this expression could be faked, but these nuances are out of the scope of this section.

6.1.1. Background

With respect to the works found in the literature focusing on recognising human reactions to stimuli related to affect display, we start with the ones based on visual stimuli. Diekhoff et al. [219], for example, used magnetic resonance imaging to examine how certain images with fearful facial expressions created a bias in participants that altered their perception of emotion recognition in neutral faces. However, several studies have also explored the influence of acoustic stimuli in emotion recognition. For example, Redondo et al. [220] conducted a study in which 159 participants rated 111 sounds regarding valence and arousal level. Vasconcelos et al. [221] investigated the accuracy with which experimentees recognised vocal emotions from nonverbal human vocalisations regarding valence, arousal, and dominance levels. Regarding tactile stimuli, it is worth mentioning the study by Tsalamlal et al. [222] in which the authors evaluated the influence of a haptic stimulus on visual stimuli. To do so, participants indicated the valence level suggested by various facial expressions. At the same time, a stream of air was applied with varying degrees of intensity to their left arm. The authors concluded that the tactile stimuli significantly influenced the experimentees' valence perception.

When considering how to capture the user's affect display during human-robot interaction, we discovered that much of the literature focuses on visual and auditory stimuli. Huang et al. [223], for example, attempted to recognise emotions during human-computer interaction by combining facial detection with an analysis of the user's electroencephalographs. Similarly, Breazeal et al. [224] investigated the recognition of a user's affective communicative intent without focusing on the linguistic content of the speech, instead attempting to recognise prosodic patterns that communicate prohibition, attention, request, and comfort for a robot to analyse. Castillo et al. [225–227] proposed monitoring the facial and gestural expression, activity and behaviour, and relevant physiological data of the elderly to infer and recognise their emotions. The goal was to use emotion control strategies to enhance the care and quality of life for elderly people who want to continue living at home. They used music, colour, and light to stimulate their emotions and shift them into happy and pleasant attitudes.

Despite being scarce, research such as that of Yohanan [17], Altun [7], or Andreasson [168] validated the relevance of tactile stimuli analysis when analysing the user's affect display using a social robot. Finally, Ahmed et al. [228] begin with the premise that emotions affect touch perception and then focus on the perception of emotions by virtual reality agents, concluding that haptic responses can provide a measure of people's experience in human-computer interaction.

Gesture	Definition
Stroke	Move your hand with gentle pressure.
Rub	Move the hand repeatedly with firm pressure.
Tickle	Touch with light finger movements.
Scratch	Rub with the fingernails.
Tap	Strike the with a quick light blow or blows using one or more fingers.
Slap	Quickly and sharply strike with an open hand.
Hit	Deliver a blow with either a closed fist or the side or back of your hand.

Table 6.1: Definitions of the touch gestures used for this experiment.

6.1.2. Experimental Study

To endow a social robot with the ability to respond to the user's affect display, we must first understand how people perceive stimuli. In a typical interaction environment, stimuli tend to appear grouped rather than individually. As a result, evaluating just a stimulus alone could lead to inaccurate results. Based on this premise, a study was planned to collect and analyse the valence and arousal perceived by users when exposed to the target stimuli simultaneously. The visual ones were presented through the appearance of different images on a screen, while the experimenter provided tactile stimuli to make it appear as natural as possible. The users then gave their perception of the valence and arousal level produced by these two stimuli using a graphical user interface designed to automate the data gathering and ease the subsequent analysis.

We define seven kinds of touch stimuli in this study based on their duration, intensity, and form. We chose them from the set of six gestures defined in Section . We also added 'hit' despite its negative connotation since we expected it to have more extreme valence and arousal values, which could help to have a more diverse set of gestures. Table 6.1 summarises the set of touch gestures used in the experiment along with comprehensive definitions.

Regarding facial expressions, we used Paul Ekman's six basic emotions [229] for the simple expressions (shown in Figure 6.1), adding a 'neutral' one. The following expressions with their abbreviations were used in this study: angry (AN), afraid (AF), disgusted (DI), sad (SAD), neutral (NE), surprised (SU), and happy (HAP). In this experiment, we used images from the Karolinska Directed Emotional Faces (KDEF) database [230]. The combination of touch and vision stimuli was the main factor to analyse in this preliminary study. The study's main objective was to analyse the influence of this factor on the valence and arousal perceived by the user. Therefore, we formulated the following hypotheses:

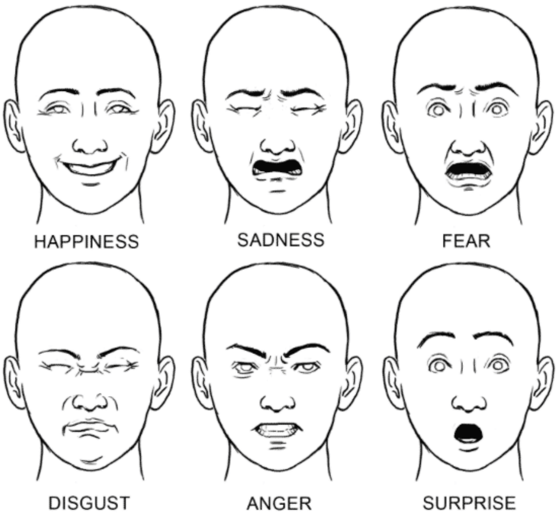


Figure 6.1: Paul Ekman's six basic emotions: happy (HAP), sad (SAD), fear/afraid (AF), disgusted (DI), angry (AN) and surprised (SU) [229].

- **H1**: The combination of tactile and visual stimuli significantly impacts the valence perceived by the user.
- **H2**: The combination of tactile and visual stimuli significantly impacts the arousal perceived by the user.

6.1.2.1. Participants

The study on affect display included 50 subjects, 29 of them were male, and 34 were under 30 years old. None of the participants had any prior knowledge of the experimental procedure, user interface, or images shown during the study. Combining the sets of touch and facial stimuli, we obtained 49 unique combinations. To eliminate bias, we created five cases, each made up of 20 randomly chosen touch and face combinations. Each user was presented with one of these cases, trying to ensure balance among case instances for our dataset.

6.1.2.2. Procedure

First, the participant is conducted to the room where the experiment takes place, then sits down and proceeds to fill out a data protection document, including personal and contact details and the identifier that serves as a pseudonym.

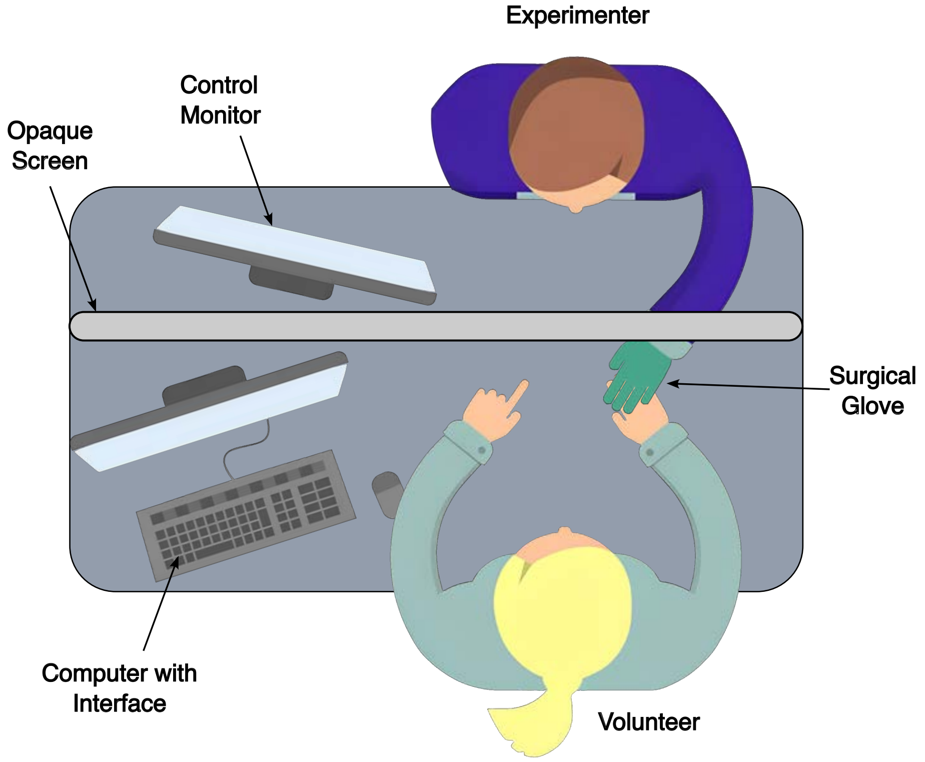


Figure 6.2: Affect display stimuli evaluation experiment setup. The experimenter is hidden behind an opaque screen and wears protective clothing to conceal his/her age and gender.

Then, the experiment began with the participant exposed to the two types of stimuli at the same time: A picture of a person's face with a specific facial expression appeared on the application screen (see Figure 6.3), and simultaneously, the experimenter performed a touch gesture on the user's left arm. The experimenter was behind an opaque screen, and his/her arm was covered with a surgical glove and a long sleeve to prevent the subject from guessing his/her age or gender. The experimental setup for this experiment is shown in Figure 6.2.

As Figure 6.3 shows, the results of valence and arousal levels are plotted on the X and Y axes, inspired by Russell's circumplex model [132]. This model derives from Rusell's work, where he asked participants to categorise 28 emotion terms based on perceived similarities. Russell then utilised a statistical technique to arrange the emotion ratings based on positive correlations, thus forming a circle with similarly linked emotion terms. This multidimensional scaling analysis revealed two bipolar dimensions: valence and activation/arousal. Therefore, any emotion can be represented by a dimension of unpleasantness/pleasantness (valence) and a dimension of high arousal/low arousal (activation). In our interface, both levels in the circumplex range from −100 to 100. The −100 scale represents the most unpleasant valence and the most relaxing in terms of arousal, whereas 100 represents a very pleasant and high arousal level. To modify the

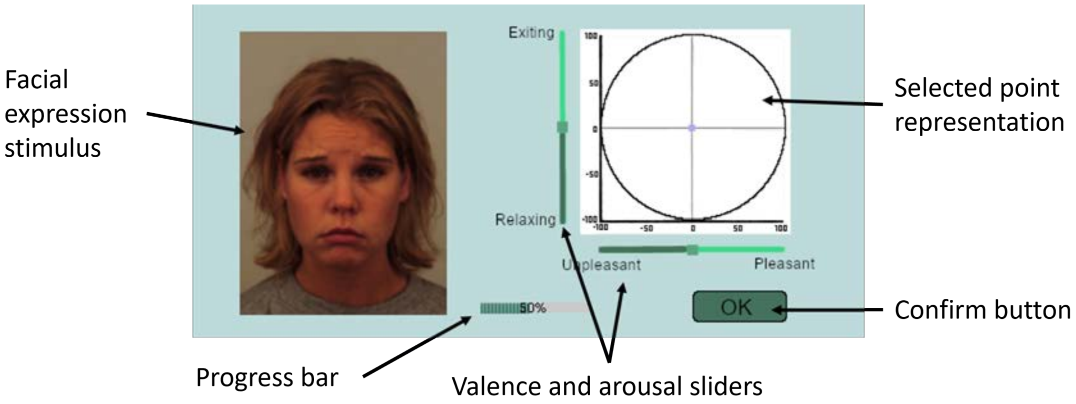


Figure 6.3: Graphic interface designed for the experiment.

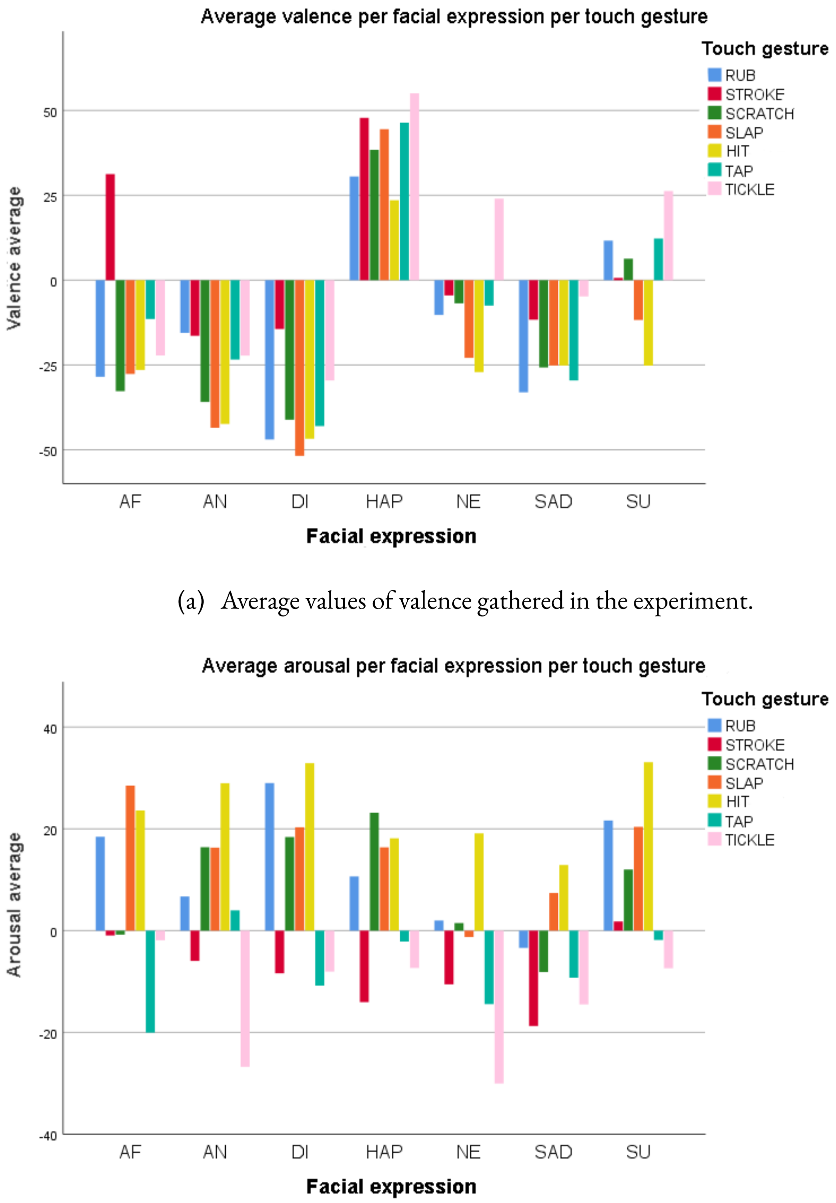
values of valence and arousal, the interface included two sliders attach to each axis, which the user could move freely. After that, the user pressed the "OK" button to continue to the next pair of stimuli. The experiment lasted five to seven minutes on average, with 20 image and touch combinations performed in each case.

6.1.3. Results and Discussion

To analyse the data, we set a significance threshold of $\alpha = 0.05$. The goal of the general analysis of the results from the tests performed on the 50 users was to find a relationship between tactile and visual stimuli and the levels of valence and arousal. To test whether we could perform a MANOVA for both valence and arousal, first, we ensured that all data had a normal distribution using the Shapiro-Wilk method [231] ($p > .05$ in both cases). Afterwards, we check for the correlation between both dependent variables. Since this assumption is not met ($r = .15$, $p < .001$), we perform one ANOVA analysis per dependent variable. This analysis allowed comparing the differences between the means of the different groups. In our case, by performing an ANOVA on the influence of the combination of touch and expression on the value of valence and arousal, we discovered that the combination of the two stimuli had a significant impact on the affect display perceived by the user with respect to the valence $F(951, 48) = 18.068$, $p < .001$, $\eta_p^2 = .477$ and the arousal $F(951, 48) = 5.478, p < .001$, $\eta_p^2 = .217$, thus validating our hypotheses, **H1** and **H2**, respectively.

With these findings, we obtained the means for each combination of stimuli, yielding the results depicted in Figure 6.4. These graphs showed the mean valence and arousal obtained for each gesture and facial expression combination. The ANOVA analysis showed that the combination of stimuli significantly influences valence and arousal. Looking at the results of Figure 6.4a, which shows the average valence obtained in each combination, we observed that the facial emotions 'afraid', 'angry', 'disgusted', 'neutral' and 'sad' had primarily negative values, outweighing the tactile information. These results were consistent with the fact that these facial expressions are commonly associated with negative emotions. However, in the case of the 'afraid' face, the valence obtained from the 'stroke' gesture was positive. Therefore, while facial expressions are relevant in the perception of affect display, they can be affected by the contact performed at that moment, turning an unpleasant feeling into a pleasant one. The same effect can be seen with the 'happy' expression, which aids in perceiving all gestures as pleasant. We can see, however, that the more abrupt gestures, such as 'hits', achieved a lower level of valence than the rest of the touches studied. In the case of the 'surprised' facial expression, we observed diverse outcomes. Because the level of the valence of 'surprised' emotion in Russell's circumflex was low, it can be considered a pleasant or unpleasant expression depending on the user. In this case, where the facial expression is unimportant, we can see how the touch gestures significantly modulate the valence, ranging between 26 and -25.

Complementarily, in Figure 6.4b, we display how the arousal results in more uneven values for each facial expression. For this reason, we decided to group the results by the kind of touch gesture instead of trying to find some patterns, which resulted in Figure 6.5. The figure showed that looking at the touch gestures, the results were more aligned, implying that for the arousal variable, the type of gesture was more significant than the facial expression, in contrast to the data obtained with the valence. In this case, we observed that the 'tap', 'scratch', 'slap', and 'hit' gestures were primarily positive, whereas the 'stroke', 'rub', and 'tickle' gestures were mainly negative. These outcomes were linked to the definitions of each of the gestures. While 'tap', 'scratch', 'slap', and 'hit' are gestures that involve applying pressure to the user's arm, where the intensity is brief but intense, 'stroke', 'rub', and 'tickle' imply a soft gesture on the user with less pressure, resulting in a negative arousal value. In this analysis, we also noticed that, as with valence, the visual stimuli had some influences on the user's perception. In the case of 'tap', for example, we saw that arousal drops to negative values in the presence of 'sad' facial expressions, just as it did with 'scratch'. Finally, we created the *affect_display* database with all the valence and arousal results, which the robot used to estimate the user's affect display.



(a) Average values of valence gathered in the experiment.

(b) Average values of arousal gathered in the experiment.

Figure 6.4: Average values of valence and arousal gathered in the experiment. The horizontal axis shows the facial expressions afraid (AF), angry (AN), disgusted (DI), happy (HAP), neutral (NE), sad (SAD) and surprised (SU).

6.1.4. Integration in a Social Robot

This section describes an application that allowed the robot to recognise and respond to various communicative intentions expressed by the user. This application was created using the results presented in Section 6.1.3.

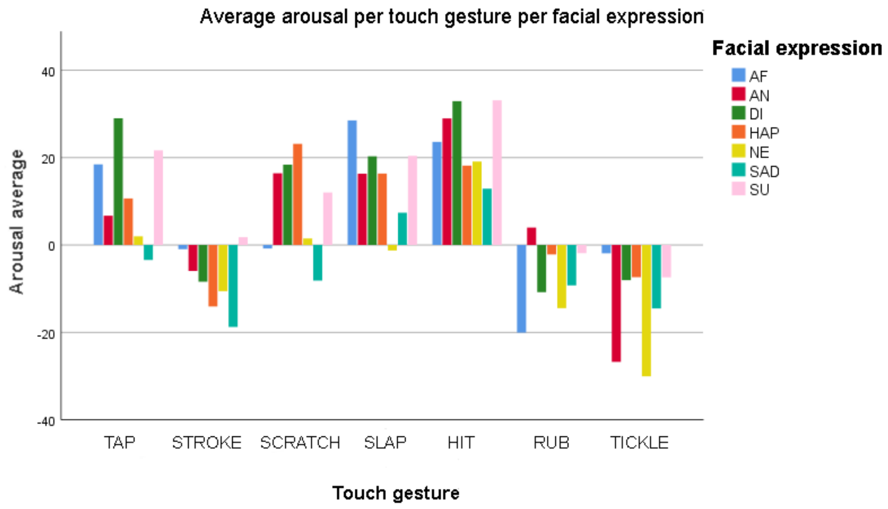


Figure 6.5: Average arousal values (y-axis) as a function of touch gesture (x-axis) and facial expression (color).

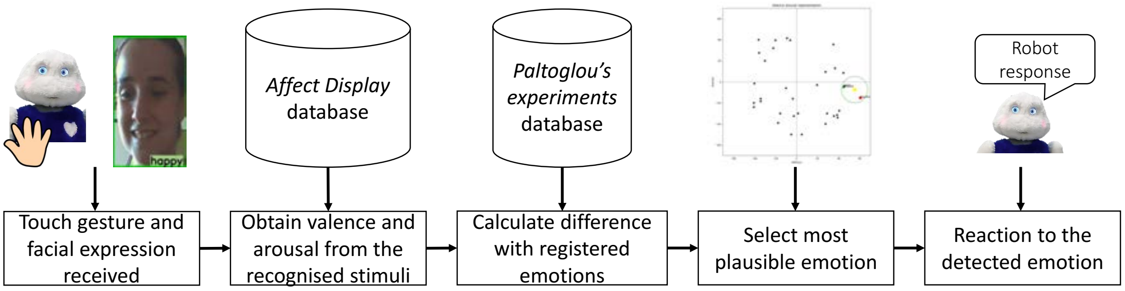

Affect Display database
Paltoglou's experiments database
Robot response
Touch gesture and facial expression received
Obtain valence and arousal from the recognised stimuli
Calculate difference with registered emotions
Select most plausible emotion
Reaction to the detected emotion

Figure 6.6: Flow diagram representing the affect display recognition skill we propose in this work.

The affective recognition application was integrated in the robot Mini. As we mentioned in Section 3.1.3, it was conceived to perform cognitive stimulation and companionship tasks with elderly people. The robot integrates a series of social skills, such as playing different games, storytelling, and making jokes. It can interact with the user by proactively proposing activities based on user preferences, learning from their tastes, and adapting to them. For this development, at a hardware level, Mini included an Intel RealSense®camera and the ATR setup described in Section 3.2.3 and 5.4.1.

6.1.4.1. Design of an application for affect display recognition

For stimuli detection, the robot uses, on the one hand, the ATR for touch gesture detection and, on the other hand, for facial expression recognition, the *emotions-recognition-retail-0003*[30] detector, based on the neural network developed by Intel. Figure 6.6 shows the application flowchart developed to recognise the users' affect display and react accordingly. When the robot detects both stimuli, it attempts to recognise the user's affect display by loading the data from the *Affect Display* database.

We decided to derive the 2-dimensional coordinates (valence and arousal) of the 35 emotions described in Russell's circumplex [131] from the works of Gobron et al. [232] and Paltoglou et al. [233]. Then, we calculated the Euclidean distance between the current valence and arousal values and those obtained in Paltoglou's experiments. Furthermore, we broadened the search area by leveraging detector uncertainty. Based on the results, we adjusted the valence search range based on the confidence of the facial expression detector. On the other hand, the confidence of the touch detector was used to rescale the arousal axis. Figure 6.7 depicts an example of the detector output when attempting to recognise the user's affect display with a tactile gesture 'slap' and a facial expression 'sad' with 75% and 90% confidence, respectively. In black, we can observe the 35 possible emotions from Paltoglou's experiment, and in yellow, the point obtained from our experiments with the perceived stimulus combination. The red dot represents the closest affect display and, thus, the one selected by the robot. The green dot represents the user's potential affect displays. Finally, the green ellipse represents the robot's search area. We use the distance between the yellow point and the closest emotion as the initial radius, and the ellipse's angle corresponds to the angle between the yellow and red dots. Then, we added the detectors' uncertainty, with a weighted Y-axis from the touch detector confidence and an X-axis from the vision detector confidence. Because the touch detector's confidence is lower in the example, the Y-axis is longer than the X-axis.

Finally, the robot will select the perceived emotion and react to it verbally. To filter possible errors of the detector, the robot notifies the user if there are more than five possible emotions within the search ellipse, which is more than 15% of options from which it can select. In this case, the robot informs the user that it does not know the emotion the user is conveying. We recorded a video[31] to demonstrate the social robot recognising the affect display of the user.

[30] Emotion recognition network: https://docs.openvinotoolkit.org/latest/_models_intel_emotions_recognition_retail_0003_description_emotions_recognition_retail_0003.html

[31] Working example video: https://youtu.be/jrv8bY0ssUI

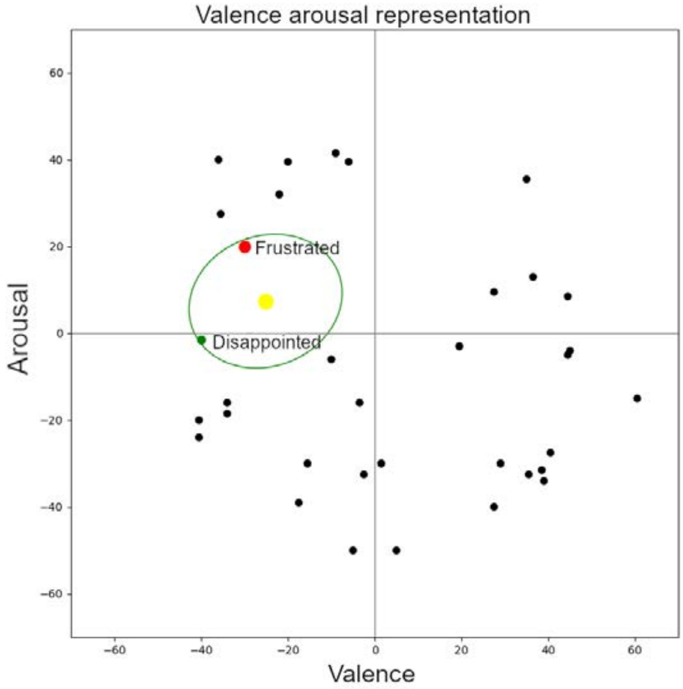


Figure 6.7: Outcome of one of the searches conducted during the robot tests as a result of a combination of a 'slap' and a 'sad' face (yellow dot). The emotion selected in this case within the range (green ellipse) is 'frustrated' (red dot).

6.1.5. Conclusions

In this experiment, we have studied how a combination of visual and tactile stimuli influences people's perceptions of affect display and seeks to apply these findings to a social robot. We experimented with 50 users to determine the perceived valence and arousal when simultaneously exposed to a combination of seven touch gestures and seven facial expressions. The data analysis revealed that the combination of touch and facial expression significantly affects the valence and arousal perceived by users ($p < 0.05$). Specifically, the analysis showed that facial expression had more influence over the perceived valence, while the touch gesture had more impact on the arousal. Based on these results, we developed an application for the robot to determine the user's affect display at any given time. Similarly, if stimuli were not detected reliably, the robot admitted that it did not know how the user felt, resulting in a more natural human-robot interaction.

The work presented has, however, some limitations. The first of these concerns the first part of the work regarding the experiments and the dataset creation. It is relevant to highlight that while the visual stimulus was projected through an interface so that the facial expressions corresponding to each case were the same for all participants, this was not the case for the tactile stimulus. Considering that this stimulus was applied by the person in charge of the experiment, there could be significant differences when applying the same touch gesture to different participants. A tap on one user could have strength or duration different from another tap on the following participant, as there was no mechanism or parameters to replicate the gesture. In addition, it should be noted that the fact that the facial expression is displayed on a screen while the experimenter behind a screen performs the tap may be uncomfortable or unsettling for some users. Choosing the experimental setup for this type of touch experiment is quite complex, a fact that can be appreciated through some works in the literature which use external devices to generate the stimulus. Lastly, since each participant did not experience every combination (20 out of 49 possible combinations), another limitation when interpreting the results is that we did not consider order effects.

In future research, the number of users will be increased to conduct a more generalised study, emphasising the cultural differences between subjects and with a deeper analysis concerning the relationship between each tactile-visual stimuli combination. In terms of multimodality, we could combine our proposal with other stimuli, for example, verbal and non-verbal voice recognition. There can also be improvements in the affect recognition system. We plan to incorporate a machine learning approach based on a regressor to predict the affect display more robustly, thus avoiding relying on the average values from the dataset collected to make the estimation. Lastly, we could generate more complex robot reactions based on recognising the user's affect display and gather additional information in real-time from the recognition system, such as whether the user is comfortable performing specific exercises. For example, we could use these data to anticipate their needs or change how the robot interacts with the user in real-time. The content from this section has been published in the following conference publication:

Publication

Marques-Villarroya, S., Gamboa-Montero, J. J., Jumela-Yedra, C., Castillo, J. C., & Salichs, M. Á. (2022), "Affect Display Recognition through Tactile and Visual Stimuli in a Social Robot", *In International Conference on Social Robotics. Springer, Cham*. **This work was awarded as Best Conference Paper.**

6.2. User Experience in Human-Robot Touch Interaction

In Chapter 2, we considered active touch an essential part of the thesis proposal; touching someone or something without being touched back is impossible. The literature on the affective and interoceptive effects of human-human active social touch (from the toucher's perspective) is lacking. Little is known about what drives and maintains human prosociality. In this sense, Gentsch et al. [110] conducted experiments to test the hypothesis that active stroking on others' skin is more pleasurable than on one's own. We mentioned how one of the motivations for designing the ATR system was to provide a tool to improve human-robot tactile interaction. More specifically, we wanted to focus on active touch in tactile interaction and how this affects social robotics. After all, multiple social robots currently on the market do not have active touch systems and rely on passive systems. Therefore, we ask: could actively touching a robot improve human-robot interaction?

Assessing the role of actively touching a robot in human-robot interaction raises some issues. The first step would be to identify an interaction context in which touch can play an important role. In this regard, the first solution we planned to develop was a learning context or a game in which touch could be implemented. This led us towards the second issue: establishing a baseline against which we could compare an activity that involved the sense of touch to an equivalent activity that did not. For this reason, we chose a game, though we have yet to rule out experimenting with touch in a future learning environment. Therefore, **the study in this section aimed to observe how active contact with a social robot that can perceive, distinguish, and react to touch affects human-robot interaction in the context of a memory game.** For this purpose, we developed a memory game in which the user had to memorise and reproduce a sequence. The sequence incorporated a new element in each round after the user completed it and had to be repeated from the beginning at each turn.

For one of the study conditions, we used a custom external peripheral consisting of buttons with different colours and sounds. For the other condition, which involved the sense of touch, we designed an application that takes advantage of the ATR system's possibilities. This application established the same rules as the version involving buttons. The main difference was that, in this case, the user had to touch the robot's different areas in order and memorise not only the area to be touched but also the type of gesture to be performed on the robot each time. Therefore, through this study, we also had the opportunity to indirectly evaluate the performance of the touch system since, during the game, only the ATR evaluated how the user was touching the robot. We evaluated whether, in the worst-case scenario, it could be an element that worsened the user's experience.

In addition to this, we believed that there was an extra factor in this type of study that was worth exploring, and it had to do with the social robot itself. We thought that the responses and interactions that the robot could provide during the gaming experience might be a distracting element that altered the experience. This concern was supported by works evaluating the social robot's role during user interaction, pointing out that, in some cases, it might be a distraction [234]. Therefore, for our study, we primarily wanted to evaluate whether touch interaction was a differential element that enhanced the interaction with the user. Still, at the same time, we wanted to ensure that the interactivity and expressiveness of the robotic platform itself did not cause this effect.

The last issue to address was how to evaluate the experience itself. According to Tapus et al., evaluating the user experience with a social robot is very important if we want robots to interact socially with humans and therefore, enter in their personal and private dimension [235]. In that sense, we raised the following questions. The first idea is how we could evaluate whether a robot engages a human during a certain task, and, more importantly, whether we have the required metrics to determine whether different robot behaviours can enhance the quality of human-robot interaction. Measuring the user experience [236] involving a social robotic platform implies assessing aspects of the interaction such as the users' feelings, perceptions, expectations and his/her satisfaction. This issue makes the task specially challenging [237, 238]. A characterising feature of the user experience is given by the ability of the robot to engage users in social tasks. As stated by [239] "engagement is a category of user experience characterized by attributes of challenge, positive affect, endurability, aesthetic and sensory appeal, attention, feedback, variety/novelty, interactivity, and perceived user control". We also considered intrinsic motivation [240] a crucial element to measure during human-robot interaction since it helps to determine if the user felt that he/she interacted with the robot for its inherent satisfaction or whether, by contrast, felt that needed to interact for some external outcome. Finally, since the activity involves a game, another relevant parameter to evaluate was whether the user was having fun [241] during the activity. In this section, we will focus on these three variables (engagement, intrinsic motivation and fun) as characterizing features of the quality of the experiences with a social robot in the context of a game.

6.2.1. Background

This section lays the foundations for our multidisciplinary approach in this study. We mix concepts from robotics and computer science with concepts from the analysis of human entertainment, everything in the context of social touch, discussed in Chapter 2. Since both technical and social touch aspects have been discussed in previous chapters, in this section, we focus on the different parameters used to measure both user motivation and the level of entertainment achieved during the experiments. We must highlight the importance of this work in the context of social touch, specifically active touch since after an exhaustive search on the subject, few studies measure the impact on the user interacting by actively touching the robot.

6.2.1.1. Engagement

We used several parameters to measure the user experience during the game. The first feature we considered especially relevant to measure this was user engagement. Defined by O'Brien et al. [239], user engagement is considered as a measure of the quality of user experience defined by the depth of an actor's affective, behavioural, cognitive and temporal investment during Human-Computer Interaction (HCI). The literature agreed that user engagement has affective, cognitive and behavioural elements [242, 243]. With respect to the affective component, users react emotionally to a system through frustration, interest, etc. Also, the relationship between task difficulty and users' skills determines the degree of mental effort required by users to fulfil the task, indicating a cognitive component. And lastly, behavioural engagement refers to the users' actions, such as clicking or querying while using a device. Due to its involvement in user experience and its relationship with the user's behaviour and affect, measuring user engagement could return a reliable measurement of the impact touch interaction has during the experiment [244].

In the literature, several authors have proposed different methods to study user engagement in HRI. Hall et al. [245] used controlled non-verbal cues —like nodding, blinking, and gaze aversion— to investigate how engaged the human participants felt, as indicated by responses to a post-experiment questionnaire. In addition, significant works focusing on engagement during verbal interactions were also proposed by Rich and Sidner. Rich et al. [246] analysed engagement through mutual and directed gaze and correlated it with spoken utterances. On the other hand, Sidner et al. [247, 248], via manual labelling, used gaze signals to distinguish between head nods and quick looks. In another experiment concerning gaze analysis, Ishii et al. [249] combined gaze patterns for conversational agents recorded using eye trackers. Nevertheless, Ivaldi et al. [250] preferred to use used post-experimental questionnaires to assess participants' levels of engagement as well. However, they also measured participants' levels of engagement indirectly using RGB-D data to track the timing of their responses to the robot's stimuli, the rhythm of their interactions, and their directional gaze. Sanghvi et al. [251] preferred to assess engagement automatically from videos of robot interactions using visual cues related to body posture, specifically the inclination of the trunk and back. Similar measures have been used to evaluate behaviours in medical contexts using audio features and video analysis [252–255]. Anzalone et al. [256] proposed a methodology based on metrics that can be quickly retrieved from readily available sensors to assess the engagement elicited during interactions between social robots and human partners. Their metrics were primarily extracted by static and dynamic behavioural analysis of posture and gaze.

We opted for a questionnaire-based approach among these techniques to measure user engagement. This questionnaire will be supplied to the user after the exercise has finished. Among the options available [242, 257], we decided to use the User Engagement Scale (UES) as the primary tool to measure user engagement [243]. The UES was built through an iterative scale development and assessment process that included gathering, refining, and assessing the appropriateness of possible items, pretesting items, and performing two major internet surveys in the e-commerce area. Studies that used the subscales or the entire UES revealed that the questionnaire generally had good reliability and validity. The latest research on this subject has emerged recently, with a shorter version of the initial 31-elements survey [244]. We took advantage of this more concise version of the questionnaire by combining it with other relevant reports that gather information regarding other interesting parameters that report user experience, such as intrinsic motivation or fun.

6.2.1.2. Intrinsic motivation

The following parameter we considered relevant for measuring the user experience during the activity proposed in the experiment was intrinsic motivation. According to Deci et al. [240], intrinsic motivation can be defined as performing an activity for its inherent satisfaction instead of some consequence. When intrinsically motivated, a person is moved to act for the fun or the challenge involved in the activity rather than external artefacts, pressures, or rewards. In contrast, extrinsic motivation is defined as acting to gain some separable outcome. There lies the difference between extrinsic and intrinsic motivation. Given the vast gap between intrinsic and extrinsic motivation, psychologists have attempted to construct hypotheses regarding which characteristics of activities make them intrinsically compelling for some people (but not all) at certain times. For example, the same activity might be intrinsically motivating for a person at a given time but no more later on.

These notions about motivation have been applied and studied regarding social robotics. For example, Saerbeck et al. [258] studied how a socially supportive robot affects students' performance in learning a language. Lee et al. [259] conducted another similar study where children practised language learning with socially assistive robots twice a week for eight weeks. They reported an increase in their speaking skills and a significant enhancement in their motivation. Deublein et al. [260] studied how a robot can increase college students motivation. However, they explicitly tested different versions of motivational behaviours without finding significant differences in the participants' motivation.

The study of motivation also has meaningful connections with the concept of trust. In a recent study, Zorner et al. [261] proposed a scaled-up, immersive, science fiction HRI scenario for intrinsic motivation on human-robot collaboration. They proved that their scenario was an appropriate tool to measure trust in human-robot interaction and the influence of non-verbal communication on human trust in robots.

There are several ways to measure the intrinsic motivation behind performing an activity. Most of these methods have been oriented towards measuring motivation during learning activities. For example, the Academic Motivation Scale (AMS), designed by Vallerand et al. [262], divides intrinsic motivation into different subscales, such as knowledge, accomplishment and stimulation. Alternatively, Deci and Ryan [263] presented motivation as a spectrum from entirely intrinsic to wholly extrinsic. They based this approach on the self-determination theory [262, 263]. The idea behind this theory is that the motivation of humans is linked to the basic psychological needs for autonomy, belonging, and competence. More specifically, this theory predicts that social circumstances that facilitate fulfilling these three demands can preserve or even boost intrinsic motivation while supporting the internalisation and integration of extrinsic motivation. Additionally, the Intrinsic Motivation Inventory (IMI) is connected to the self-determination theory [264]. It is a multidimensional measurement device intended to assess participants' subjective experience related to target activity in laboratory experiments. It has been used in several experiments related to intrinsic motivation and self-regulation [265–267]. Between the two questionnaires mentioned, we discarded the AMS since it leaned more towards learning activities —having more items specifically related to this use case— and instead decided to use the IMI.

6.2.1.3. Fun

Since the experiment's main activity is a game, the last notion we wanted to measure was fun. Even though the concept of fun is frequently emphasised, the concept underlying the phrase and its measurement is not always apparent. The concepts 'fun' and 'enjoyment' are commonly used interchangeably in academic literature. Nonetheless, it has become common for articles describing enjoyment to leave their interpretation of fun vague in recent years. According to Tisza and Markopoulos [268], 'enjoyment' refers to happy emotions, whereas 'fun' refers to a broader, subtle and more challenging to grasp and define. Therefore, fun is not only an experience but is also connected to intrinsic motivation, as it can be a solid factor in encouraging children to try new experiences and challenges. In this aspect, Malone and Lepper [269] emphasised the importance of the intrinsic motivation that could be evoked by the optimal level of curiosity, fantasy and challenge. Moreover, according to Bisson and Luckner [270], fun and play can be a catalyst for eradicating socially limiting elements that are ingrained in us.

Despite the increased interest in quantifying the fun experience, there currently needs to be more reliable measurement techniques. Some of these techniques measure product liking with preliterate youngsters. In the case of adults, the instruments used most extensively assess game enjoyment and engagement, as well as the gaming experience across multiple dimensions. Examples of these tools are the questionnaires used to assess intrinsic motivation and engagement mentioned before [240, 244]. On the one hand, the *Fun Toolkit* is a set of tools that targets the teenage age group in order to measure their preference for products [271]. However, its main drawback is that it handles fun as a unidimensional construct. On the other hand, the FUN scale, proposed by Tasci et al. [272], considers fun to be a multidimensional construct, yet, it has been validated to assess the fun value of a tourist destination as a product among adults, so it could be challenging to adapt to our specific context.

According to Tisza and Markopoulos [268], having multiple dimensions would help to conceptualise and define fun instead of treating the concept as an umbrella term. They introduced an instrument designed to fill this gap by providing a comprehensive questionnaire called *FunQ*. The *FunQ* questionnaire is used to test how a learning activity maps on its different dimensions. However, their authors suggest applying it not only concerning learning but in other activities in which fun can play an important role, such as participation in experimental studies, human-computer interaction, playful activities and experiences. This questionnaire is a theoretically founded instrument that handles fun as a multidimensional construct and focuses on the personal experience while engaged.

6.2.2. Study Implementation

Following the definition of the theoretical background of the study, this section defines the set of elements that comprised the environment in which the study took place. On the one hand, we established the two conditions related to the peripherals, the button system and the game implemented via the touch system, both of which were integrated as two 'skills' of the robot used in this work, Mini (explained in Section 3.1.3). Next, we will discuss the role of the robot's expressiveness and interactivity and the options we propose as additional conditions of the experiment.

6.2.2.1. Button-based game

This game is similar to *Simon Says* [273], an electronic game created in 1978 based on the traditional game, in which one of the participants says an action and the rest must perform it. The original electronic game consisted of four buttons, each of a different colour, which light up randomly while emitting a sound. Once the sequence finishes, the player must reproduce the sequence in the correct order. Therefore, we developed a similar device (see Fig. 6.8) in

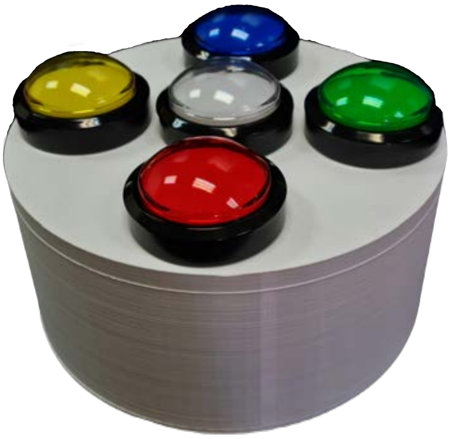


Figure 6.8: Peripheral used for the button-based game.

the laboratory. Through this peripheral, the user must use visual and auditory memory and attention to play the game correctly. The number of available buttons in this game was increased to five, but the instructions were the same as in the electronic game and the buttons also had integrated LEDs. The game consisted of memorising the sequence and performing it after. The length of the sequence determines the difficulty of the game. First, the robot asks the user whether he/she wants to listen to the rules. If the response is affirmative, the robot proceeds to briefly explain the rules using its voice and display them using text on its tablet. Afterwards, the robot has two tasks, to represent the sequence in the button box and to interact with the user. Examples of this interaction are explaining the game's rules if necessary, remembering the score, rewarding the user or motivating him/her to continue playing.

Afterwards, the game goes as follows. First, the robot asks the user if he/she wants to remember the rules or the game's instructions. Once this step is completed, the LEDs belonging to each button start to light up. As mentioned above, the user's primary function is to follow the sequence of lights by pressing the appropriate coloured button. The robot is in charge of displaying the sequence with random colours that the peripheral shows. The game consists of adding a random colour to the same sequence. The robot counts and plays various motivational comments when the user gets the sequence right. If the user presses the wrong button, i.e., the button LED that is lit does not match the button pressed, the robot notifies the end of the game and turns on all the LED lights in the box. Finally, the correct sequence is displayed on the peripheral, and the robot notifies and the user whether he/she surpassed his/her personal best.

6.2.2.2. Tactile game

A new version of the previous game involving the touch system has been specifically designed for this experiment and essentially has the same rules as in the case discussed above. Instead of repeating a sequence of buttons, the player had to repeat a touch sequence that will increment as the player can replicate it. This sequence had to be performed on the robot's surface, specifically three areas of the robot's body. These locations are the arms and the belly of the robot. In addition, the user also had to memorise a touch gesture to take advantage of the possibilities offered by the touch system. The gestures available to the user during the test were: tap, slap, stroke, tickle, rub and scratch. These gestures are the ones that appeared in Section 5.4. The robot generated random gesture/area combinations for each game. Since the robot does not have LEDs in the contact locations mentioned before, in order to provide the combination, it used its voice to indicate the touch gesture, and at the same time, it moved the contact location. This was done instead of using only voice commands in order to shorten the duration of giving the user the complete sequence.

In addition to the differences in sequence content, there were two further differences. The first difference appeared during the rules explanation sequence. In this sense, the operation was initially the same, i.e. the robot explained the game's rules in general. The main difference in this game version was that, as an added element, the robot also explained the different types of touch available to the user. More specifically, the robot did this in two ways to ensure the user understood the meaning of each gesture. Firstly, a definition was displayed in line with the ones presented in Yohanan's dictionary [17]. Once the robot gave the user the gesture's definition verbally and textually using its tablet, it displayed a video demonstration of the gesture using the tablet once again. The videos are intended to prevent possible conflicts between gestures that are a priori similar or whose formal definition alone might not solve ambiguous concepts related to them. For example, the gestures' duration or the strength involved in performing them. This was similarly done in the tests described in Section 5.4.

The last difference was how the robot behaved while the user was performing the sequence of touch gestures. In this case, the robot informed via voice whether the last gesture the user performed was incorrect instead of using noises, as it happened with the button-based peripheral. Since touching the robot is not exactly equivalent to pushing a button, in this version of the game, the robot also had to explicitly inform the user whether the gesture was correct, briefly using its voice. In addition, the game included a threshold in order to evaluate whether the touch gesture was in fact incorrect or a false negative. This feature was based on the touch gesture confidence value provided by the online classifier. If the latter occurred, the robot indicated to the user that it could not understand the gesture and asked the user to repeat it again. Lastly, once the sequence has been completed correctly, as with the button system, the robot

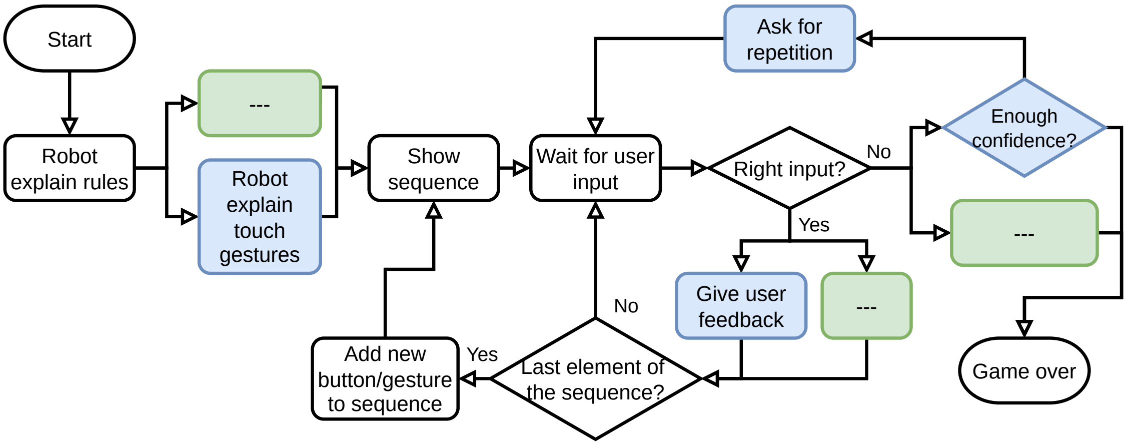


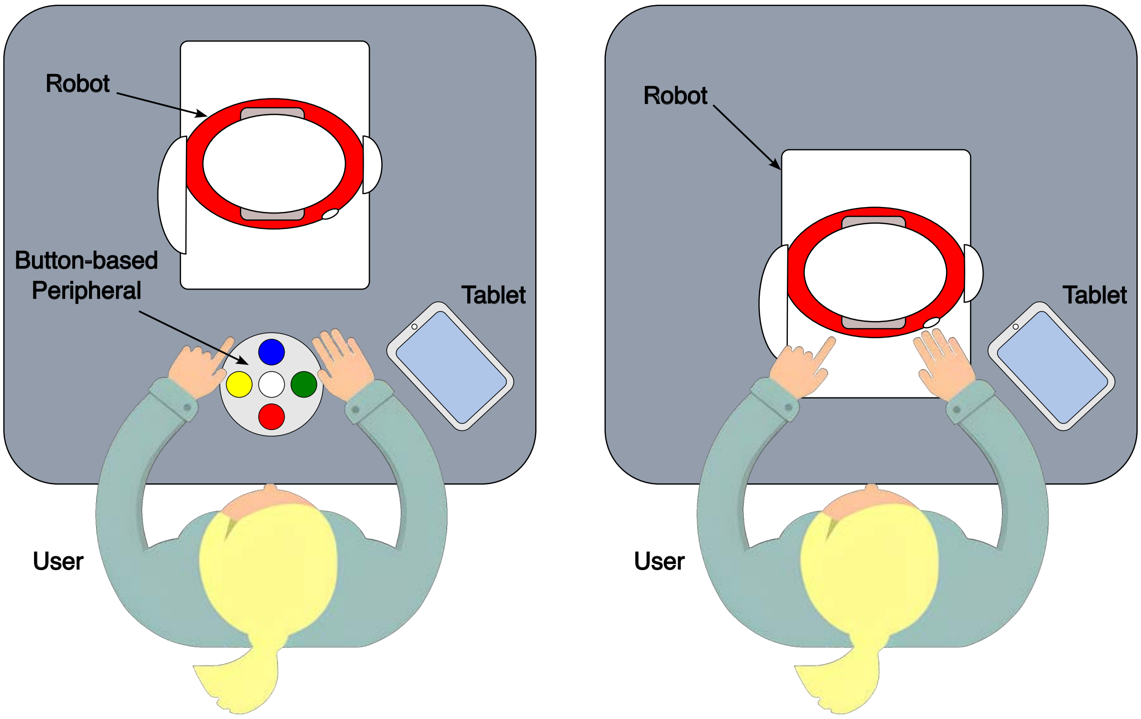
Figure 6.9: Flowchart of the memory game representing the differences between the two peripherals used: buttons (green) and touch gesture recognition (blue).

indicated to the player that it completed the sequence successfully and continued with the next round. This continued until the user makes a mistake, as it happened with the button-based game. In that case, the robot notifies the user, and the game is over. Figure 6.9 shows the complete flowchart of the game. The chart highlights the differences between both iterations using boxes in blue for the touch-based game and green for the button-based one.

6.2.2.3. The social robotic platform

The last element of the study is the robotic platform. We used the Mini robot placed on a table for this experiment. The robot's location varied slightly depending on the peripheral involved in the interaction. In the case of the button device, the robot was positioned just behind it. In the case of using the touch system embedded in the robot, Mini was positioned close to the edge of the table to facilitate the user's physical contact with the robot. Both setups are displayed in Figure 6.10.

As explained in Section 3.1.3, Mini has an expressiveness system that includes LEDs, speakers to emit voice and non-verbal sounds, eye displays to show expressions, and servo motors, which allow it to move its head, arms and body. While the way it displays its expressiveness may be unique because of the combination of devices that produce it, the quality itself is not. One of the main qualities of a social robot is its expressiveness, so it is an element that will be present during human-robot interaction, regardless of the robotic platform used.



(a) Button-based experimental setup.

(b) Touch-based experimental setup.

Figure 6.10: Experimental setups of the user experience experiment.

6.2.3. Methods

We established that, during the game, the robot would play the role of animator-guide rather than that of a rival. This decision was motivated by the game's rules. Therefore, as far as the robot's expressiveness is concerned, this was manifested through encouraging comments in each successful round, joking comments and signs of disappointment during the defeat. In addition, the robot kept track of the user's successful rounds, and finally, when the user lost, it indicated the number of successful rounds and the correct sequence. As indicated before, we theorised that this behaviour by the robot might influence the user during the test. Moreover, we think this effect could have a significant influence when comparing the button system and the touch system. For example, the user might focus on the button system and not on the robot's feedback. Equivalently, in the case of the touch system, expressivity could influence the user experience due to the fact that during the touch-based game, the user is interacting directly with the robot. Therefore, we have decided to treat expressiveness as a factor with two different conditions: expressiveness present, i.e. the natural state of the robot, and minimising expressiveness, giving minimal information to the user to continue the game and be aware of events such as making a mistake and ending the game.

We designed a 2x2 between-subject design user study to test four experimental conditions: the presence or absence of the robot's expressiveness and the choice of peripheral. The peripherals were either the button-based game or the game based on the ATR system. The four resulting conditions are the following:

1. *Buttons* and *No expressiveness* condition (BN): We use the button-based game and minimum expressiveness from the robot interaction. The robot will only conduct the game and give the user essential guidelines to play.
2. *Touch* and *No expressiveness* condition (TN): The same as before, but instead of using the button-based peripheral, the user will play directly with the robot using the touch system.
3. *Buttons* and *Expressiveness* condition (BE): The expressiveness is added to the button-based game.
4. *Touch* and *Expressiveness* condition (TE): We include expressiveness in the button-based game.

The two conditions, the peripheral choice and the addition of expressiveness, were the independent variables of our experiment, while engagement, intrinsic motivation and fun were the dependent variables. Now that the variables are set, we can formulate the following hypotheses:

- **H1a**: The tactile version of the game significantly increases engagement.
- **H1b**: There is no influence on the peripheral condition from the expressiveness condition in terms of engagement.
- **H2a**: The tactile version of the game significantly increases intrinsic motivation.
- **H2b**: There is no influence on the peripheral condition from the expressiveness condition in terms of intrinsic motivation.
- **H3a**: The tactile version of the game significantly increases fun.
- **H3b**: There is no influence on the peripheral condition from the expressiveness condition in terms of fun.

The next step was to create a unified questionnaire to gather all this information. The questionnaire consisted of a combination of the three standard questionnaires presented in Section 6.2.1: UES [244] for engagement, IMI [264] for intrinsic motivation, and FunQ [268] for measuring fun.

6.2.3.1. Designing the questionnaire

Our main priority when designing the unified questionnaire was gathering enough data to ensure the reliability of the responses while minimising assessment time and respondent fatigue. To assess engagement from the user, we used the short form of UES. User Engagement Scale Short Form (UES-SF) has proven to be sufficiently accurate and valid and is frequently used in digital contexts [234, 274, 275]. The 12 elements from the form used a 5-point Likert scale, ranging from 1-'strongly disagree' to 5-'strongly agree'. These items correspond to four different categories, with three items each: *Focused Attention*, *Perceived Usability*, *Aesthetic Appeal* and *Reward Factor*.

Next, we measured intrinsic motivation with IMI. In order to introduce it in our study, according to its authors, first, we had to decide which of the variables (factors) we wanted to use based on what theoretical questions we were addressing. Then, we used the items from those factors, randomly ordered [266]. In our case, we used the *Interest/Enjoyment* and the *Perceived Competence* items. The first subscale is regarded as a self-report measure of intrinsic motivation; consequently, while the complete questionnaire is referred to as the Intrinsic Motivation Inventory, only one subscale examines intrinsic motivation. As a result, this subscale often has more items than the other subscales. The perceived competence concept is a positive predictor of both self-report and behavioural measures of intrinsic motivation. These two concepts, Interest/Enjoyment and Perceived Competence have 13 items in total.

Finally, to assess fun, we used the FunQ questionnaire, with 18 items divided into six dimensions: *Autonomy*, *Challenge*, *Delight*, *Immersion*, *Loss of Social Barriers* and *Stress*. From these dimensions, we decided to remove Stress, Autonomy and Loss of Social Barriers because they had a less significant effect on the 'Experienced Fun' concerning the other categories, according to the authors [268]. Also, they are the categories that showed the questions were out of the context of the experiments. An example of this, for the Loss of Social Barriers, one of the questions is, 'During the activity, I talked to others easier than usual', and the experiment does not involve the participation of multiple subjects simultaneously. Altering the questionnaire by removing dimensions is a procedure seen in another study implementing the FunQ form [241].

Once the elements of each questionnaire were chosen, we merged them into a final version. The first step was to detect common elements. Four pairs of questions with exceptionally similar meanings were detected. For example, one of the FunQ questions was, "During the activity, I had fun" for the Delight dimension, while for the Interest/Enjoyment dimension of IMI it was, "This activity was fun to do". In this case, we decided to leave the FunQ question. Appendix B.1 shows the complete merged questionnaire with the common items highlighted in the same colour. Next, since most of the people that participated in the experiment only understood

Spanish, we followed a set of steps to adapt the questionnaire: (i) each of the items used in the final questionnaire was translated into this language, trying to preserve the original meaning of the question in the translation, (ii) backward translation from Spanish, and finally, (iii) comparison of the original and the backward translated English text, solving the discrepancies. The complete questionnaire in Spanish is shown in Appendix B.2.

Lastly, for each item in the final form, we used a 5-point Likert scale, ranging from 1-'strongly disagree' to 5-'strongly agree'. To score engagement and intrinsic motivation, we computed the average of the scores of each of its items. For the FunQ, however, the resulting score was obtained as the sum of the scores of the items without averaging. We also collected demographic data about age, gender and mood. At the end of the form, we left an open and non-compulsory 'Comments' question, where the participants could give their opinion about the study and the elements involved in the study, such as the peripherals, the system's performance or the expressiveness of the robot.

6.2.3.2. Participants

A total of 83 people volunteered in the study. Regarding gender, 37 participants identified themselves as female and 46 as male (45/55%). In terms of age, 26 belonged to the 18-24 group (31%), 14 were in the 25-34 group (17%), three were in the 35-44 group (4%), 8 participants were in the 45-54 age group (10%), five belonged to the 55-64 age group (6%), and 27 were older adults, with 65 years or more (32%). All the volunteers were assigned in a random manner to the four conditions, resulting in $n_{BN} = 21$, $n_{TN} = 21$, $n_{BE} = 20$ and $n_{TE} = 21$.

6.2.3.3. Procedure

The experiment was carried out as follows. The user entered a room where the robot and the experimenter were located. The experimenter briefly explained to the participant what the experiment consisted of. After this, the participant was asked if he/she had any questions. The participant then took a sit and proceeds to fill out a data protection document, including personal and contact details and the identifier that serves as a pseudonym. Once this document was completed, the experimenter left and the user proceeded to interact with the robot.

The experiment consisted of playing the memory game explained in Section 6.2.2. Although there were different conditions, none fundamentally altered the game's mechanics, which continued until the player made a mistake. The moment the user committed a mistake, he/she had the opportunity to continue playing or could complete the questionnaire. The game lasted between 10 and 20 minutes per game, and an entire session, including filling in the quiz, lasted on average between 25 and 35 minutes.

6.2.4. Results

This section covers the statistical analyses carried out using the SPSS software and the information extracted from the comments that 44 participants of the experiment optatively wanted to express. An alpha significance level of $\alpha = .05$ was used for the statistical analyses.

6.2.4.1. Quantitative Results

The first step before the analysis was to detect and remove all possible outliers the dataset might contain. By analysing the interquartile range for each dependent variable, we removed seven cases from the dataset that SPSS considered outliers. The final dataset comprised 77 samples, 19 from the BN condition, 20 from the BE condition, and 20 and 18 for the TN and TE conditions, respectively.

We planned to perform a 2-way MANOVA with this dataset with the engagement, the intrinsic motivation and the fun. To do this analysis, we had to test whether the assumptions were met. The first assumption implies that the data should be normally distributed. A set of Shapiro-Wilk tests verified that, on the one hand, the fun and the intrinsic motivation met this premise by returning their respective non-significant tests $p = .213$ and $p = .428$. Nonetheless, deviation from normality was significant in the case of the engagement ($p = .021$). The next condition was the correlation between variables. In this case, all possible combinations between the variables showed a significant correlation: engagement-motivation ($r = .711$, $p < .001$), engagement-fun ($r = .692, p < .001$), and fun-motivation ($r = .713, p < .001$). The last assumption implies verifying the homogeneity through Levene's tests. All three variables, engagement ($p = .639$), motivation ($p = .132$) and fun ($p = .219$), showed no significance, thus revealing homogeneity of variances.

The 2-way MANOVA for the intrinsic motivation and the fun showed a significant effect of using the touch system over the buttons on the engagement and the motivation variables at once, with a Wilks' $\Lambda = .917$, $F(2, 72) = 3.247, p = .045$, $\eta_p^2 = .083$. The expressiveness condition, however, showed no combined effect on intrinsic motivation and fun, with a Wilks' $\Lambda = .926, F(2, 72) = 2.871, p = .063, \eta_p^2 = .074$. There was no significant interaction between the peripheral choice and the robot's expressiveness for intrinsic motivation and fun combined with a Wilks' $\Lambda = .975$, $F(2, 72) = .914, p = .405$, $\eta_p^2 = .025$. The subsequent univariate

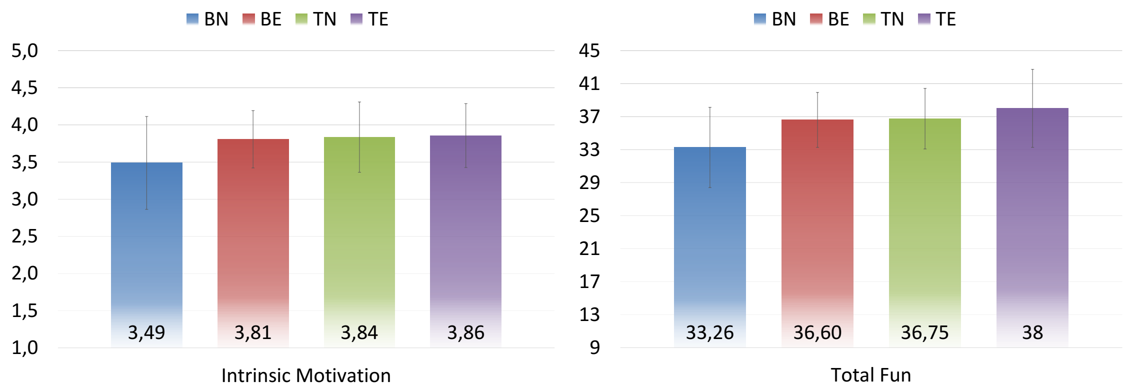


Figure 6.11: Charts with the average values for the intrinsic motivation (left) and the total fun (right). The error bars represent the standard deviations.

ANOVA for the intrinsic motivation revealed a non-significant effect of the touch system over the buttons $F(1, 73) = 3.202, p = .078$ and neither when the expressiveness was introduced $F(1, 73) = 2.346, p = .130$, and also no interaction between these two factors $F(1, 73) = 1.792$, $p = .185$. These results disproved **H2a** but validated **H2b**. With respect to the fun variable, the univariate ANOVA showed a significant effect of the touch system over the buttons $F(1, 73) = 6.582, p = .012, \eta_p^2 = .083$ (validating **H3a**) and the expressiveness $F(1, 73) = 5.799, p = .019$, $\eta_p^2 = .074$, but no interaction between these two factors $F(1, 73) = 20.918, p = .277$, proving **H3b**. Figure 6.11 shows a bar chart comparing the averages of the intrinsic motivation and total fun-dependent variables for each condition.

For the engagement dependent variable, we used two Independent-Samples Mann-Whitney U Test, one for the peripheral and another for the expressiveness conditions. The first test indicated that the touch system group had a significantly higher engagement level than the group that used the buttons with a Mann-Whitney $U = 948.5, p = 0.034$. However, this was not true when comparing the groups based on whether the robot's expressiveness was present, with a Mann-Whitney $U = 875, p = 0.171$. A subsequent two-way ANOVA supported these results, showing a significantly higher level of engagement for the touch system factor $F(1, 73) = 5.098$, $p = .027, \eta_p^2 = 0.065$, but not for the expressiveness factor $F(1, 73) = 2.409, p = .125$. Therefore, the combined results from the U-test and the ANOVA proved **H1a**. The ANOVA did not show an interaction between the peripheral choice and the inclusion of the expressiveness over the level of engagement $F(1, 73) = .154, p = .696$, thus validating **H1b**. A descriptive bar chart comparing the averages of the engagement variable for each condition is shown in Figure 6.12. Finally, Table 6.2 contains all the dependent variable relevant figures for every condition.

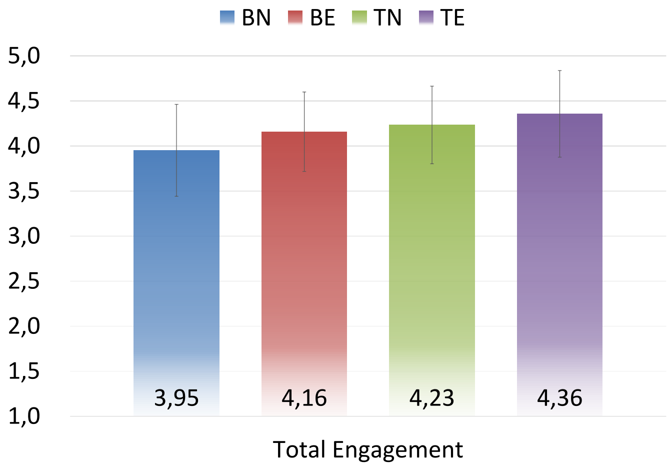


Figure 6.12: Charts with the average values for engagement. The error bars represent the standard deviations.

Table 6.2: Averages and standard deviations of the dependent variables and the number of samples (N) for all the combinations of the two factors: peripheral choice (the buttons peripheral or the touch system) and the presence of the robot's expressiveness. The scales for total engagement and intrinsic motivation are 1-5 and for total fun 9-45.

Dependent Variable	**Peripheral**	**Expressiveness**	$\bar{X}$	σ	**N**
Total Engagement	Buttons	No	3,95	0,510	19
		Yes	4,16	0,441	20
	Touch	No	4,23	0,431	20
		Yes	4,36	0,481	18
Intrinsic Motivation	Buttons	No	33,26	4,863	19
		Yes	36,60	3,331	20
	Touch	No	36,75	3,669	20
		Yes	38,00	4,728	18
Total Fun	Buttons	No	3,49	0,625	19
		Yes	3,81	0,386	20
	Touch	No	3,84	0,473	20
		Yes	3,86	0,430	18

6.2.4.2. Qualitative Results

Of the 83 participants, 44 used the ‘Comments’ section of the questionnaire to provide qualitative feedback on the experiment. The main aspects commented on were related to (i) the application’s usefulness, (ii) the interaction platform (robot and peripherals), (iii) the design of each tested application’s design and (iv) the instructions’ wording.

The touch interaction scenario was mentioned positively by eight of the participants, with an emphasis on human-robot interaction. Comments such as, “It was straightforward and convenient to touch it and listen to the answers. I like and find it interesting to touch the robot. Two pieces of iron, that I can touch and hug, I loved it” (ID = 025). Another comment was, “I found it very nice to touch it. Touching it feels like you get familiar with it; it is going to be easier” (ID = 027). Positive comments on the interaction with the button peripheral were also received. For example, “I had a lot of fun, and it is very nice. I was bored, and you made me have a very nice time” (ID = 001) or “I find it fun to revisit the classic game, Simon, in a slightly different format” (ID = 075).

Some participants mentioned negative aspects of the rules of the touch interaction-based game. These comments relate to the rules’ length and the expression format. Some users propose interactive rules, adding an interaction after each explanation. In the case of the button peripheral, participants did not object to the rules.

We also gathered feedback from the open ‘Comments’ question regarding the design and use of both platforms. For example, some of the users belonging to the control group of the button peripheral commented that the tone of the sounds, the intensity of the light or the speed of the game should be improved. On the other hand, the participants who experimented with the touch control group said that there are sometimes false positives in the recognition. For example, one subject commented, “I had some problems with the recognition of the tickling in the stomach, otherwise quite reliable and correct” (ID = 037).

Finally, it could be observed that users of the control groups containing robot expressiveness (BE and TE) included in some of their comments that the robot was attractive. For example, one volunteer commented, “The eyes are very attractive. I felt very comfortable. Looking at the eyes attracts me. Depending on who the activity is for, it would be interesting. This little guy could visit me more often” (ID = 026). Some participants also suggested making the robot comment more on the subjects’ performance during the game, teasing them.

6.2.5. Discussion

The experiments' results showed a significant increase in two parameters, engagement and fun when users use the touch system to participate in the game. These results aligned with our hypotheses **H1a** and **H3a**, indicating a positive effect on the user experience when the user interacted with the robot physically rather than through the button system. The case of intrinsic motivation is more complex to analyse since, despite the refutation of hypothesis **H2a** in the univariate ANOVA, a significant increase could be seen when the interaction of this variable was analysed together with fun through a MANOVA. This analysis was supported by the fact that both dependent variables were correlated.

Regarding the remaining hypotheses (**H1b**, **H2b** and **H3b**), the results showed that the variations of the other dependent variables were in no case related to the action of expressiveness, another of the hypotheses that we wanted to test through the experiments carried out. In this sense, it is important to highlight how, although it was not one of the hypotheses to be validated in this experiment, the effect of expressiveness was significant in terms of the fun experienced by the user. This observation is relatively new since in other contexts, such as learning, the robot's presence was even a distracting element that impacted negatively the user's experience [234]. We were able to make this observation by introducing more complex measures of fun.

We also obtained relevant information through the open-ended question in which we asked participants for their feedback on the experiment. In these comments, users could express their opinion about interacting through the touch system with the user. Some of the comments were positive, indicating that qualitatively, for them, interacting with the robot directly with games was pleasant and interactive. Some of the feedback also evaluated the performance of the touch system in some cases where there were problems detecting contact. This has allowed us to know that, although the performance in general terms has been good, there is still work to be done in terms of detection in some cases. Besides this, the users also appreciated the button game, although some participants indicated they found it monotonous sometimes.

In conclusion, we have demonstrated through these experiments that the use of the touch system significantly improves the user experience in the context of interaction through a memory game. This effect has been demonstrated independently of the expressiveness and interactivity of the robot. However, these results must be put in context, in the context of a fun game. Nonetheless, this methodology could be extended to other contexts where a social robot plays an important role and where touching the robot could be an element that might enhance the user experience, for example, in a learning environment. In addition, we must emphasise that the tests have been carried out with the system working in real-time, without the need to teleoperate the robot. This fact allows us to demonstrate how the ATR system is a useful tool to be used in scenarios involving the tactile perception of the robot, avoiding the need for active control and supervision by an experimenter.

6.2.5.1. Limitations

We believe that some elements in the work help contextualise it and might also help design future follow-up studies. Although the results showed that the introduction of touch in human-robot interaction was a factor that significantly improved the interaction, the results have to be framed in the context of the memory game. In this sense, some factors could be considered in future work, such as knowing the game mechanics beforehand in cases where the participant is playing the touch game. The Simon game was particularly popular in the '90s and 2000s; therefore, this may cause the participant to compare the touch-based game with the original version subconsciously. However, we tried to minimise this factor by hiding the button system when collecting the data. Preference for memory games may also be a relevant factor and, in some cases, may lead to a more or less positive rating of the game. This factor could also be included in a future questionnaire derived from the fact that it has been used for this experiment. Finally, another factor to consider is that, despite trying to maximise the equivalence between both conditions of the peripheral factor, the games are not equivalent. For example, in the case of the game involving the touch system, memorising two elements, such as the area and the type of gesture, may imply an increase in difficulty that contributes to a more enjoyable gaming experience. However, we consider this fact as one of the advantages of introducing a system such as ours, as it increases the possibilities this type of game can offer for interaction.

In addition to what has been mentioned above, we have not studied the interaction effect of other factors collected through the questionnaire for this experiment. Examples of this are age or gender. Although the latter factor has remained reasonably balanced, in the case of age, there are some groups for which we would need more samples to obtain valid results. We can explore a future experiment focusing on these aspects with a larger number of users across all age groups. Cultural differences can also be explored in such a study since most participants were Spanish or Spanish-speaking for this experiment, which does not allow us to generalise the results. Our study also did not consider the user's previous emotional state, a factor that could be explored further in future studies. Assessing this factor may help us to understand whether, for example, taking part in the experiment may have changed the participant's mood after the experiment. Finally, we can also improve this study by using other types of techniques in addition to assessing user engagement through questionnaires. For instance, some works mentioned in Section 6.2.1 use different vision-based techniques to perceive how engaged the user is through their gaze [252–255] or body posture [251]. This combined with our approach will result in a setup more similar to the one Ivaldi et al. proposed in their work [250].

6.2.6. Conclusions

In this section, we presented a study with 83 participants evaluating the impact of actively touching a robot during a play activity in which touch plays a major role. This activity consisted of memorising and replicating a sequence that becomes incrementally longer and more complex. The main factor of the study was the device used to play the game. This factor had two conditions. On the one hand, the basic condition consisted of a button game based on the popular Simon game. On the other hand, the ATR touch system was used, as the sequence consisted of a set of touch gestures executed in order. For the latter, we designed an application that takes advantage of the ATR system's possibilities. In addition to this factor, we have also decided to evaluate whether the expressiveness of the robot during the game can be a factor that might influence the user experience during the game. We decided to employ a questionnaire to measure the user experience in terms of three parameters: engagement, intrinsic motivation and fun.

The study's results have shown, on the one hand, that interacting directly with the robot significantly improved the user experience in terms of engagement and fun and, to a lesser extent, intrinsic motivation. The study also allowed us to prove that this effect occurs independently of the expressiveness and interactivity displayed by the robot. In addition, this work was a test for the ATR touch system in the context of real-time social touch research, being a valuable tool for this touch-interaction-related experiments. Through the touch system, it was possible to conduct a social touch experiment mainly through the robotic platform without the need to employ mock-ups or teleoperation. Finally, through the integration of the touch system and the creation of the memory game, another advantage arose: we were able to completely replace the button system, allowing us to reduce the number of external devices that the robot requires to provide fun and interactivity to the user.

The results from this study also unveiled possibilities for future work. Firstly, more playful activities could be designed that use the touch system and test other user skills, such as games that test the user's reflexes. From this work, it could also be explored whether direct interaction with the robot through such games can positively affect cognition, extending the study to participants with different levels of cognitive impairment. The study could also be improved by integrating different systems (e.g. vision-based perception systems) to assess whether the user is engaged during these tests or by extending it to assess effects related to the participant's cultural background, age or gender.

6.3. Summary

This chapter focused on applying the designed ATR system. We used it for its true purpose: to enhance research regarding social touch in robotics and to design more sophisticated detectors. The studies centred on active social touch; the first centred on how to use touch contact and vision to design an application based on affect communication, and the second one on how active touching a social robot can affect a user's behaviour.

In the first experiment, we studied how a combination of visual and tactile stimuli influences people's perceptions of affect display. We applied the findings to make a social robot capable of affect recognition. Firstly we experimented with 50 participants to determine the perceived valence and arousal when simultaneously exposed to seven touch gestures and seven facial expressions. The data analysis revealed that touch and facial expression significantly influence how users perceive valence and arousal. In particular, the analysis revealed that facial expression had a more significant effect on perceived valence than touch gesture did on arousal. Based on these findings, we developed an application for the robot to determine the user's affective display at any given time.

In the second study, we evaluated the impact of actively touching a robot during a human-robot interaction game involving memorising and replicating a sequence. The game has two main factors. The first factor is the peripheral employed during the game: the basic condition is a button game, and the second condition of this factor uses the ATR touch system. For the second factor, we decided to evaluate whether the expressiveness of the robot during the game affects user experience. We used a questionnaire to measure engagement, intrinsic motivation, and fun. According to the study, in which 83 volunteers participated, interaction with the robot improves user engagement, fun, and intrinsic motivation. The study showed that this effect is independent of the robot's expressiveness and interactivity. This work also allowed us to evaluate the ATR touch system in real-time social touch research.

Chapter 7

Federated Learning in Human-Robot Touch Interaction

Up to this point, we have proposed a system to detect when, where, and how physical contact between a human and a robot occurs. As an example, we can define that at some moment (*when*), someone has performed a tap gesture (*how*) on the robot's left shoulder (*where*). However, the proposal still has an important limitation: the system learns only once using examples of previous interactions with a single robot. In this sense, large-scale approaches like distributed learning paradigms help to alleviate this problem by incrementing the number of robots that participate in the knowledge-gathering process.

To achieve this large-scale and distributed learning, each robot must share the knowledge it acquires through its local interactions with the other robots. Thus, all robots benefit from the newly acquired examples. A virtual keyboard on a smartphone or the speech recognition technology itself can be considered an example. Each new voice command or keystroke is relayed to a server that optimizes future predictions (predictive text) or speech recognition. This central server is refining an Natural Language Processing (NLP) model to deliver to all users. Therefore, the well-known significant network effects are achieved: the greater the number of users in the system, the better the system will function for each individual user. This present-day example is criticised by some users who do not want their personal data to be used on central servers for often unknown purposes [276]. Precisely from this criticism, approaches like Federated Learning (FL) [277] and new laws like the European General Data Protection Regulation (GDPR)[32] have recently emerged.

Federated learning allows the system to continue to benefit from the advantages of network effects when it comes to distributed learning without exposing —or sending over the network— any of the user's information. How is it possible to achieve this? The key is the following: the client nodes and the server do not have nor do they receive all the training samples; instead, they share the trained models. These learned models contain meta-information that has no connection with a particular user. In the case of learning using artificial neural networks, the weights and biases of the different nodes of the network are transmitted. In this chapter, we propose integrating this paradigm into our use case, and therefore **improving our system with a distributed and scalable learning approach that can learn collaboratively and incrementally while respecting the privacy of the user's information.**

This chapter is structured as follows. First, Section 7.1 reviews the literature related to the federated learning paradigm. Afterwards, Section 7.2 describes the design and the implementation of the main contribution of this chapter – the federated learning module. The fourth section, Section 7.3, describes the experimental part of this chapter: the set of gestures selected, how the dataset was created, and the different metaparameters of the system. Section 7.4 presents the experimental results obtained from the baseline approach compared with those obtained from the federated proposal. Finally, Section 7.5 analyses these results and explores the system's limitations and future research paths.

7.1. Background

Federated learning could not be defined without the concept of distributed machine learning, its predecessor. Multi-node machine learning methods and systems that are intended to enhance performance, increase accuracy, and scale to more significant input data quantities are referred to as distributed machine learning [278]. For many algorithms, increasing the amount of input data can considerably minimize the learning error and is frequently more efficient than employing more complicated techniques. Over the years, there have been significant advances related to distributed learning. Verbraeken et al. [279] made a meta-analysis of everything related to distributed machine learning. In this survey, the authors highlighted the importance of data privacy. Furthermore, the work mentions that federated learning systems can be deployed so that the different edge devices that form the system can learn together while preserving the confidentiality of the local proprietary data.

[32] https://eur-lex.europa.eu/eli/reg/2016/679/oj

In 2016, Brendan McMahan and Jakub Konečný coined the concept of federated learning [277, 280]. These authors applied their work to smartphones and focused mainly on improving NLP systems to improve speech recognition and predictive keyboard systems. According to these early works, FL is defined as a machine learning environment in which the goal is to train a high-quality centralised model while the training data remains distributed over a large number of nodes, which each have unreliable and relatively slow network connections. In this way, each node independently computes and updates its local model based on its local data, and communicates this update to a central server, where node-side updates are aggregated to compute a new global model.

In 2019, the concept of federated learning expanded to include all decentralized collaborative machine learning techniques that preserve privacy, resulting in two variations of the original concept [281]:

- *Horizontal federated learning:* Also named Homogeneous Federated Learning. In this paradigm, rows of data are available with a consistent set of features. To be more precise, this would be the type of data fed into a supervised machine learning task, where each row may be implicitly or explicitly associated with a context.
- *Vertical federated learning:* Also referred to as Heterogeneous Federated Learning. In this case, data is vertically partitioned (partitioned by features instead of examples) and the resulting models are shared among different companies and organisations.

The horizontal approach is the one that fits better to our use case, since each federated node is going to contain complete instances per case. Furthermore, horizontal federated learning allows the user to continue to benefit from the advantages of network effects inherently associated with distributed learning without exposing any of the user's data. To achieve this, instead of receiving the client nodes and the server all the training samples, they share the trained models. These learned models contain meta-information that has no connection with a particular user. In the case of learning using artificial neural networks, the weights and biases of the different nodes of the network are transmitted.

In federated learning, $F_1, ..., F_n$ represent data owners, all of whom wish to train a machine learning model by consolidating their respective data $D_1, ..., D_n$. A conventional method would try to gather all the data together, using $D = D_1 \bigcup ... \bigcup D_n$ to train a global model, M_{sum}. In contrast, a federated learning system is a machine learning scheme where the data owners collaboratively train a model, M_{fed}, without exposing their data D_i to the others. Furthermore, the precision of M_{fed}, indicated as V_{fed}, should be very close to the performance of M_{sum}, denoted

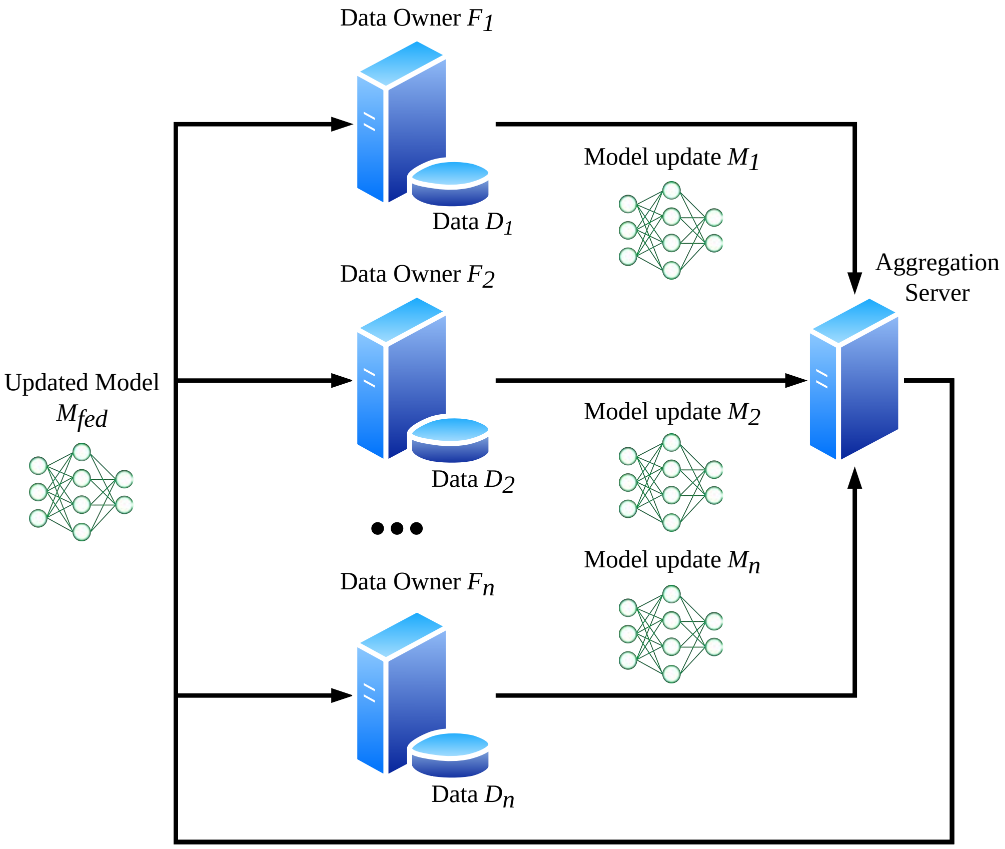


Figure 7.1: Federated Learning scheme.

as V_{sum}. Formally, let E be a nonnegative real number; if $|V_{fed} - V_{sum}| < E$, we say that the federated learning algorithm has E-accuracy loss. The described scheme is shown in Fig. 7.1.

If the reader has a certain degree of knowledge or familiarity with machine learning techniques, it may be interesting to know that by simply transmitting these local models[33], a global model can be refined. Furthermore, if this is true, how could this merging of local models into a perfected global model be done? there are currently several approaches to performing this fusion of models. In the case of using artificial neural networks at each node, the node averages the weights present in its network with the new weights that were received. That is, we transfer weights, not instances. However, in the case of using other Machine Learning (ML) algorithms, it is not always possible to use federated learning techniques. Non-parametric models, in general, can be problematic since their configurations often heavily depend on the exact data that was used to train them.

[33] The particular model that is transmitted depends on the specific machine learning technique used in the federated model. We use artificial neural networks, and the transmitted model consists of the actual connection structure between the neurones and the weights associated with each of the connections.

With respect to the fields where federated learning is currently being implemented, Li et al. [282] recently published a review of current federated learning applications that describes: a) applications for mobile devices, including those mentioned to improve natural language processing, Internet of Things (IoT) use cases, as well as self-driven cars; b) industrial engineering, such as visual inspection or to detect credit card fraud efficiently; c) healthcare, in systems such as disease prediction. Also, in the same year, Aledhari et al. [283] presented a meta-analysis summarising the different variants of FL implementation as well as the fields where they are currently being used. In this sense, it should be mentioned that most machine learning techniques that currently use this technique are based on what is known as deep learning [284], which is usually based on deep/convolutional artificial neural networks, although it can be applied to other ML techniques such as multilayer perceptron-based neural networks. This author mentions the following applications: predictive text on smartphones (GBoard); ranking browser history suggestions; visual object detection; patient clustering to predict mortality and hospital stay time; drug discovery; Functional Magnetic Resonance Imaging (fMRI) analysis (fMRI data are related to different kinds of neurological disease or disorders); brain tumour segmentation; and distributed medical databases. In these surveys —due to the relatively short period of time since this technique was first conceived— there are still no use cases in our field of research, social robotics, and more particularly, there are no use cases related to human-robot tactile interaction.

7.2. System Design and Implementation

This section's objective is to design and implement a system based on the federated learning paradigm, allowing knowledge to be shared among several nodes, in this case, social robots, without compromising the privacy of the robot user's data. The use case is as follows: several users have to interact with different robots, but their interaction is limited to a particular platform. In this case, the simulated scenario is intended to resemble a nursing home where multiple users can interact with the same robot, but the platform cannot share information directly with robots present in other nursing homes or with a server in the cloud.

Since tactile interaction with a robot occurs sporadically, it cannot be ensured that when the aggregation server requires the nodes to be updated, they will have sufficient information to improve the system optimally. Therefore, it would be impractical to force the nodes to train simultaneously. For these reasons, the system's design involves asynchronous elements, as clients can request the model on demand without relying on the server to summon them. This last feature has relevance since the asynchronous relationship between the clients and the server, events such as uneven learning samples and different learning progress are some of the challenges federated learning has to face at the moment [285, 286].

As the work from Kolod et al. shows [287], the communications framework is another significant element in the FL diagram. Their work is a comparison between multiple open-source FL frameworks. For the proposal presented in this chapter, the various open-source options mentioned by Kolod et al. were considered, but most of them were discarded due to their extreme dependence on deep learning libraries such as *TensorFlow* [288] or *Keras* [289] and their orientation towards large-scale deployments. However, due to the fact that, in our use case, the amount of clients is not abundant, we think using algorithms such as deep neural networks might introduce unnecessary overhead. It is also possible that the system will not be able to converge to a solution due to the scarcity of the samples. For these reasons, instead of using a deep learning-oriented FL framework and taking into account that our system already integrates this tool, we prefer to design the federated network using ROS as the communications infrastructure and the *scikit-learn* library to integrate the required machine learning tools.

Currently, ROS does not have a library capable of combining its communications middleware with the infrastructure of a federated approach. Therefore, our system implements such an infrastructure from scratch by taking advantage of ROS tools. The base communication protocol in ROS is the publish/subscribe model through 'topics'. But, its many-to-many one-way transport is not appropriate for Remote Procedure Call (RPC) request/reply interactions, like those required in a distributed system such as a federated one. Therefore, the request/reply will be made via ROS *Services*, defined by a pair of messages: one for the request and one for the reply. A ROS node acting as the server offers a service under a string name, and a client calls the service by sending the request message and awaiting the response. Despite the advantages, ROS-developed systems tend to have limitations when the nodes in the system are numerous and belong to different network domains. Therefore, a large deployment with multiple robots placed across domains might require a much more refined ROS network configuration, for example, to stablish communication through a Virtual Private Network (VPN) [290].

According to Wang et al. [291], a federated system should be based on two different groups of components: first, the agents (also called nodes), and second, the information from the model. More specifically, the agents are defined as the different components of the system that exchange information about the model. Following the architecture presented in FigureThe scenario proposed in this work distinguishes two kinds of agents:

- *Clients.* They train their machine learning models locally and send their parameters to the server to update the global model. More specifically, the clients in our system contain an Artificial Neural Network (ANN) as the estimator of choice for the model. Clients are represented by several social robots with different users assigned to them. These users will touch the robot in various ways to expand the dataset from which the federated system will take the training instances[34].

- *Server.* The server is in charge of building the global model by adding the parameters of the local models and sending them back to the clients. It consists of a central computer.

After the nodes, the next crucial component of the FL system is the information that is transmitted between the agents. More specifically, this information could be divided into two different groups:

- *Client-server information.* It is composed of the metaparameters from the client's local model after it has been trained. More specifically, it consists of a ROS Service containing two matrices. The first one is a 3×3 matrix containing the weights (or coefficients) of the ANN, while the second one contains a 2×2 matrix with the biases. Additionally, an extra parameter has been introduced called 'subject'. This parameter allows the client to asynchronously request the current version of the system without sending its local model.

- *Server-client information.* The server aggregates the weights to the global model, and then sends this updated model back to the client that sent its model. The server sends the information in the same format as it was received, as two matrices, with both the weights and the biases of an ANN.

The FL process can be split into seven phases. Figure 7.2 shows the schematic of these phases, and they are described in detail below.

1. Each client starts by creating a generic, untrained model based on the same metaparameters. This is done by using the same random seed in the initialisation.

2. Each client asynchronously gathers a certain amount of instances representing touch gestures by a series of unique users, as a result of successive tactile interactions. In this work, and for testing purposes, the system will save the instances (in a stratified fashion) in two subsets: training and testing, respectively.

3. The client then sends the trained local model to the server. Since the system is expected to work asynchronously, the client decides when to upload its model. At this point, we also

[34] Note that in this work the words 'client' and 'robot' are used interchangeably.

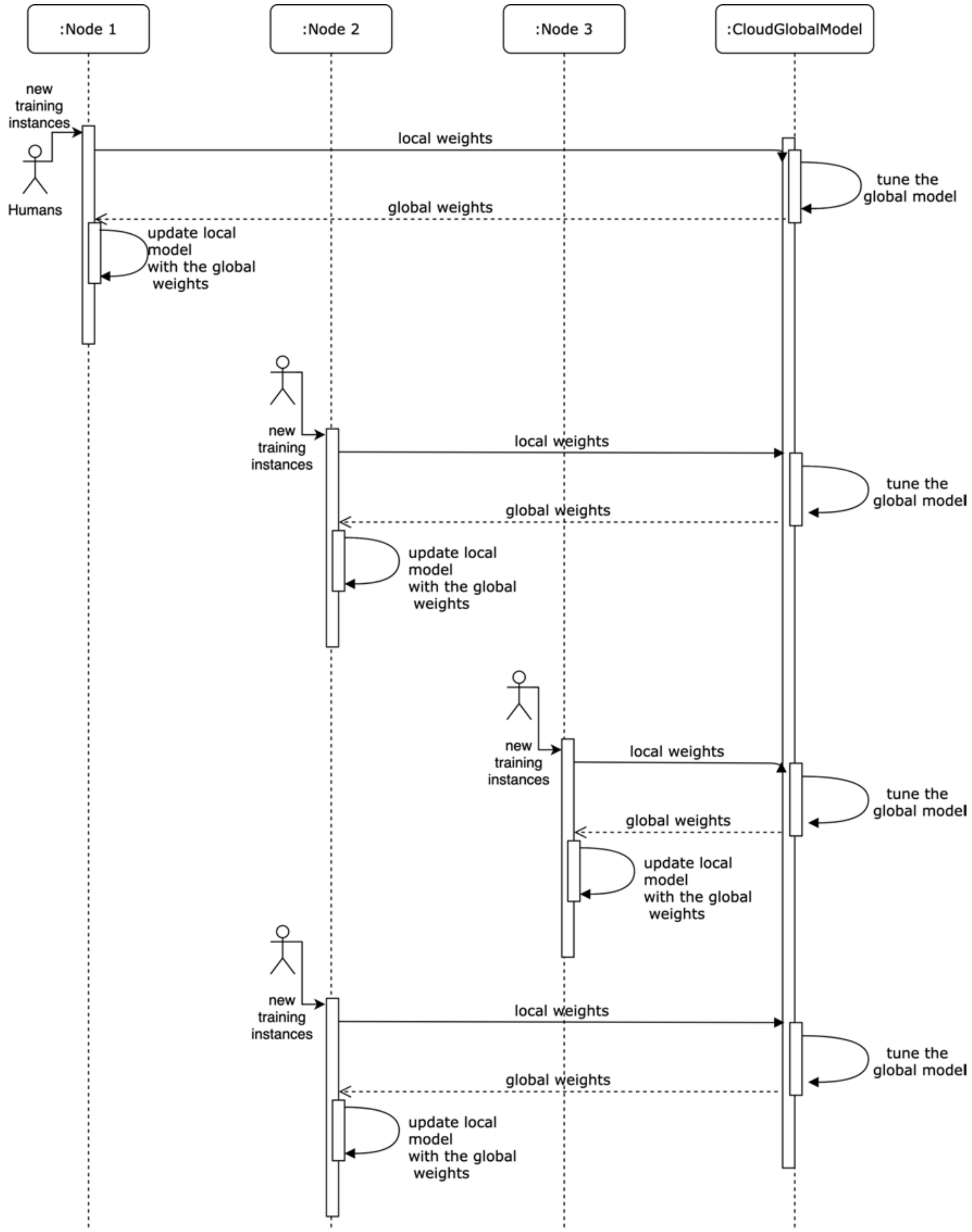


Figure 7.2: Phases of the proposed federated system

studied how the system would behave if the local model is updated with the one on the server before training the model with the last version of the local dataset. This is relevant since, as we mentioned earlier, the clients will upload their models asynchronously, and after some time without contacting the server, a client's local model might be outdated.

4. For the nodes to be encouraged to share with the rest the information that their models have learned (the weights of their artificial neural networks), it is necessary to determine under which criteria this sending of weights takes place. Specifically, each node shares its weights with the rest under any of these conditions: *i)* the first time the node connects to the Internet or when the connection was lost and is reconnected later; *ii)* when the user explicitly indicates it; *iii)* after a certain number of new touch instances.

5. When the server receives the client model, it aggregates the parameters of its current model and the client model. In this case, the aggregation algorithm will be a variation of the *FedAvg* algorithm, with an added averaging weight correction. The weights in the aggregation process between the server model and the client model will vary with each interaction between the agents. More specifically, we consider two weighted averaging models based on an exponential function: the first one will consist of an exponential decay function, and the second one will be an exponential growth function.

6. The server returns the model to the client that requested permission to upload its model. Then, the client will test its model using the previously mentioned test data, and it will calculate the current accuracy/error.

7. The process of sending and receiving the model between the server and the client will continue indefinitely, with each client retraining with its incremental local dataset as its corresponding ATR system gathers more training instances. In this case, in order to carry out the tests, the number of instances has to be finite, so the system will continue until all the information is consumed in the training process.

As it is mentioned in Phase 4-i, the system should be able to handle client node disconnection, a frequent event in a federated architecture. For that, ROS provides tools to check the connection between the client and the server before the client uploads its model. In addition, we have also improved the system by using the Python library *socket*[35], which allows the client to check if it has access to the Internet.

[35] *socket* library webpage: https://docs.python.org/3/library/socket.html

7.3. Method

This section covers the experimental part of the case study, including the data collection procedure, the dataset creation process, and the various system metaparameters and their values for the experiments.

7.3.1. Experimental setup

After defining the elements that compose the proposal, we set up an experimental environment designed to test the capabilities of the federated system. This experiment presented a small-scale system designed to operate with a few nodes that generate instances sporadically. For this experiment, we employed the dataset of 3280 touch gesture instances from 28 users described in Section 5.4. The original dataset was split to generate seven sub-datasets containing instances of four users each. Prior to that, the original dataset was simplified, passing from the original seven gestures to the four ones shown in Figure 7.3: *tap*, *slap*, *tickle* and *stroke*. This simplification was intended to enable a comparison between a previous experiment, the one presented in Section 5.2, and the distributed learning.

7.3.2. Meta-parameters

We can define three different groups for the metaparameters of the proposed federated system, coinciding with three different abstraction levels. These levels are the neural network hyperparameters, client parameters, and federated server parameters.

7.3.2.1. Artificial neural network-related parameters

This is the lowest level of abstraction. To avoid increasing the complexity of the experiment, we decided to modify two hyperparameters of the multilayer perceptron present in scikit-learn[36]. Following prior experiments regarding federated systems [277, 287], we found that the minibatch size, number of epochs, and learning rate tend to have the most impact in these environments.

At first, preliminary testing with the datasets showed that the **minibatch size** B had the greatest impact on the system at a global level. This outcome seemed to be coherent with how the system is designed since each communication round is defined by the number of instances

[36] It is a particular case of the implementation of an artificial neural network.

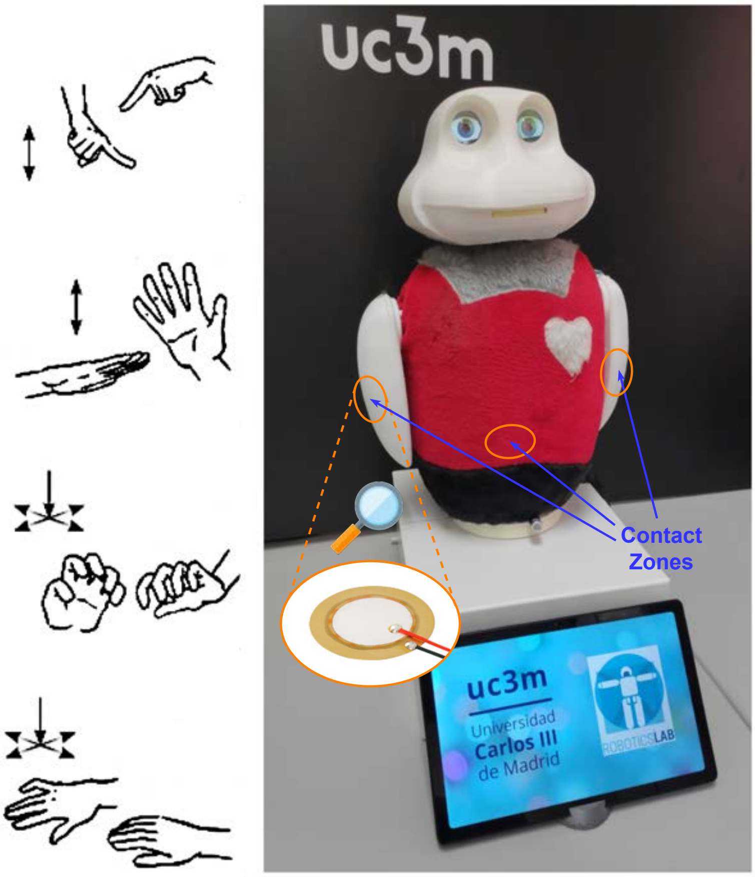


Figure 7.3: Main elements of the experimental setup of the federated system. The touch gestures representation is on the left side, and the Mini robot with the sensing zones containing the piezoelectric microphones is on the right. The touch gesture images are extracted and adapted from the work of [54].

invested in the round, a number that is intimately related to the number of instances that the perceptron uses per epoch to train. Following the literature, we selected 25, 50, and 75 instances per batch for the tests. We also performed testing by modifying the **number of epochs** E. In this case, the testing was less intensive and performed upon the best values obtained from changing the minibatch size. The values selected are 250, 500, 750, and 100 epochs per minibatch. The rest of the hyperparameters of the neural network retained their default values. For these experiments, we highlight the **learning rate mode**, which remained *constant*; the **learning rate initialisation value** η, which remained 0.001, and the **warm start** parameter, which was changed to *True* in this case, since the network has to be trained each communication round. Finally, the **activation function** remained a rectified linear unit function ($ReLU$), and the **solver** was *adam*.

7.3.2.2. Client-related parameters

The next level corresponds to the federated client. In this instance, the main parameter that we will modify during the test will be the **number of instances per round (per client)** n_{tk}. This parameter has an intimate relationship with the more common *number of communication rounds* (T), as shown in Eq. 7.1.

$$T = \frac{n_k}{n_{tk}}, \tag{7.1}$$

where T is the total number of communication rounds, n_k is the total number of instances a client has, and n_{tk} is the number of instances per round per client. We decided to use n_{tk} instead of T since it might help to express more clearly how a federated system would be triggered in an asynchronous environment. Despite this, as Eq. 7.1 shows, the terms are equivalent.

To set the ranges for n_{tk} in the test phase, we opted to find a balance between n_{tk}, n_k, and B. After some preliminary testing, the best results were obtained when these three metaparameters followed the rule shown in Eq. 7.2:

$$B \leq n_{tk} < \frac{n_k}{2}. \tag{7.2}$$

7.3.2.3. Server-related parameters

One of the main characteristics of FL is aggregating each client's model into a single model, so the aggregation function should not decrease the model's accuracy. When calculating a regular average, each data point has equal weight, and thus they contribute equally to the final value. Weighted averages, on the other hand, weight each data point differently. Normally, the aggregation of the parameters in the server is based on the amount of data in every node. In this study, as we explained in the previous subsection, the number of instances n_{tk} will be a 'trigger' for the client to upload its model, despite the more commonly seen number of communication rounds T. This is because of the asynchronous design of the system: the clients will not be expected to upload their models simultaneously, coordinated by the server. For this reason, the server can't know the amount of data a single client has compared to the others when this client is demanding an update. Thus, designing an aggregation scheme depending on the amount of data each client contributed to the last communication round is impossible. Therefore, we decided to make the weights fluctuate depending on the total number of interactions between the server and the clients.

The next step is to model the function and define the requirements that the weight fluctuation should meet. The first decision to make on this matter was if the weight of the local model concerning the model present in the server should decrease or increase over time. We decided to test both a weight growth function and a weight decay function. The next step was to decide on the shape of the function. In this case, we opted for a non-linear function to make the aggregation more impactful in the early stages of the training. Lastly, the bounds of the function are between 0 and 1, representing the upper and lower limits of a weighted average. The relation between the server's model parameters and the client's uploaded model is shown in Eq. 7.3.

$$\omega = \omega_0 \cdot (1 - \tau) + \omega^{'} \cdot \tau, \tag{7.3}$$

where ω represents the server's model parameters after the aggregation, $\omega^{'}$ represents the parameters of the client's model, ω_0 represents the server's model parameters before the aggregation, and τ is the output of the function modelled according to the constraints mentioned before:

$$\tau = b \cdot e^{-\lambda_d \cdot x}. \tag{7.4}$$

Equation 7.4 shows the first function that meets the requirements listed before. In this case, the function decays depending on the number of requests sent to the server. b is the bias or initial value after the server receives a client's model parameters (the first time, the server will adopt the weights of the client directly), and λ_d is the decay rate of the function. Equivalent to the decay function is the growth function shown in Eq. 7.5, which is designed to be symmetric to the former function concerning the x axis.

$$\tau = 1 - (1 - b) \cdot e^{-\lambda_g \cdot x}. \tag{7.5}$$

Figure 7.4 shows both functions for bias values of 0.25, 0.5, and 0.75, and a λ rate of 0.025. For the experiments we used 0.25, 0.5, and 0.75 for the bias, and for the rate, we used values from 0.01 to 0.05. These ranges of values were estimated heuristically. From now on, we will refer to the bias b as the **federated bias** and to λ (both λ_d and λ_g) as the **federated rate**, and whether the function grows or decays (thus specifying whether we are using λ_d or λ_g) will be considered the **federated mode**.

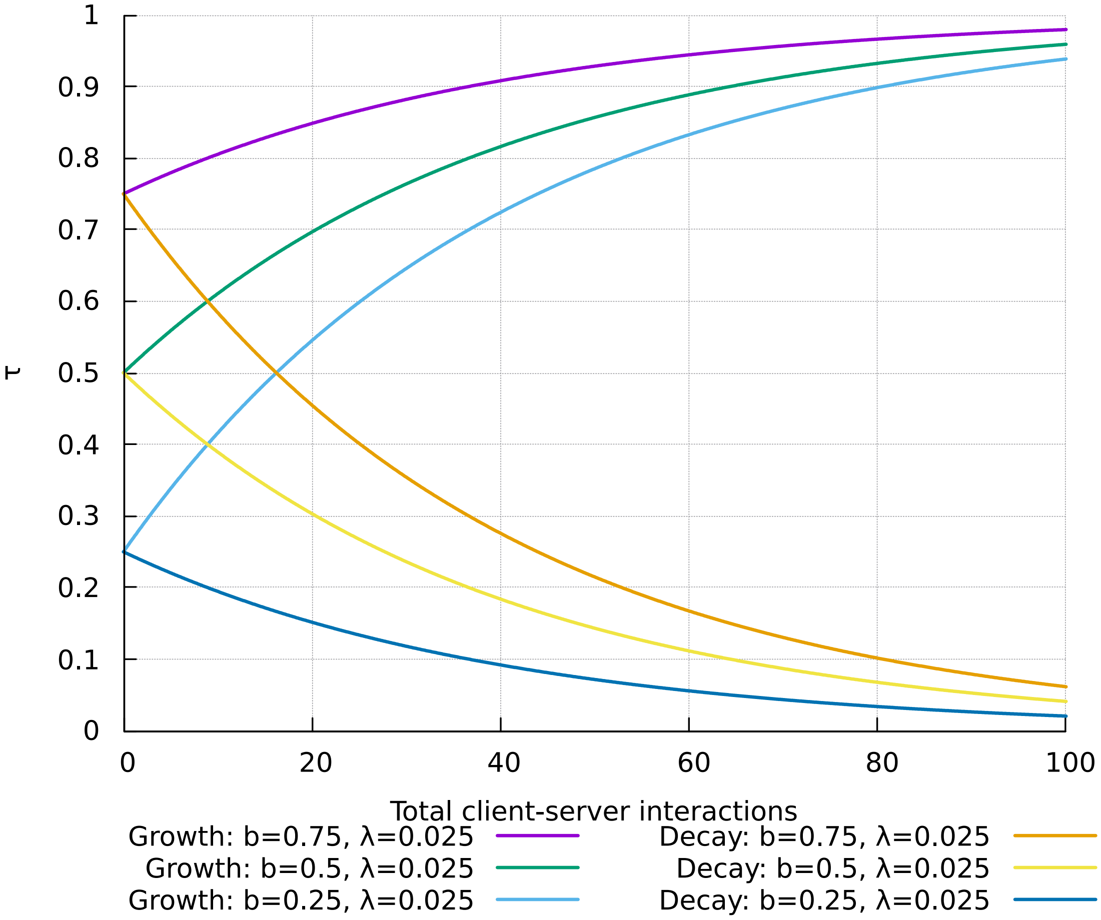


Figure 7.4: Growth and decay exponential functions for the weight correction in the aggregation phase

7.4. Results

The system will be tested and evaluated with respect to two baseline scenarios. The first scenario will use a distributed model on the server to test against the global dataset, which is a more conventional version of the system. The second reference scenario is more similar to the federated system. Still, in this case, the main difference will be that each client will train in isolation with their local datasets without sharing their models or their data. Therefore, we replicate the same incremental training described in Section 7.2, but, in contrast to the federated phases, without sharing information. This way, we can determine if incremental training provides an advantage. Lastly, we will assess the performance of the federated system itself. After setting the metaparameters as described before, the system will follow the steps explained in Section 7.2, training incrementally in each communication round. For this evaluation, we need to clarify that the clients will train successively in order.

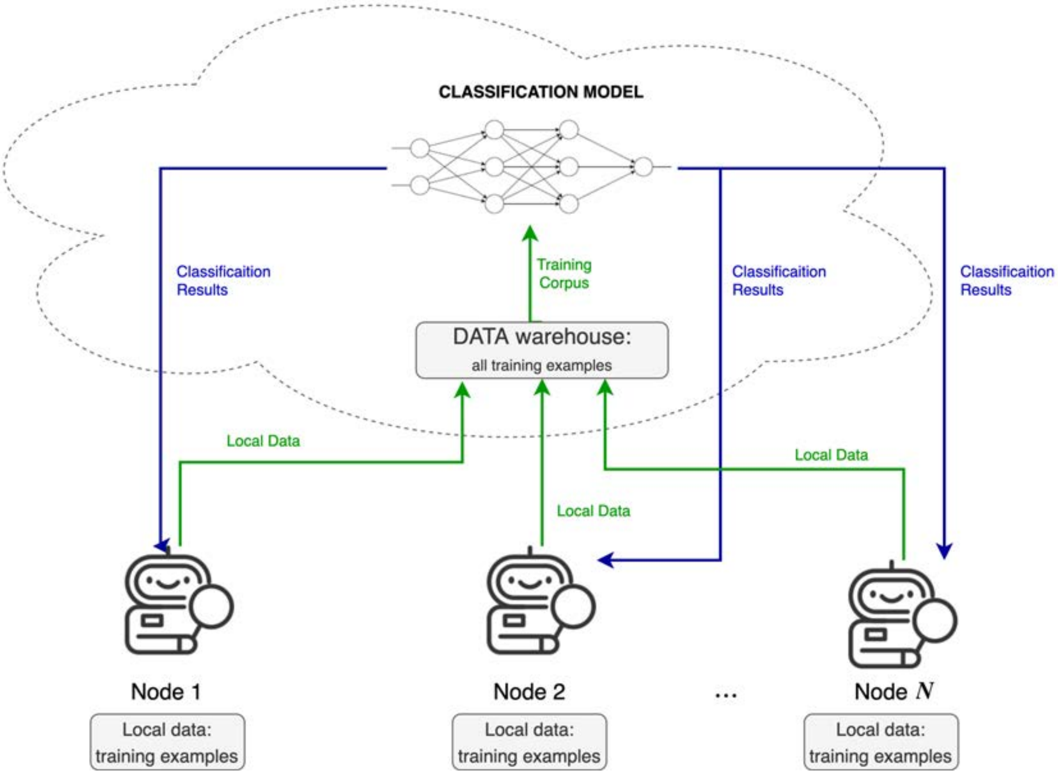


Figure 7.5: Distributed system

7.4.1. Distributed system

First, we present the distributed solution, or the conventional approach, which assumes that instances are shared directly between nodes to build a global dataset to train a classifier model, in this case, a neural network. The pipeline of this architecture is shown in Figure 7.5.

The proposed ANN model and the dataset gathered for this experiment (described in Section 7.2), divided into two subsets, 80% training and 20% testing, achieved an F-score of 0.747 on the test set. In order to test if the incremental features of the federated system might pose an advantage by themselves, we performed a similar experiment (with a distributed system), but in this case, with incremental training for 200 and 400 instances. The F-score achieved with this strategy showed no significant improvements. The best results (tuning the neural network) were 0.718 for 200 instances per round and 0.722 for 400 instances per round.

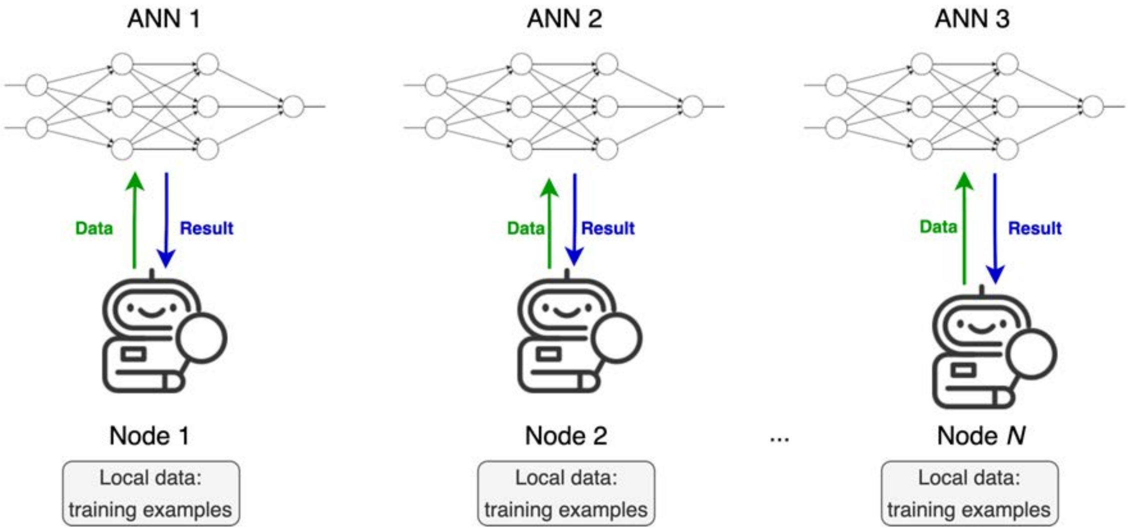


Figure 7.6: Locally-trained isolated clients

Table 7.1: Results of the locally-trained clients scenario

Instances per round	**Minibatch size ANN**	**Max. F-score**	**Min. F-score**	**Avg.**	σ
75	*25*	0.798	0.634	0.714	0.060
75	*50*	0.833	0.667	0.731	0.059
75	*75*	0.810	0.647	0.734	0.055
150	*25*	0.822	0.689	0.756	0.056
150	*50*	0.825	0.687	0.765	0.059
150	*75*	0.822	0.696	0.765	0.054

7.4.2. Locally-trained isolated clients

The scenario presented in this approach is more similar to the federated system. In this case, the main difference with respect to the latter is that the clients train in isolation with their local datasets without sharing their models or their data. As in the previous subsection, the objective is to check if the incremental nature of the training may provide an advantage by itself. Table 7.1 shows the results of the best and worst clients in terms of the mean and the standard deviation for the combinations of the metaparameters mentioned in Section 7.3.2. Figure 7.6 shows the pipeline of this approach.

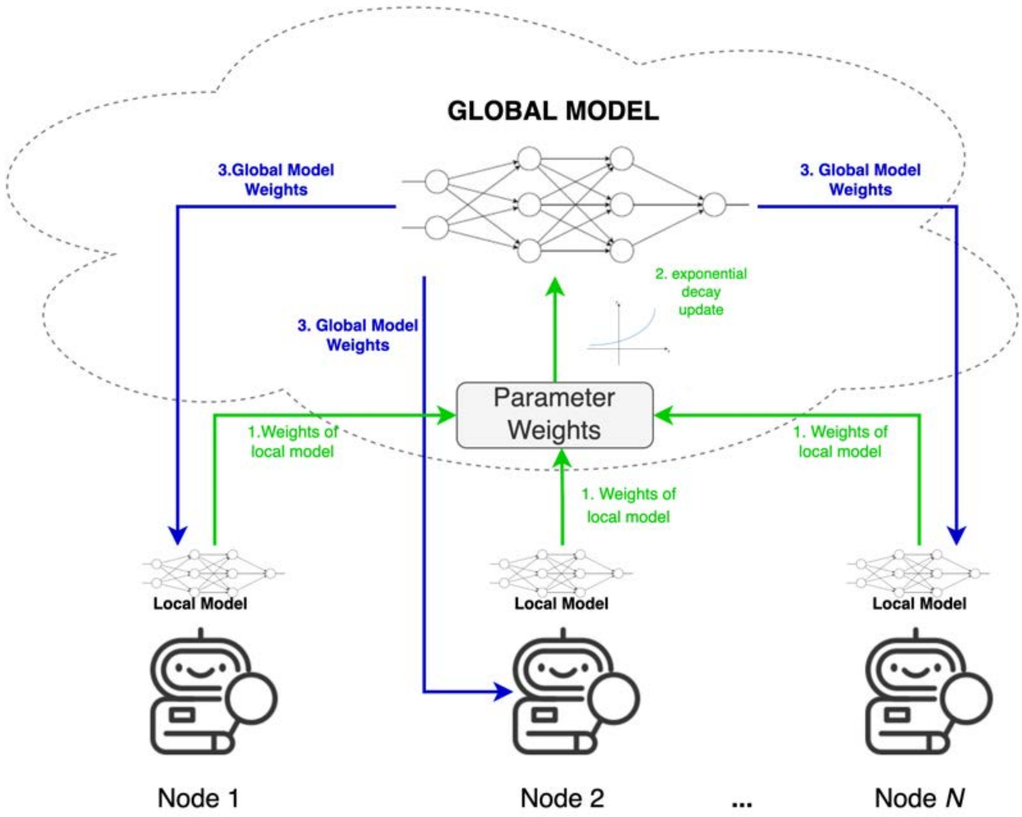


Figure 7.7: Our approach: federated learning

7.4.3. Federated approach

Lastly, our proposal in this work is the federated approach. In the *FedAvg* aggregation algorithm presented by [277], the weights of the different local models are averaged by the server to provide new weights and, thus, a new aggregated model. A variant of the *FedAvg* aggregation algorithm is presented for this system. More specifically, the weights of the incoming model, in the average calculated between the server and client models, are corrected in each iteration using an exponential function. Two different options explained in detail in Section 7.3.2.3, have been tested, an exponential decay function and an exponential growth function, respectively. We also performed no weight variation tests with a 'vanilla' version of the *FedAvg* system. The federated system's architecture is shown in Figure 7.7.

For this experiment, we went a step further by also measuring the impact that receiving the current global model has on the system right before training the model with the last version of the local dataset, as explained in the fourth step of the system's phases in Section 7.2. For

Table 7.2: Best results of the federated system per client compared with the results of the isolated clients' scenario. Each client's column represents its best F-score and the combination of metaparameters the whole system had when it achieved this score. Below the F-score value, the table shows the best score this client achieved during the locally-trained scenario and the last row, the difference. A positive difference implies that the federated approach improved this result.

	Client 1	**Client 2**	**Client 3**	**Client 4**	**Client 5**	**Client 6**	**Client 7**	
Federated mode	Weight decay	Weight growth	Constant weight	Weight decay	Weight decay	Weight growth	Weight growth	
Federated bias	0.75	0.75	0.75	0.75	0.5	0.5	0.75	
Federated rate	0.01	0.025	0	0.01	0.01	0.025	0.01	
Local train samples	150	75	75	150	75	150	150	
Minibatch size	50	50	25	50	75	50	50	
Pull?	No	No	No	No	No	No	No	**Average**
F-score	**0.759**	**0.887**	**0.788**	**0.800**	**0.793**	**0.861**	**0.852**	**0.820**
Best locally-trained	0.745	0.833	0.758	0.769	0.701	0.823	0.822	0.779
Difference	+0.014	+0.054	+0.030	+0.031	+0.092	+0.038	+0.030	+0.041

clarity, 'pull' indicates that the client has 'pulled' the global model before training, and 'no pull' indicates that the client has skipped this step.

Table 7.2 summarises the best results obtained by following the steps described above, with the corresponding system metaparameters and the difference between this result and the client's best result in the isolated case (explained before). The client models trained with *FedAvg* obtained a mean F-score of 0.822 on their own local test sets. This is slightly better than the local learning F-score of 0.781, which occurs when the clients train incrementally and are isolated from the server. Surprisingly, 'pulling' the model before training did not provide the best results for any client.

7.5. Discussion

Table 7.3 shows a complete comparison of the results. In the results from Section 5.2.2 (Table 5.10), we showed the logistic regression classifier performance in Mini with the same

Table 7.3: Summary of the results obtained in this study compared to the results obtained with four gestures in Section 5.2.2. The first row shows the multi-class results in the robot Mini obtained in Section 5.2.2, and the next rows display the F-scores obtained in the three scenarios tested in this Chapter.

Source		F-score
Previous multiclass results (Table 5.10)	Best estimator (Logistic)	0.870
	Equivalent model (MLP)	0.787
Distributed system	Regular	0.737
	Incremental	0.721
Locally-trained isolated clients	Average between best clients	0.779
Federated system	**Average between best clients**	**0.820**

four touch gestures ('tap', 'slap', 'tickle', and 'stroke') reached an F-score of 0.870. In this same experiment, the MLP model had an F-score of 0.787 (the first row in Table 7.3). This has particular relevance since the model introduced in the federated system is based on the same estimator. Currently, most state-of-the-art human-robot touch recognition solutions incorporate approaches based on similar models. Our work displays performance results similar to those shown in Section, as well as those of our proposed MLP model. The reader can perceive at first glance that the results improve upon not only previous work but also two more baseline approaches constructed with the same dataset as the federated system. First, the distributed system was divided into two options: a regular, one-time fit with the complete dataset and incremental training constructed like the federated system. In this case, using an artificial neural network with the same hyperparameters gives the worst results, with an F-score decay of 0.08 concerning the average F-score of the federated clients (as shown in the table, 0.822) in the case of the regular fit, and it provides an even worse F-score when the training is done incrementally.

The second baseline is a decentralised system in which the clients train incrementally, but in this case, without sharing any information, they train and validate with their datasets. These tests were designed to prove that the federated aggregation improved a much simpler distributed system. As the table shows, our approach improves all aspects of the results, from the best client's F-score to the average F-score computed from all the clients.

7.5.1. Limitations and lessons learned

Despite its advantages, the system also has its shortcomings. First, asynchronous federated learning aims to provide a freer learning environment for the nodes and reduce the loss of precision caused by extremely unrestrained learning. Despite this, it also some limitations that can impact the system's performance in the long term. For example, events such as uneven learning samples and different learning progress may arise in asynchronous federated learning because it is unreasonable to expect the nodes with large differences to update the global parameters equally. We expect to explore in further work the performance of our system in this aspect by augmenting the number of nodes and running the system for longer periods.

Following this idea, in future experiments, we could also explore if there is a maximum server capacity limit to attend simultaneous and uncoordinated updates of hundreds of nodes. In that case, it would be necessary to establish hardware and software mechanisms to avoid potential server congestion. However, in the tests carried out, we are far from such congestion in processing information. Concerning the software measures, it would be possible to design a queuing request system (or queues with priorities) so that requests can be attended to as soon as possible during a demand peak. It must be considered that updating the weights in the nodes would not require very low latency since it would not be critical for a node to continue its regular functioning while waiting for the next update. For the hardware measures, thanks to the cloud infrastructures provided by many service providers (such as AWS, Azure or Google Cloud), it would be relatively simple and low-cost to scale the server's computing power to the actual demand needs of our system.

Also, since the dataset for each of the robots is unique since it is composed of a unique set of users, the data distributions of the federated clients (robots) might differ greatly. This phenomenon is known as 'non-IID data distribution' [292], and it may cause severe model divergence, especially for parametric models in horizontal FL, which is the case presented in this chapter. More specifically, among the categories of non-IID data presented in [293], our dataset manifests a label distribution skew because the users were each asked to perform a random set of gestures. Many authors have proposed federated learning specifically to tackle the non-IID problem [277, 287]. However, this learning paradigm still has to face some challenges in this aspect, especially in long-term scenarios, due to the heterogeneity in local data distributions [292, 294–296].

In our case, taking into account that the dataset only suffers from one of the aforementioned skews, it has been decided to compensate for the effects that may arise from the non-IID distribution through a modification of the aggregation algorithm, that is, the successive modification of the average between the server model and the client model through an exponential function. Tests in this sense have been performed both with a decay function, in order to give less importance to the new model successively, and with an exponential growth function because the training of the multilayer perceptron in each round is performed with an incremental number of samples, which implies that the retrained models will perform better in successive rounds of communication.

The results have improved upon the results presented as the baseline in a system composed of seven nodes, but there is still room for improvement. In this regard, studying the maximum and minimum number of clients that the system could handle could also inspire further work concerning this system. In our particular case, this analysis was conditioned by the fact that the dataset was designed *ad-hoc* for this work, and we did not want to enlarge it artificially. In the case of wanting to scale the system to thousands of nodes running in parallel and wanting to update their weights with the central server, it could cause that, due to the limited resources of the server, it would have to incorporate some policy (such as the one presented in the works by [297] and [298]) to decide the priority with which to attend each request from the nodes.

On another matter, as a decentralised machine learning technique, federated learning addresses privacy concerns by distributing the training work to distributed users. However, this also brings some new security concerns. Privacy issues arise from two main aspects: the server's vulnerability and the clients' vulnerability. In the case of the server, the centralised aggregating scheme might be vulnerable to the malfunction of the former. Moreover, attackers may learn private information from these model parameters. With respect to the clients, threats can come from malicious participants. As an example, the Byzantine attack could also be implemented in the learning scheme. In a Byzantine attack, malicious client users may provide bad or low-quality updates to the server when they get the global model from the server. In this sense, more privacy-oriented approaches like the one presented in the works by [285, 286] have tried to provide solutions consisting of, for example, introducing a distributed peer-to-peer update scheme instead of the more common centralised update system or an update verification phase.

Finally, to end this discussion section, we would like to summarize in Table 7.4, the strengths and weaknesses of the work proposed here, in conjunction with other relatively similar and relevant works, but in other application areas. These application areas involve smartphones, autonomous electric vehicles, smart sensors and niche software.

7.6. Summary

This chapter proposes a federated learning system that can learn from decentralised data without exposing users' personal information. The nodes learn locally from their own datasets to interpret the tactile interaction between humans and the nodes, which are social robots in this case.

The system functions as follows. After collecting a certain number of instances in its local dataset, a node trains its local model locally. The model parameters are then uploaded to the server. After updating the global model, the server returns the resulting parameters to this node.

Application Domain	Studies	Pros	Constraint
Smartphone keyboards [299, 300]	Learn out-of-vocabulary words	Expanding the vocabulary of the keyboard without exporting sensitive text	Strongly relies on a learned probabilistic model
Smart devices motion sensors [301]	Human activity recognition	Identifies and reject erroneous clients	A little bit worse performance that centralized models
Image representation [302]	Obtain various types of image representations from different tasks	Be validated on three kinds of FL settings	More beneficial for the smaller dataset than the larger one in horizontal FL
Text mining [291]	Spam filtering and sentiment analysis	Provides guarantees on both data privacy and model F-score	Should take more reliable measurements
Electric vehicles [303]	Federated energy demand prediction	Applied the clustering-based energy demand learning method to improve the prediction F-score further	Need to be more stable and flexible
Robot network [304]	Robots imitation learning	Increases imitation learning efficiency of local robots in cloud robotic systems	Need to further work on convergence justification of the fusion process
Our approach - Human-Robot Touch Interaction	**Improve touch detection and classification**	**Asynchronous and distributed system, client-driven, multi-user and multi-robot system with incremental learning**	**Untested in production with thousands of nodes working in parallel and vulnerable cyber-attacks**

Table 7.4: A summary of some relevant federated learning-based applications, their pros and cons, and the proposed system.

All nodes perform this operation asynchronously without waiting for the remaining nodes or synchronising the learning process. During the learning process, the nodes only communicate with the parameter server; they have no information about the remaining nodes other than the shared global parameters.

In our approach, we used federated learning for the first time in a social robotics use case, specifically in the field of human-robot touch interaction. This enables collaborative and distributed learning by encouraging each robot to share its knowledge with other robots in the same environment without exposing its own data. This work also contributes to improving the previously presented touch system through a multi-robot, multi-user, and distributed learning approach. In addition, we present a client-driven asynchronous federated system in which clients decide when to upload their models to the server rather than the server forcing them to do so synchronously. The results obtained improved the rest of the approaches proposed as a baseline: a distributed version of the system with a global dataset; a system composed of multiple isolated clients training with only their local sets of samples; and, finally, since we were classifying 4 different gestures, the results of the MLP (the same algorithm implemented for our Federated system) presented in our previous work, in Section 5.2.

The contents from this chapter were included in the following journal publication:

Journal publication

Gamboa-Montero, J. J., Alonso-Martin, F., Marques-Villarroya, S., Sequeira, J., & Salichs, M. A. (2023). "Asynchronous federated learning system for human–robot touch interaction". *Expert Systems with Applications*, 211, 118510.

Chapter 8

Conclusions

This chapter gathers the main conclusions from the work presented in this document. It summarises the various achievements and contributions made during the course of the work. In addition, this chapter also describes the various limitations of the system. All these limitations, however, also define a set of future works that will allow continuing to develop the system presented from a technical point of view. For example, at a hardware level, we could incorporate other sensors into the system to further improve the system's recognition capabilities. On the other hand, at a software level, new automatic learning techniques to obtain high-level information about physical contact. These options would also allow us to explore the possibilities and limits of the system in terms of *sensor fusion*, a crucial concept related to our work and discussed in Section 2.1. Additionally, these improvements also unveil new applications of the system in tactile interaction and new methodologies to study social touch.

8.1. Achievements and Contributions

In the first place, we have to mention that the main goal, **designing a tactile sensing system able to improve human-robot social interaction**, has been successfully achieved in this work. The system was designed to classify touch gestures and locate their source. Compared with the literature, our work was able to achieve competitive results. More importantly, the system was integrated into real social robotic platforms and tested in a real environment during a touch interaction game. In this sense, our system proved to be a suitable human-robot touch interaction tool that could evaluate touch interaction from more complex aspects, such as its affective components. In our work, we approached this main goal from both a technical standpoint and human-robot interaction one. This makes our work multidisciplinary, as Chapter 2 showed. In this work, we merge the world of touch sensing technologies, smart tactile sensing systems and their applications in robotics with the study of social touch, centred on the effects that tactile interaction has on humans, not only during human-human interaction but also during human-robot interaction. In this sense, we conducted two studies focused on the latter, which also helped to develop a methodology in such studies in the future, where the ATR system could play a very promising role.

Regarding the hardware implementation, we designed a smart tactile sensing system to adapt to a social robot's particularities. **Different iterations of the system were tested on three social robotic platforms: Maggie, Mbot and Mini.** This implied adapting our design to their complex, curved surfaces made from hard and soft materials. Our design has successfully integrated acoustic sensing into an intelligent touch recognition system, which involved exploring different sensory and data acquisition technologies and evaluating the options available regarding interfaces and sound cards. Our design was focused on avoiding hindering the tactile interaction with the robot and ensuring the best positioning of sensors for optimising sound acquisition. We strived to maintain the complexity of the hardware deployment as low as possible, avoiding many sensors while covering significant areas of the outside of the robot.

Through this work we went through a sequence of hardware integrations, most of which are shown in chapter 3. In this evolution, we started from more bulky and expensive systems, such as the *Oyster Schaller* and *Recon3D* combination shown in the Maggie robot (Section 3.2.1), to the more compact integration shown in Mini, where we used much simpler and cheaper piezoelectric sensors and a customised soundcard. This also allowed us to reduce the volume occupied by the system substantially. Thanks to this volume reduction, it was possible to integrate it into a small desktop robot like Mini. **One of the main innovations we present is the introduction of sound-based sensing for touch recognition in social robotics.** As we show in detail in Chapter 2, this technology is far from novel in itself, as it has been mainly integrated into flat surfaces and interactive touch monitors. In these works, acoustic sensing has demonstrated recognising higher-level information. Above all, it has demonstrated advantages of great relevance to the field of tactile sensing technology, such as a large recognition range and a fast response speed. By introducing this technology in the field of social robotics, we have established a bridge from which multiple possibilities open up since, as mentioned in section 2.1.3, multiple works have deepened the concept of recognition and localisation of tactile gestures in the field of social robotics.

However, integrating contact microphones in social robotic platforms is a complex task. Social robots consist of complex, curved surfaces and piezoelectric microphones require a flat surface in order to pick up the vibration of the surface on which they are installed correctly. Regarding our implementation, in the first proofs of concept of the system, the sensors' surface installation on the social robot's polished and paint-covered surface showed promising results. At this point, two main problems emerged. Firstly, the appearance of a social robot is a fundamental characteristic. Therefore, altering this appearance with a device and its corresponding cable placed on the robot's surface could not be considered a successful integration. In addition to this, and more importantly, positioning the touch sensor close to the area where the interaction can occur is something that can condition the touch interaction itself, as the user can touch either the sensor directly or the cable to which it is connected. The logical solution was to introduce the sensors under the robot's surface to solve this problem. In this case, the problem is how the robot's inner surfaces are usually made. These surfaces often need to be better finished and, therefore, rough. In addition, these surfaces, because they are internal, make it even more difficult for the microphone to make proper contact with the surface to be sensed. To solve this problem, we propose using mouldable adhesive materials to create this surface and ensure perfect contact and adhesion of the device. Virtually all tests have shown that the combination of microphone and material does not impair the system's performance.

As the literature shows, in the field of social robotics, the sensors mainly used for tactile perception tasks use force and capacitive technologies. As work like Yohanan's demonstrates [305], such deployments tend to increase costs significantly as the contact surface grows. In his case, a social robot of about half a metre requires a full mesh of force sensors to recognise and localise touch interaction. In addition to the costs involved, the assembly and calibration of these systems are something the authors highlight. Therefore, one of our objectives in this regard was to ensure that the assembly of the system was not entirely conditioned by the platform on which it was to be installed. In other words, as far as possible, systems with similar components should be easy to assemble regardless of the platform. In this respect, our system partially fulfils this objective. While it was relatively easier to position the microphones in Maggie and Mbot, in the case of Mini, we did have to make slight modifications to the design in order to be able to insert the contact microphones into the arms. However, it should be noted that a single microphone on the arm senses the whole surface. In addition, the capacitive sensors' role in the system should also be mentioned as they are, in some cases, part of the default deployment of the system. Therefore, again, we cannot claim that the simplicity of assembly is complete, as in the robotic platforms we have used in this work, the system benefitted from the already installed capacitive sensing. Nevertheless, we have to mention that this has not involved the installation of extra capacitive sensors. In this case, our system has been integrated with the platform, improving its tactile capabilities.

The software deployment is highly modular, and we prioritised using tools such as ROS, known for this quality. Consequently, ROS served as our primary design tool for the system. ROS offers several extra benefits, including flexibility and scalability. By offering reliable and straightforward mechanisms to communicate each function, ROS enables the isolation of each robot functionality in a different node or set of nodes without affecting the architecture as a whole. We also focused on avoiding altering the platform's software, so **we integrated our system into the architecture of the robotic platforms.** In the process, we made the system exceptionally portable, using tools such as *Docker*, that allowed the containerisation of the system. Using these tools, we could achieve an easy software deployment regardless of the platform, avoiding the installation of extra libraries or tools in the robot.

Our ATR system is able to identify and recognise how the user is touching the robot. We implemented machine learning techniques for this purpose and defined a set of different touch gestures adapted to the features of the robotic platform. The ATR described in Chapter 4 is a software system able to recognize touch gestures through sound and capacitive sensor data analysis. The basic touch gesture recognition version of the system has the following. First, a Sound Acquisition (SA) stage, where the system, helped by the present sound architecture (ALSA and Pulseaudio systems), captures the information from the microphones and samples it. Our system controls these software interfaces and can set their parameters at this stage. Secondly, in the Touch Activity Detection (TAD) stage, the system uses the SNR and also the information from the capacitive sensors to minimize false positives (i.e., sounds that are not considered physical contact) and detect when the gesture is commencing. In this stage, we introduced ChucK, an on-the-fly sound processing tool that allowed real-time signal analysis. Afterwards, the third stage consists of Feature Extraction (FE), where the system delimits the duration of the physical contact and extracts the main features of the sound signal. Then, we proceeded to the Dataset Creation (DC) stage. In this stage, the system generates an instance representing the physical contact, which is then stored in a dataset for the next step. Finally, in the Touch Gesture Classification (TC) stage, we introduce machine learning tools to evaluate whether the information from the audio signal can be used to differentiate between different types of touch gestures.

We evaluated this touch classification approach through the tests presented in Section 5.1. In those tests designed at the very beginning of this work, we evaluated a system with just one piezoelectric receiver as a proof of concept. In those experiments, where we used Maggie as the robotic platform, 25 volunteers participated. And after gathering a dataset composed of 1981 samples, the first version of the system achieved an F-score of 0.81, using an LMT classifier in train/test evaluation focusing on touch gesture classification with one piezo microphone and 4 touch gestures ('tap', 'slap', 'stroke', 'tickle'). These results validated the initial proposal; therefore, we could expand the idea to a larger number of receivers.

With more microphones, an opportunity arose: we could improve the system to localise the source of the touch contact. For that, we explored two different approaches. On the one hand, we used machine learning techniques that allowed us to expand the classification of the touch gesture. On the other hand, we implemented sound analysis techniques, more specifically, SRP-PHAT, to classify the location more precisely. In the first approach, **we expanded the use of machine learning techniques, converting the contact location into a label to be classified by the system.** For that, we used two approaches. On the one hand, we kept each label as a separate multi-class machine learning problem. Despite this, both problems shared the same attributes extracted from the sound signal. On the other hand, we approached this problem through another category of machine learning techniques: multi-target algorithms. Through this approach, the algorithm establishes a relationship between both labels.

In this case, throughout the development of the entire work, we designed two different tests, mainly because this system was, in the end, the one employed for the HRI experiments from Chapter 6, covered later. In the first experiment, described in Section 5.2, we opted to maintain the four initial touch gestures and test the system on two different platforms: Maggie and Mini. Even though Mini also had foam in its chest, for this first test with multiple microphones, we preferred to use rigid materials for both platforms. Moreover, we were already testing two platforms with two different materials, fibreglass for Maggie and plastic in the case of Mini. For these experiments, we created two different new datasets since increasing the microphones implied more attributes in the instance; therefore, we could not use the one designed for the previous test with one microphone. Each dataset involved 20 users, and in terms of samples, the Maggie dataset comprised 3572 instances and the Mini dataset 2777. With those datasets, we obtained in the multi-class tests an F-score of 0.858 for Maggie and 0.870 in the case of Mini. For the multi-target tests, we had to employ a different metric for the evaluation, the Hamming score, also called multi-label accuracy. In this case, we achieved a Hamming score of 0.904 in the case of Maggie and 0.912 in the case of Mini. Both approaches demonstrated that the system could be successfully scaled using more microphones to cover larger surfaces of a robotic platform.

After validating our proposal with these offline evaluations, we decided to focus on Mini to create a suitable dataset for online classification. We focused on this robotic platform since, at that time, it was the only active platform in the laboratory. Through this test, we tried to expand further on some aspects that were not covered in enough detail in the previous test. Some of these aspects were the maximum suitable touch gestures that the system could distinguish in a desktop platform and also whether the system could handle microphones placed on different materials on the same robot. As we mentioned, the previous tests were focused on rigid materials, but one of Mini's features is that its chest is made of foam. In addition, foam is a material generally used for toys, as it is soft and squishy, which makes it more appealing to the touch. As a result, we could expand the number of gestures to 6, but more importantly, through a series of enhancements to the dataset, we did not lose performance in the process. Quite the contrary, since we achieved in multi-target online classification a Hamming score of 0.934. The dataset gathered from this experiment was originally composed of 3280 samples from 28 users, and the final dataset, after the modifications and augmentation using SMOTE, was composed of 5568 instances. This test validated the system for being used in the experiments presented in Chapter 6.

By implementing the second touch localisation approach, presented in Section 4.2.2, **we explored the possibilities that the sound signal analysis techniques offered to the ATR**. After thoroughly exploring the literature and the methods that could better adapt to our setting, a curved-shaped surface of a social robot, we decided that the more suitable method for implementation was SRP-PHAT. This technique uses features such as time differences between audio signals perceived between microphones to model the surface where contact occurs as a probability map. Even though in our work, we present this approach as a case of study, the results on the robot Mbot were satisfactory, being comparable to the ones presented in the literature, in Section 2.1.2. We achieved an average error of $3.63cm$ in the Mbot's thin, curved fibreglass surface. Although this approach was not fully integrated with the touch gesture classification and localisation modules, the system is prepared to do so, and we consider it a fascinating future work.

As we mentioned in the beginning, our goal did not stop with the technical aspects of the system. In this sense, we were a bit more ambitious, so we could introduce the system in the field of human-robot touch interaction, and more specifically, in the field of *social touch*. Therefore, we set to evaluate the possibilities that the system offered in terms of **affective communication and effects on the behaviour**, concepts that were explored in detail in Sections 2.2.2 and 2.2.3. We first proceeded to study how the system could contribute to affective communication. In order to achieve this, the first condition that the system had to meet was being sufficiently modular to be part of more sophisticated multimodal perception systems that capture higher-level affective messages. In this first experiment, we successfully demonstrated how our system could be part of a high-level one that combined information from the user's facial expression with the tactile information provided by the ATR. Despite the good preliminary results, this work was designed as a use case, and more evaluations must be carried out in the future.

Another important condition accomplished is that **our system operates in real-world applications**. Due to this, and as we explained in Section 4.3, we ruled out any purely offline integration of the system. This was motivated by the challenges proposed by Shiomi et al. [19] and presented in Section 2.2.4. By having a system working in the real world, we could prepare evaluations as the ones presented in Section 6.2, where the ATR system is the main tool evaluating the impact that human-robot touch interaction had on humans in the context of a memory game. **This experiment proved that interacting with the robot positively impacted engagement and fun, metrics that measure changes in the user's behaviour but also served as a qualitative evaluation of the ATR system.** In this aspect, we indirectly tested how the users perceived the system, and with some exceptions, and also judging by the quantitative results, the interaction was not worsened by slow response times or misclassifications. These results also offer new opportunities for designing human-robot touch interaction studies using the ATR as the main tool. In this sense, we project conducting studies in the future in other environments where touch could be a valuable communicative vehicle, such as learning environments. Alternatively, we could also explore more options regarding entertainment applications that could integrate the ATR system that evaluates other cognitive-related skills besides memory, like the user's reflexes.

Regarding the last implementation, described in Chapter 7, we explored **introducing more advanced state-of-the-art machine learning techniques**. Federated Learning, a distributed learning technique, is one example of this. We implemented a use case application where seven nodes were learning asynchronously from different users present only in each platform. With the implementation of this technique, we achieved three objectives. In the first place, we accelerated the process of gathering samples and achieved a common knowledge base. Up to this point, these are benefits common to the distributed machine learning paradigm. Secondly, and according to the current privacy-oriented trend regarding devices, Federated Learning offers protection for the privacy of the user of each of the devices since the element shared between robots is not the data itself but the machine learning model hyperparameters. Finally, we also achieved better results than the ones achieved in previous works, and not only this, the system improved a basic distributed architecture using the same dataset and the results obtained by the nodes training isolatedly without sharing data.

Finally, regarding the scientific contribution of the work presented, we have to highlight that **this work involved 5 in-person studies**. In the first place, there were 3 performance-related evaluations of the system in its different stages, contained in Chapter 5. Throughout these experiments, we gathered a total of 94 volunteers that helped test the system. On the other hand, in this work, we presented 2 HRI-related experiments. In this case, these experiments gathered 133 participants. So in total, throughout this work, **we were able to gather 227 participants for our experiments**, and it needs to be noted that all these evaluations were in-person. This required a huge investment in time, not only for the experimenters but also for the experimentees.

Regarding scientific production, in total, during the course of this research, 5 indexed journal publications have been published (at this moment (Q1 and Q2), and there are two extra manuscripts in review; 7 international conference papers, one of them awarded as the **best paper** of the ICSR 2022 14th International Conference on Social Robotics, and 5 Spanish national conference papers.

8.2. Future Research

Despite the achievements, the work presented also has room for improvement and promising future work. The first opportunity in this regard is related to the data-gathering process. Through Chapter 5, we observed that despite the system being portable hardware- and software-wise due to the introduction of lighter and compact hardware components and the use of containerisation tools such as Docker, we were not able to reutilise the datasets when elements like the number of microphones or the materials changed during implementations. In this sense, we did not explore in depth whether there is an option to fuse all the knowledge across robotic platforms to achieve an acoustic touch gestures database. As explained before, we implemented distributed learning paradigms like federated learning to cope with this issue when we have multiple robots with the same setup. However, this implementation is still just introduced in this work and needs to be fully integrated into the ATR.

Another interesting implementation is related to gathering instances from the non-supervised stage of the system, in this case, the OC stage. When solving a machine learning problem, it is often necessary to acquire labelled data through a knowledgeable human agent or a physical experiment. Because of the expense involved in the labelling process, it may not be possible to acquire large training sets that are completely labelled. On the other hand, the acquisition of unlabeled data is relatively inexpensive. When used in conjunction with a limited amount of labelled data, unlabeled data can significantly improve learning accuracy. Weak supervision is a branch of machine learning where noisy, constrained, or unreliable sources help generate supervision signals for labelling large amounts of training data [306, 307]. This method lessens the burden of acquiring hand-labelled data sets, which can be expensive or unworkable in certain circumstances. Rather, cheap weak labels are used with the knowledge that despite being flawed, they can still be used to build a robust predictive model. Semi-supervised learning is a type of weak supervision where a small amount of labelled data is combined with a large amount of unlabeled data during training [308, 309]. In our system, a technique that could fit as a future implementation is a semi-supervised technique known as pseudo-labelling. Pseudo-labelling [310] is a semi-supervised learning method used to improve the model's performance in case of limited labelled data. The basic idea is to use the model's prediction on unlabelled data as synthetic labels, or *pseudo-labels*, for that data during training. The model is first trained on the labelled dataset and then used to make predictions on the unlabelled dataset. The more confident predictions are considered pseudo-labels and are used to train the model further. It helps us a large amount of unlabelled data and reduces the entropy of the model's predictions, which might lead to better performance on test data.

Another intriguing effect we addressed in this work was the influence of ambient noises and how the contact microphones registered those. Nevertheless, in our experiments, this rarely happened as the intensity of sounds propagating in the air lowers when changing to a solid material. Despite this, we acknowledged that intense ambient noises, when they could exert vibrations on the robot's shell, could be registered by the contact microphones and therefore cause false positives. This was the main reason for introducing traditional touch-sensing technologies, such as capacitive touch sensors, as elements from which the ATR could benefit. However, this addresses one aspect of the impact of noise in the system. In our work, we did not detail the effects that internal and external noises could have in the system during a gesture that is currently happening, i.e., how these noises could pollute the sample. During the data-gathering processes detailed in Chapter 5 the robot was static to make the interaction with the user significantly easier. In this aspect, introducing the system to a long-term scenario in a real environment could help us understand the system's robustness against these phenomena and could be an interesting opportunity to evaluate other touch interaction scenarios.

Regarding the different studies about the meaning of touch, we could explore further possible cultural biases that exist during the interaction. Following this idea, this kind of information could be gathered in future studies to establish the distinctions in culture, gender or age. As mentioned, one of the objectives of this platform is to be a tool for studying how these factors, mentioned by Saarinen and Shiomi in Section 2.2.4, could condition the touch interaction. Some of these factors were not gathered in the performance experiments from Chapter 5 since these experiments were approached from a technical standpoint. However, in the cases presented in Chapter 6, some information regarding age and gender was gathered. Therefore a preliminary study could be done to evaluate whether these factors might be related to how the volunteer experienced the contacts on his/her arm or how the game was experienced. In this sense, we could even design new studies that could take advantage of the methodology already developed but are more oriented towards evaluating these factors. How the users are presented with the touch gestures could also be evaluated in more detail. Even though the users were presented with a video and definition of the touch gesture, we acknowledge that observing how the gesture has to be performed might have biased the user. We do not know to what extent, but this aspect could be explored in future studies.

Another future goal for the system could be integrating into other social robotic platforms more prone to touch interaction. For example, robots with similar morphology to Yohanan's Haptic Creature or Shibata's Paro could be an interesting platform to integrate our system or develop touch-related experiments. Currently, in the Social Robotics group in the UC3M, we are a 'huggable' and portable robot, similar to the previous example, where the ATR could be integrated. Interestingly, integrating the ATR in such platforms could allow us to explore more touch gestures, expanding the current set of gestures. As the reader might recall, the original set from Yohanan, adapted to the Haptic Creature, comprised gestures we had to discard, such as 'cradle' or 'rock'. We could test in future studies system's performance when classifying these kinds of gestures, which, according to Yohanan himself, had a significant affective component. Another advantage of introducing the system in these robotic platforms is that they are designed to emit little internal noise, making them suitable companions (nobody wants a companion that squeaks loudly). Furthermore, they are equipped with low-noise servomotors that would not interfere with the tactile detection of the ATR.

However, it is important to note that these platforms and, more importantly, these extra touch gestures also had an extra component that a sound-based system like ours might not be able to handle correctly: movement. These platforms are meant to be moved around, and although the ATR in its current iteration might not capture these gestures through their sound fingerprint alone, we could integrate an extra sensor into the equation. Along Section 2.1, we emphasise the importance of sensor fusion in STS systems. This was the primary concept behind having multiple microphones and combining their information with the touch detection provided by capacitive sensing. We could go a step further in this direction and combine another valuable source of information in this kind of soft-skinned robot: an inertial measurement unit (IMU). An IMU, through the measurement of the robot's orientation and the angular and linear acceleration, could help in both TAD and FE stages. Regarding the TAD phase, the system could evaluate sudden accelerations as the start of a certain type of touch gesture. In addition to this, the information regarding acceleration and orientation could be processed in the time and frequency domains and provide extra information to the instance that would represent the touch gesture.

Bibliography

[1] T. Field, *Touch*, en. MIT Press, Oct. 2014.

[2] L. K. Frank, 'Tactile communication,' *Genetic Psychology Monographs*, vol. 56, pp. 209–255, 1957.

[3] A. Montagu, *Touching: The Human Significance of the Skin*, en. Harper & Row, 1978.

[4] M. A. Heller and W. Schiff, *The Psychology of Touch*, en. Taylor & Francis, 1991.

[5] M. J. Hertenstein, J. M. Verkamp, A. M. Kerestes and R. M. Holmes, 'The communicative functions of touch in humans, nonhuman primates, and rats: A review and synthesis of the empirical research,' eng, *Genetic, Social, and General Psychology Monographs*, vol. 132, no. 1, pp. 5–94, Feb. 2006.

[6] J. Nie, M. Park, A. L. Marin and S. S. Sundar, 'Can you hold my hand? Physical warmth in human-robot interaction,' in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2012, pp. 201–202.

[7] K. Altun and K. E. MacLean, 'Recognizing affect in human touch of a robot,' *Pattern Recognition Letters*, vol. 66, pp. 31–40, Nov. 2015.

[8] A. Gallace and C. Spence, 'The science of interpersonal touch: An overview,' *Neuroscience & Biobehavioral Reviews*, Touch, Temperature, Pain/Itch and Pleasure, vol. 34, no. 2, pp. 246–259, Feb. 2010.

[9] S. E. Jones and A. E. Yarbrough, 'A naturalistic study of the meanings of touch,' *Communication Monographs*, vol. 52, no. 1, pp. 19–56, Mar. 1985.

[10] L. S. Löken, J. Wessberg, I. Morrison, F. McGlone and H. Olausson, 'Coding of pleasant touch by unmyelinated afferents in humans,' en, *Nature Neuroscience*, vol. 12, no. 5, pp. 547–548, May 2009.

[11] Q. Liu, S. Vrontou, F. L. Rice, M. J. Zylka, X. Dong and D. J. Anderson, 'Molecular genetic visualization of a rare subset of unmyelinated sensory neurons that may detect gentle touch,' en, *Nature Neuroscience*, vol. 10, no. 8, pp. 946–948, Aug. 2007.

[12] C. Breazeal, 'Toward sociable robots,' *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 167–175, 2003.

[13] C. Bartneck and J. Forlizzi, 'A design-centred framework for social human-robot interaction,' in *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759)*, Sep. 2004, pp. 591–594.

[14] G. Huisman, 'Social Touch Technology: A Survey of Haptic Technology for Social Touch,' *IEEE Transactions on Haptics*, vol. 10, no. 3, pp. 391–408, Jul. 2017.

[15] B. Reeves and C. I. Nass, *The media equation: How people treat computers, television, and new media like real people and places*, ser. The media equation: How people treat computers, television, and new media like real people and places. New York, NY, US: Cambridge University Press, 1996.

[16] M. M. Jung, X. L. Cang, M. Poel and K. E. MacLean, 'Touch Challenge '15: Recognizing Social Touch Gestures,' in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15, New York, NY, USA: ACM, 2015, pp. 387–390.

[17] S. Yohanan and K. E. MacLean, 'The Role of Affective Touch in Human-Robot Interaction: Human Intent and Expectations in Touching the Haptic Creature,' en, *International Journal of Social Robotics*, vol. 4, no. 2, pp. 163–180, Apr. 2012.

[18] M. M. Jung, M. Poel, D. Reidsma and D. K. J. Heylen, 'A First Step toward the Automatic Understanding of Social Touch for Naturalistic Human–Robot Interaction,' English, *Frontiers in ICT*, vol. 4, 2017.

[19] M. Shiomi, H. Sumioka and H. Ishiguro, 'Survey of Social Touch Interaction Between Humans and Robots,' *Journal of Robotics and Mechatronics*, vol. 32, no. 1, pp. 128–135, 2020.

[20] R. Jütte, 'Haptic perception: An historical approach,' en, in *Human Haptic Perception: Basics and Applications*, M. Grunwald, Ed., Basel: Birkhäuser, 2008, pp. 3–13.

[21] L. Zou, C. Ge, Z. J. Wang, E. Cretu and X. Li, 'Novel Tactile Sensor Technology and Smart Tactile Sensing Systems: A Review,' en, *Sensors*, vol. 17, no. 11, p. 2653, Nov. 2017.

[22] B. D. Argall and A. G. Billard, 'A survey of Tactile Human–Robot Interactions,' en, *Robotics and Autonomous Systems*, vol. 58, no. 10, pp. 1159–1176, Oct. 2010.

[23] S. J. Yohanan, 'The Haptic Creature : Social human-robot interaction through affective touch,' eng, Ph.D. dissertation, University of British Columbia, 2012.

[24] J. Paradiso, C. K. Leo, N. Checka and K. Hsiao, ‘Passive acoustic sensing for tracking knocks atop large interactive displays,’ in *2002 IEEE SENSORS*, vol. 1, Jun. 2002, 521–527 vol.1.

[25] F. McGlone, J. Wessberg and H. Olausson, ‘Discriminative and affective touch: Sensing and feeling,’ eng, *Neuron*, vol. 82, no. 4, pp. 737–755, May 2014.

[26] M. I. Tiwana, S. J. Redmond and N. H. Lovell, ‘A review of tactile sensing technologies with applications in biomedical engineering,’ en, *Sensors and Actuators A: Physical*, vol. 179, pp. 17–31, Jun. 2012.

[27] R. S. Dahiya, G. Metta, M. Valle and G. Sandini, ‘Tactile Sensing—From Humans to Humanoids,’ *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, Feb. 2010.

[28] C. Wang, L. Dong, D. Peng and C. Pan, ‘Tactile Sensors for Advanced Intelligent Systems,’ en, *Advanced Intelligent Systems*, vol. 1, no. 8, p. 1 900 090, 2019.

[29] L. E. Hollander, G. L. Vick and T. J. Diesel, ‘The Piezoresistive Effect and its Applications,’ *Review of Scientific Instruments*, vol. 31, no. 3, pp. 323–327, Mar. 1960.

[30] D. Damjanovic, ‘Ferroelectric, dielectric and piezoelectric properties of ferroelectric thin films and ceramics,’ en, *Reports on Progress in Physics*, vol. 61, no. 9, p. 1267, Sep. 1998.

[31] X. Liu, I. I. Iordachita, X. He, R. H. Taylor and J. U. Kang, ‘Miniature fiber-optic force sensor based on low-coherence Fabry-Pérot interferometry for vitreoretinal microsurgery,’ EN, *Biomedical Optics Express*, vol. 3, no. 5, pp. 1062–1076, May 2012.

[32] M. Grunwald, *Human haptic perception: Basics and applications*. Springer Science & Business Media, 2008.

[33] M. L. II, D. Hall, A. D. Poularikas and J. Llinas, Eds., *Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition*, 2nd ed. Boca Raton: CRC Press, Jan. 2017.

[34] S. Jia and V. J. Santos, ‘Tactile Perception for Teleoperated Robotic Exploration within Granular Media,’ *ACM Transactions on Human-Robot Interaction*, vol. 10, no. 4, 34:1–34:27, Jul. 2021.

[35] P. Mittendorfer and G. Cheng, ‘Humanoid Multimodal Tactile-Sensing Modules,’ *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 401–410, Jun. 2011.

[36] Q. Leboutet, F. Bergner and G. Cheng, ‘Online Configuration Selection for Redundant Arrays of Inertial Sensors: Application to Robotic Systems Covered with a Multimodal Artificial Skin,’ in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 10 873–10 879.

[37] G. Walker, ‘A review of technologies for sensing contact location on the surface of a display,’ en, *Journal of the Society for Information Display*, vol. 20, no. 8, pp. 413–440, 2012.

[38] P. Lopes, R. Jota and J. A. Jorge, ‘Augmenting touch interaction through acoustic sensing,’ in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11*, New York, New York, USA: ACM Press, Nov. 2011, p. 53.

[39] M. Ono, B. Shizuki and J. Tanaka, ‘Touch & activate: Adding interactivity to existing objects using active acoustic sensing,’ in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ser. UIST '13, New York, NY, USA: Association for Computing Machinery, Oct. 2013, pp. 31–40.

[40] J. P. Nikolovski, ‘Moderately reverberant learning ultrasonic pinch panel,’ *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 60, no. 10, pp. 2105–2120, Oct. 2013.

[41] Y. Liu, J. P. Nikolovski, M. Hafez, N. Mechbal and M. Verge, ‘Acoustic wave approach for multi-touch tactile sensing,’ in *2009 International Symposium on Micro-NanoMechatronics and Human Science*, Nov. 2009, pp. 574–579.

[42] K. Firouzi, A. Nikoozadeh, T. E. Carver and B. P. T. Khuri-Yakub, ‘Lamb Wave Multitouch Ultrasonic Touchscreen,’ *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 12, pp. 2174–2186, 2016.

[43] R. Xiao, G. Lew, J. Marsanico, D. Hariharan, S. Hudson and C. Harrison, ‘Toffee: Enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation,’ in *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, ser. MobileHCI '14, New York, NY, USA: Association for Computing Machinery, Sep. 2014, pp. 67–76.

[44] J.-Y. Jeong, J.-H. Kim, H.-Y. Yoon and J.-W. Jeong, ‘Knock&Tap: Classification and Localization of Knock and Tap Gestures using Deep Sound Transfer Learning,’ in *Companion Publication of the 2021 International Conference on Multimodal Interaction*, ser. ICMI '21 Companion, New York, NY, USA: Association for Computing Machinery, 2021, pp. 1–6.

[45] A. Seshan, ‘ALTo: Ad Hoc High-Accuracy Touch Interaction Using Acoustic Localization,’ *arXiv:2108.06837 [cs]*, Aug. 2021.

[46] H. R. Nicholls and M. H. Lee, ‘A Survey of Robot Tactile Sensing Technology,’ en, *The International Journal of Robotics Research*, vol. 8, no. 3, pp. 3–30, Jun. 1989.

[47] H. Liu, D. Guo, F. Sun, W. Yang, S. Furber and T. Sun, ‘Embodied tactile perception and learning,’ en, *Brain Science Advances*, vol. 6, no. 2, pp. 132–158, Jun. 2020.

[48] H. Iwata and S. Sugano, 'Design of human symbiotic robot TWENDY-ONE,' in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 580–586.

[49] S. Šabanović, C. C. Bennett, W.-L. Chang and L. Huber, 'PARO robot affects diverse interaction modalities in group sensory therapy for older adults with dementia,' in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, Jun. 2013, pp. 1–6.

[50] T. Morita, H. Iwata and S. Sugano, 'Development of human symbiotic robot: WENDY,' in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 4, May 1999, 3183–3188 vol.4.

[51] T. Minato, Y. Yoshikawa, T. Noda, S. Ikemoto, H. Ishiguro and M. Asada, 'CB2: A child robot with biomimetic body for cognitive developmental robotics,' in *2007 7th IEEE-RAS International Conference on Humanoid Robots*, Nov. 2007, pp. 557–562.

[52] K. Goris, J. Saldien, B. Vanderborght and D. Lefeber, 'Mechanical design of the huggable robot probo,' *International Journal of Humanoid Robotics*, vol. 08, no. 03, pp. 481–511, Sep. 2011.

[53] D. Silvera Tawil, D. Rye and M. Velonaki, 'Touch modality interpretation for an EIT-based sensitive skin,' in *2011 IEEE International Conference on Robotics and Automation*, IEEE, May 2011, pp. 3770–3776.

[54] D. Silvera-Tawil, D. Rye and M. Velonaki, 'Interpretation of social touch on an artificial arm covered with an EIT-based sensitive skin,' *International Journal of Social Robotics*, vol. 6, no. 4, pp. 489–505, 2014.

[55] W. R. B. Lionheart, 'EIT reconstruction algorithms: Pitfalls, challenges and recent developments,' en, *Physiological Measurement*, vol. 25, no. 1, p. 125, Feb. 2004.

[56] A. Flagg, D. Tam, K. MacLean and R. Flagg, 'Conductive fur sensing for a gesture-aware furry robot,' in *2012 IEEE Haptics Symposium (HAPTICS)*, Mar. 2012, pp. 99–104.

[57] S. Albawi, O. Bayat, S. Al-Azawi and O. N. Ucan, *Social Touch Gesture Recognition Using Convolutional Neural Network*, en, Research article, 2018.

[58] S. Müller and H.-M. Gross, 'Making a Socially Assistive Robot Companion Touch Sensitive,' en, in *Haptics: Science, Technology, and Applications*, D. Prattichizzo, H. Shinoda, H. Z. Tan, E. Ruffaldi and A. Frisoli, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 476–488.

[59] D. Hughes, A. Krauthammer and N. Correll, 'Recognizing social touch gestures using recurrent and convolutional neural networks,' in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2315–2321.

[60] N. Zhou and J. Du, 'Recognition of Social Touch Gestures Using 3D Convolutional Neural Networks,' en, in *Pattern Recognition*, T. Tan, X. Li, X. Chen, J. Zhou, J. Yang and H. Cheng, Eds., ser. Communications in Computer and Information Science, Singapore: Springer, 2016, pp. 164–173.

[61] M. D. Cooney, S. Nishio and H. Ishiguro, 'Recognizing affection for a touch-based interaction with a humanoid robot,' in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 1420–1427.

[62] Z. Zeng, M. Pantic, G. I. Roisman and T. S. Huang, 'A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, Jan. 2009.

[63] F. A. Geldard, 'Some Neglected Possibilities of Communication,' *Science*, vol. 131, no. 3413, pp. 1583–1588, May 1960.

[64] M. Hertenstein, D. Keltner and B. App, 'Touch communicates distinct emotions.,' *Emotion*, 2006.

[65] H. Olausson, J. Wessberg, I. Morrison, F. McGlone and A. Vallbo, 'The neurophysiology of unmyelinated tactile afferents,' eng, *Neuroscience and Biobehavioral Reviews*, vol. 34, no. 2, pp. 185–191, Feb. 2010.

[66] I. Morrison, L. S. Löken and H. Olausson, 'The skin as a social organ,' eng, *Experimental Brain Research*, vol. 204, no. 3, pp. 305–314, Jul. 2010.

[67] A. H. Crusco and C. G. Wetzel, 'The Midas Touch: The Effects of Interpersonal Touch on Restaurant Tipping,' en, *Personality and Social Psychology Bulletin*, vol. 10, no. 4, pp. 512–517, Dec. 1984.

[68] D. Erceau and N. Guéguen, 'Tactile Contact and Evaluation of the Toucher,' *The Journal of Social Psychology*, vol. 147, no. 4, pp. 441–444, Aug. 2007.

[69] T. Field, 'Touch for socioemotional and physical well-being: A review,' en, *Developmental Review*, vol. 30, no. 4, pp. 367–383, Dec. 2010.

[70] M. J. Hertenstein, 'Touch: Its Communicative Functions in Infancy,' *Human Development*, vol. 45, no. 2, pp. 70–94, 2002.

[71] H. F. Harlow, 'The nature of love,' *American Psychologist*, vol. 13, pp. 673–685, 1958.

[72] H. F. Harlow and R. R. Zimmermann, 'The Development of Affectional Responses in Infant Monkeys,' *Proceedings of the American Philosophical Society*, vol. 102, no. 5, pp. 501–509, 1958.

[73] R. Heslin and T. Alper, 'Touch: A bonding gesture,' *Nonverbal interaction*, pp. 47–75, 1983.

[74] P. W. Miller, 'Nonverbal Communication. Third Edition. What Research Says to the Teacher,' en, NEA Professional Library, P, Tech. Rep., 1988.

[75] E. McDaniel and P. A. Andersen, 'International Patterns of Interpersonal Tactile Communication: A Field Study,' en, *Journal of Nonverbal Behavior*, vol. 22, no. 1, pp. 59–75, Mar. 1998.

[76] A. Sorokowska *et al.*, 'Affective Interpersonal Touch in Close Relationships: A Cross-Cultural Perspective,' en, *Personality and Social Psychology Bulletin*, vol. 47, no. 12, pp. 1705–1721, Dec. 2021.

[77] J. K. Burgoon, J. B. Walther and E. J. Baesler, 'Interpretations, Evaluations, and Consequences of Interpersonal Touch,' *Human Communication Research*, vol. 19, no. 2, pp. 237–263, Dec. 1992.

[78] J. D. Fisher, M. Rytting and R. Heslin, 'Hands Touching Hands: Affective and Evaluative Effects of an Interpersonal Touch,' *Sociometry*, vol. 39, no. 4, pp. 416–421, 1976.

[79] J. Hornik, 'Tactile Stimulation and Consumer Response,' *Journal of Consumer Research*, vol. 19, no. 3, pp. 449–458, Dec. 1992.

[80] C. L. Kleinke, 'Compliance to requests made by gazing and touching experimenters in field settings,' en, *Journal of Experimental Social Psychology*, vol. 13, no. 3, pp. 218–223, May 1977.

[81] N. Guéguen and C. Jacob, 'The effect of touch on tipping: An evaluation in a French bar,' en, *International Journal of Hospitality Management*, vol. 24, no. 2, pp. 295–299, Jun. 2005.

[82] N. Guéguen, C. Jacob and G. Boulbry, 'The effect of touch on compliance with a restaurant's employee suggestion,' en, *International Journal of Hospitality Management*, vol. 26, no. 4, pp. 1019–1023, Dec. 2007.

[83] E. Anisfeld, V. Casper, M. Nozyce and N. Cunningham, 'Does Infant Carrying Promote Attachment? An Experimental Study of the Effects of Increased Physical Contact on the Development of Attachment,' *Child Development*, vol. 61, no. 5, pp. 1617–1627, 1990.

[84] I. Bretherton, 'The origins of attachment theory: John Bowlby and Mary Ainsworth,' *Developmental Psychology*, vol. 28, pp. 759–775, 1992.

[85] S. J. Weiss, P. Wilson, M. J. Hertenstein and R. Campos, 'The tactile context of a mother's caregiving: Implications for attachment of low birth weight infants,' en, *Infant Behavior and Development*, vol. 23, no. 1, pp. 91–111, Jan. 2000.

[86] M. Main, N. Kaplan and J. Cassidy, 'Security in Infancy, Childhood, and Adulthood: A Move to the Level of Representation,' *Monographs of the Society for Research in Child Development*, vol. 50, no. 1/2, pp. 66–104, 1985.

[87] M. S. Takeuchi, H. Miyaoka, A. Tomoda, M. Suzuki, Q. Liu and T. Kitamura, 'The effect of interpersonal touch during childhood on adult attachment and depression: A neglected area of family and developmental psychology?' *Journal of Child and Family Studies*, vol. 19, pp. 109–117, 2010.

[88] C. Sue Carter, 'Neuroendocrine Perspectives on Social Attachment and Love,' en, *Psychoneuroendocrinology*, vol. 23, no. 8, pp. 779–818, Nov. 1998.

[89] R. C. Fraley and P. R. Shaver, 'Adult Romantic Attachment: Theoretical Developments, Emerging Controversies, and Unanswered Questions,' en, *Review of General Psychology*, vol. 4, no. 2, pp. 132–154, Jun. 2000.

[90] A. K. Gulledge, M. H. Gulledge and R. F. Stahmannn, 'Romantic Physical Affection Types and Relationship Satisfaction,' *The American Journal of Family Therapy*, vol. 31, no. 4, pp. 233–242, Jul. 2003.

[91] R. Feldman, 'Oxytocin and social affiliation in humans,' en, *Hormones and Behavior*, Oxytocin, Vasopressin and Social Behavior, vol. 61, no. 3, pp. 380–391, Mar. 2012.

[92] K. Uvnäs-Moberg, 'Physiological and Endocrine Effects of Social Contact,' en, *Annals of the New York Academy of Sciences*, vol. 807, no. 1, pp. 146–163, 1997.

[93] K. Uvnäs-Moberg, I. Arn and D. Magnusson, 'The psychobiology of emotion: The role of the oxytocinergic system,' en, *International Journal of Behavioral Medicine*, vol. 12, no. 2, pp. 59–65, Jun. 2005.

[94] K. Maclean, 'The impact of institutionalization on child development,' en, *Development and Psychopathology*, vol. 15, no. 4, pp. 853–884, Dec. 2003.

[95] C. Beckett *et al.*, 'Do the Effects of Early Severe Deprivation on Cognition Persist Into Early Adolescence? Findings From the English and Romanian Adoptees Study,' en, *Child Development*, vol. 77, no. 3, pp. 696–711, 2006.

[96] H. T. Chugani, M. E. Behen, O. Muzik, C. Juhász, F. Nagy and D. C. Chugani, 'Local Brain Functional Activity Following Early Deprivation: A Study of Postinstitutionalized Romanian Orphans,' en, *NeuroImage*, vol. 14, no. 6, pp. 1290–1301, Dec. 2001.

[97] C. A. Nelson, 'A Neurobiological Perspective on Early Human Deprivation,' en, *Child Development Perspectives*, vol. 1, no. 1, pp. 13–18, 2007.

[98] S. L. Master, N. I. Eisenberger, S. E. Taylor, B. D. Naliboff, D. Shirinyan and M. D. Lieberman, 'A Picture's Worth: Partner Photographs Reduce Experimentally Induced Pain,' en, *Psychological Science*, vol. 20, no. 11, pp. 1316–1318, Nov. 2009.

[99] B. Ditzen *et al.*, 'Effects of different kinds of couple interaction on cortisol and heart rate responses to stress in women,' en, *Psychoneuroendocrinology*, vol. 32, no. 5, pp. 565–574, Jun. 2007.

[100] K. M. Grewen, B. J. Anderson, S. S. Girdler and K. C. Light, 'Warm Partner Contact Is Related to Lower Cardiovascular Reactivity,' *Behavioral Medicine*, vol. 29, no. 3, pp. 123–130, Jan. 2003.

[101] J. A. Coan, H. S. Schaefer and R. J. Davidson, 'Lending a Hand: Social Regulation of the Neural Response to Threat,' en, *Psychological Science*, vol. 17, no. 12, pp. 1032–1039, Dec. 2006.

[102] V. M. Drescher, H. W. Gantt and W. E. Whitehead, 'Heart Rate Response to Touch,' en-US, *Psychosomatic Medicine*, vol. 42, no. 6, pp. 559–565, Nov. 1980.

[103] S. J. Whitcher and J. D. Fisher, 'Multidimensional reaction to therapeutic touch in a hospital setting,' eng,*Journal of Personality and Social Psychology*, vol. 37, no. 1, pp. 87–96, Jan. 1979.

[104] M. J. Hertenstein and J. J. Campos, 'Emotion Regulation Via Maternal Touch,' en, *Infancy*, vol. 2, no. 4, pp. 549–566, 2001.

[105] M. Hertenstein, R. Holmes, M. McCullough and D. Keltner, 'The communication of emotion via touch.,' *Emotion*, 2009.

[106] H. A. Elfenbein and N. Ambady, 'On the universality and cultural specificity of emotion recognition: A meta-analysis,' *Psychological Bulletin*, vol. 128, pp. 203–235, 2002.

[107] A. Fotopoulou, M. von Mohr and C. Krahé, 'Affective regulation through touch: Homeostatic and allostatic mechanisms,' en, *Current Opinion in Behavioral Sciences*, vol. 43, pp. 80–87, Feb. 2022.

[108] M. von Mohr, L. P. Kirsch and A. Fotopoulou, 'Social touch deprivation during COVID-19: Effects on psychological wellbeing and craving interpersonal touch,' *Royal Society Open Science*, vol. 8, no. 9, p. 210 287, Sep. 2021.

[109] A. Serino and P. Haggard, 'Touch and the body,' en, *Neuroscience & Biobehavioral Reviews*, Touch, Temperature, Pain/Itch and Pleasure, vol. 34, no. 2, pp. 224–236, Feb. 2010.

[110] A. Gentsch, E. Panagiotopoulou and A. Fotopoulou, 'Active Interpersonal Touch Gives Rise to the Social Softness Illusion,' en, *Current Biology*, vol. 25, no. 18, pp. 2392–2397, Sep. 2015.

[111] G. Huisman, M. Bruijnes, J. Kolkmeier, M. Jung, A. Darriba Frederiks and Y. Rybarczyk, 'Touching Virtual Agents: Embodiment and Mind,' en, in *Innovative and Creative Developments in Multimodal Interaction Systems*, Y. Rybarczyk, T. Cardoso, J. Rosas and L. M. Camarinha-Matos, Eds., ser. IFIP Advances in Information and Communication Technology, Berlin, Heidelberg: Springer, 2014, pp. 114–138.

[112] J. B. F. van Erp and A. Toet, 'Social Touch in Human–Computer Interaction,' *Frontiers in Digital Humanities*, vol. 2, 2015.

[113] A. Haans and W. IJsselsteijn, 'Mediated social touch: A review of current research and future directions,' en, *Virtual Reality*, vol. 9, no. 2, pp. 149–159, Mar. 2006.

[114] J. Gratch, J. Rickel, E. Andre, J. Cassell, E. Petajan and N. Badler, 'Creating interactive virtual humans: Some assembly required,' *IEEE Intelligent Systems*, vol. 17, no. 4, pp. 54–63, Jul. 2002.

[115] C. Breazeal, 'Emotion and sociable humanoid robots,' en, *International Journal of Human-Computer Studies*, Applications of Affective Computing in Human-Computer Interaction, vol. 59, no. 1, pp. 119–155, Jul. 2003.

[116] B. Robins, F. Amirabdollahian, Z. Ji and K. Dautenhahn, 'Tactile interaction with a humanoid robot for children with autism: A case study analysis involving user requirements and results of an initial implementation,' in *19th International Symposium in Robot and Human Interactive Communication*, Sep. 2010, pp. 704–711.

[117] B. Robins and K. Dautenhahn, 'Tactile Interactions with a Humanoid Robot: Novel Play Scenario Implementations with Children with Autism,' en, *International Journal of Social Robotics*, vol. 6, no. 3, pp. 397–415, Aug. 2014.

[118] S. Costa, H. Lehmann, K. Dautenhahn, B. Robins and F. Soares, 'Using a Humanoid Robot to Elicit Body Awareness and Appropriate Physical Interaction in Children with Autism,' en, *International Journal of Social Robotics*, vol. 7, no. 2, pp. 265–278, Apr. 2015.

[119] C. Hieida, K. Abe, M. Attamimi, T. Shimotomai, T. Nagai and T. Omori, 'Physical embodied communication between robots and children: An approach for relationship building by holding hands,' in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 3291–3298.

[120] H. Fukuda, M. Shiomi, K. Nakagawa and K. Ueda, ''Midas touch' in human-robot interaction: Evidence from event-related potentials during the ultimatum game,' in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, ser. HRI '12, New York, NY, USA: Association for Computing Machinery, Mar. 2012, pp. 131–132.

[121] A. Haans and W. A. IJsselsteijn, 'The Virtual Midas Touch: Helping Behavior After a Mediated Social Touch,' *IEEE Transactions on Haptics*, vol. 2, no. 3, pp. 136–140, Jul. 2009.

[122] C. Bevan and D. Stanton Fraser, 'Shaking Hands and Cooperation in Tele-present Human-Robot Negotiation,' in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '15, New York, NY, USA: Association for Computing Machinery, Mar. 2015, pp. 247–254.

[123] K. Kuwamura, K. Sakai, T. Minato, S. Nishio and H. Ishiguro, 'Hugvie: A medium that fosters love,' in *2013 IEEE RO-MAN*, Aug. 2013, pp. 70–75.

[124] J. Nakanishi, K. Kuwamura, T. Minato, S. Nishio and H. Ishiguro, 'Evoking affection for a communication partner by a robotic communication medium,' vol. 3, 2013, pp. 1–4.

[125] H. Takahashi, M. Ban, H. Osawa, J. Nakanishi, H. Sumioka and H. Ishiguro, 'Huggable Communication Medium Maintains Level of Trust during Conversation Game,' *Frontiers in Psychology*, vol. 8, 2017.

[126] J. Nakanishi, H. Sumioka and H. Ishiguro, 'Virtual Hug Induces Modulated Impression on Hearsay Information,' in *Proceedings of the 6th International Conference on Human-Agent Interaction*, ser. HAI '18, New York, NY, USA: Association for Computing Machinery, 2018, pp. 199–204.

[127] ——, 'A huggable communication medium can provide sustained listening support for special needs students in a classroom,' en, *Computers in Human Behavior*, vol. 93, pp. 106–113, Apr. 2019.

[128] J. Li, 'The benefit of being physically present,' *International Journal of Human-Computer Studies*, vol. 77, no. NA, pp. 23–37, 2015.

[129] K. Wada and T. Shibata, 'Robot therapy in a care house - its sociopsychological and physiological effects on the residents,' in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 3966–3971.

[130] H. Sumioka, A. Nakae, R. Kanai and H. Ishiguro, 'Huggable communication medium decreases cortisol levels,' en, *Scientific Reports*, vol. 3, no. 1, p. 3034, Oct. 2013.

[131] J. A. Russell, 'Affective space is bipolar,' *Journal of Personality and Social Psychology*, vol. 37, pp. 345–356, 1979.

[132] ——, 'A circumplex model of affect,' *Journal of Personality and Social Psychology*, vol. 39, pp. 1161–1178, 1980.

[133] A. Saarinen, V. Harjunen, I. Jasinskaja-Lahti, I. P. Jääskeläinen and N. Ravaja, 'Social touch experience in different contexts: A review,' en, *Neuroscience & Biobehavioral Reviews*, vol. 131, pp. 360–372, Dec. 2021.

[134] G. V. Portnova, E. V. Proskurnina, S. V. Sokolova, I. V. Skorokhodov and A. A. Varlamov, 'Perceived pleasantness of gentle touch in healthy individuals is related to salivary oxytocin response and EEG markers of arousal,' en, *Experimental Brain Research*, vol. 238, no. 10, pp. 2257–2268, Oct. 2020.

[135] J. E. Lee and K. E. Cichy, 'Complex Role of Touch in Social Relationships for Older Adults' Cardiovascular Disease Risk,' en, *Research on Aging*, vol. 42, no. 7-8, pp. 208–216, Aug. 2020.

[136] J. T. Suvilehto, E. Glerean, R. I. M. Dunbar, R. Hari and L. Nummenmaa, 'Topography of social touching depends on emotional bonds between humans,' *Proceedings of the National Academy of Sciences*, vol. 112, no. 45, pp. 13 811–13 816, Nov. 2015.

[137] J. T. Suvilehto *et al.*, 'Cross-cultural similarity in relationship-specific social touching,' *Proceedings of the Royal Society B: Biological Sciences*, vol. 286, no. 1901, p. 20 190 467, Apr. 2019.

[138] J. Sequeira, P. Lima, A. Saffiotti, V. Gonzalez-Pacheco and M. A. Salichs, 'MOnarCH: Multi-robot cognitive systems operating in hospitals,' 2013.

[139] M. Salichs *et al.*, 'Maggie: A Robotic Platform for Human-Robot Social Interaction,' in *2006 IEEE Conference on Robotics, Automation and Mechatronics*, IEEE, Ed., Bangkok: IEEE, Dec. 2006, pp. 1–7.

[140] J. Messias *et al.*, 'A robotic platform for edutainment activities in a pediatric hospital,' in *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, May 2014, pp. 193–198.

[141] M. Maroto-Gómez, Á. Castro-González, J. C. Castillo, M. Malfaz and M. A. Salichs, 'A Bio-inspired Motivational Decision Making System for Social Robots Based on the Perception of the User,' en, *Sensors*, vol. 18, no. 8, p. 2691, Aug. 2018.

[142] R. Z. Marques, L. R. Coutinho, T. B. Borchartt, S. B. Vale and F. J. Silva, 'An Experimental Evaluation of Data Mining Algorithms Using Hyperparameter Optimization,' in *2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI)*, Oct. 2015, pp. 152–156.

[143] E. Fernández-Rodicio, Á. Castro-González, F. Alonso-Martín, M. Maroto-Gómez and M. Á. Salichs, 'Modelling Multimodal Dialogues for Social Robots Using Communicative Acts,' en, *Sensors*, vol. 20, no. 12, p. 3440, Jan. 2020.

[144] F. R. Allen, E. Ambikairajah, N. H. Lovell and B. G. Celler, 'Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models,' eng, *Physiological Measurement*, vol. 27, no. 10, pp. 935–951, Oct. 2006.

[145] D. Anguita, A. Ghio, L. Oneto, X. Parra and J. L. Reyes-Ortiz, 'Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine,' en, in *Ambient Assisted Living and Home Care*, J. Bravo, R. Hervás and M. Rodríguez, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2012, pp. 216–223.

[146] D. Morris, T. S. Saponas, A. Guillory and I. Kelner, 'RecoFit: Using a wearable sensor to find, recognize, and count repetitive exercises,' in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14, New York, NY, USA: Association for Computing Machinery, Apr. 2014, pp. 3225–3234.

[147] K. Kumari, P. H. Chandankhede and A. S. Titarmare, 'Design of Human Activity Recognition System Using Body Sensor Networks,' in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2021, pp. 1011–1016.

[148] M. Muaaz, A. Chelli, M. W. Gerdes and M. Pätzold, 'Wi-Sense: A passive human activity recognition system using Wi-Fi and convolutional neural network and its integration in health information systems,' en, *Annals of Telecommunications*, Jul. 2021.

[149] D. Garcia-Gonzalez, D. Rivero, E. Fernandez-Blanco and M. R. Luaces, 'A Public Domain Dataset for Real-Life Human Activity Recognition Using Smartphone Sensors,' en, *Sensors*, vol. 20, no. 8, p. 2200, Jan. 2020.

[150] F. Alonso-Martin, M. Malfaz, J. Sequeira, J. Gorostiza and M. A. Salichs, 'A Multimodal Emotion Detection System during Human-Robot Interaction,' *Sensors*, vol. 13, no. 11, pp. 15 549–15 581, 2013.

[151] F. Alonso-Martin, Á. Castro-González, J. F. Gorostiza and M. A. Salichs, 'Multidomain Voice Activity Detection during Human-Robot Interaction,' en, in *Social Robotics*, G. Herrmann, M. J. Pearson, A. Lenz, P. Bremner, A. Spiers and U. Leonards, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2013, pp. 64–73.

[152] F. Alonso-Martín, A. Ramey and M. A. Salichs, 'Speaker Identification using Three Signal Voice Domains during Human-Robot Interaction : [Late Breaking Reports],' in *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2014, pp. 114–115.

[153] O. M. Essenwanger, *Elements of statistical analysis*, eng, ser. World survey of climatology. Amsterdam: Elsevier, 1986.

[154] P. S. Gopalakrishnan, 'Current Methods in Continuous Speech Recognition,' en, in *Modern Methods of Speech Processing*, ser. The Springer International Series in Engineering and Computer Science, R. P. Ramachandran and R. J. Mammone, Eds., Boston, MA: Springer US, 1995, pp. 185–212.

[155] S. V. Vaseghi, 'Spectral Subtraction,' en, in *Advanced Signal Processing and Digital Noise Reduction*, S. V. Vaseghi, Ed., Wiesbaden: Vieweg+Teubner Verlag, 1996, pp. 242–260.

[156] W. Cochran *et al.*, 'What is the fast Fourier transform?' *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1664–1674, Oct. 1967.

[157] R. S. Stanković and B. J. Falkowski, 'The Haar wavelet transform: Its status and achievements,' en, *Computers & Electrical Engineering*, vol. 29, no. 1, pp. 25–44, Jan. 2003.

[158] G. Holmes, A. Donkin and I. Witten, 'WEKA: A machine learning workbench,' in *Proceedings of ANZIIS '94 - Australian New Zealnd Intelligent Information Systems Conference*, Nov. 1994, pp. 357–361.

[159] F. Pedregosa *et al.*, 'Scikit-learn: Machine Learning in Python,' *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[160] S. Argentieri, A. Portello, M. Bernard, P. Danès and B. Gas, 'Binaural Systems in Robotics,' en, in *The Technology of Binaural Listening*, ser. Modern Acoustics and Signal Processing, J. Blauert, Ed., Berlin, Heidelberg: Springer, 2013, pp. 225–253.

[161] S. Argentieri, P. Danès and P. Souères, 'A survey on sound source localization in robotics: From binaural to array processing methods,' en, *Computer Speech & Language*, vol. 34, no. 1, pp. 87–112, Nov. 2015.

[162] K. Nakadai and K. Nakamura, 'Sound Source Localization and Separation,' en, in *Wiley Encyclopedia of Electrical and Electronics Engineering*, John Wiley & Sons, Ltd, 2015, pp. 1–18.

[163] M. Basiri, F. Schill, P. U.Lima and D. Floreano, 'Localization of emergency acoustic sources by micro aerial vehicles,' en, *Journal of Field Robotics*, vol. 35, no. 2, pp. 187–201, 2018.

[164] I. Meza, C. Rascon, G. Fuentes and L. A. Pineda, 'On Indexicality, Direction of Arrival of Sound Sources, and Human-Robot Interaction,' en, *Journal of Robotics*, vol. 2016, e3081048, May 2016.

[165] J. C. Murray, S. Wermter and H. R. Erwin, 'Bioinspired Auditory Sound Localisation for Improving the Signal to Noise Ratio of Socially Interactive Robots,' in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 1206–1211.

[166] H. Knight, R. Toscano, W. D. Stiehl, A. Chang, Y. Wang and C. Breazeal, 'Real-time social touch gesture recognition for sensate robots,' in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 3715–3720.

[167] J. Chang, K. MacLean and S. Yohanan, 'Gesture Recognition in the Haptic Creature,' en, in *Haptics: Generating and Perceiving Tangible Sensations*, A. M. L. Kappers, J. B. F. van Erp, W. M. Bergmann Tiest and F. C. T. van der Helm, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2010, pp. 385–391.

[168] R. Andreasson, B. Alenljung, E. Billing and R. Lowe, *Affective Touch in Human–Robot Interaction: Conveying Emotion to the Nao Robot*, en-GB, 2018.

[169] C. Harrison, J. Schwarz and S. E. Hudson, 'TapSense: Enhancing finger interaction on touch surfaces,' in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ser. UIST '11, New York, NY, USA: Association for Computing Machinery, Oct. 2011, pp. 627–636.

[170] M. Goel *et al.*, 'SurfaceLink: Using inertial and acoustic sensing to enable multi-device interaction on a surface,' in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14, New York, NY, USA: Association for Computing Machinery, Apr. 2014, pp. 1387–1396.

[171] M. Cobos, F. Antonacci, A. Alexandridis, A. Mouchtaris and B. Lee, 'A Survey of Sound Source Localization Methods in Wireless Acoustic Sensor Networks,' en, *Wireless Communications and Mobile Computing*, vol. 2017, e3956282, Aug. 2017.

[172] W. T. Thomson, 'Transmission of Elastic Waves through a Stratified Solid Medium,' *Journal of Applied Physics*, vol. 21, no. 2, pp. 89–93, 1950.

[173] A. K. Pandey and R. Gelin, 'A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind,' *IEEE Robotics & Automation Magazine*, vol. 25, no. 3, pp. 40–48, Sep. 2018.

[174] S. Michieletto, D. Zanin and E. Menegatti, 'NAO Robot Simulation for Service Robotics Purposes,' in *2013 European Modelling Symposium*, Nov. 2013, pp. 477–482.

[175] C. Bielza, G. Li and P. Larrañaga, 'Multi-dimensional classification with Bayesian networks,' en, *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 705–727, Sep. 2011.

[176] R. E. Schapire and Y. Singer, 'Improved Boosting Algorithms Using Confidence-rated Predictions,' en, *Machine Learning*, vol. 37, no. 3, pp. 297–336, Dec. 1999.

[177] ——, 'BoosTexter: A Boosting-based System for Text Categorization,' en, *Machine Learning*, vol. 39, no. 2, pp. 135–168, May 2000.

[178] A. Appice and S. Džeroski, 'Stepwise Induction of Multi-target Model Trees,' in *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings*, J. N. Kok, J. Koronacki, R. L. d. Mantaras, S. Matwin, D. Mladenič and A. Skowron, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 502–509.

[179] J. Read, P. Reutemann, B. Pfahringer and G. Holmes, 'Meka: A multi-label/multi-target extension to weka,' *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 667–671, 2016.

[180] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek and I. Vlahavas, 'MULAN: A Java Library for Multi-Label Learning,' *Journal of Machine Learning Research*, vol. 12, no. 71, pp. 2411–2414, 2011.

[181] P. Szymański and T. Kajdanowicz, *A scikit-based Python environment for performing multi-label classification*, Dec. 2018.

[182] C. Rascon and I. Meza, 'Localization of sound sources in robotics: A review,' en, *Robotics and Autonomous Systems*, vol. 96, pp. 184–210, Oct. 2017.

[183] D. Di Carlo, A. Deleforge and N. Bertin, 'Mirage: 2D Source Localization Using Microphone Pair Augmentation with Echoes,' *arXiv:1906.08968 [physics]*, Jun. 2019.

[184] S. Pan *et al.*, 'SurfaceVibe: Vibration-based tap &amp; swipe tracking on ubiquitous surfaces,' in *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '17, New York, NY, USA: Association for Computing Machinery, Apr. 2017, pp. 197–208.

[185] W. Meng and W. Xiao, 'Energy-Based Acoustic Source Localization Methods: A Survey,' en, *Sensors*, vol. 17, no. 2, p. 376, Feb. 2017.

[186] C. Knapp and G. Carter, 'The generalized correlation method for estimation of time delay,' *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[187] C. Pang, H. Liu, J. Zhang and X. Li, 'Binaural Sound Localization Based on Reverberation Weighting and Generalized Parametric Mapping,' *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 8, pp. 1618–1632, Aug. 2017.

[188] Q. Zhang, H. Abeida, M. Xue, W. Rowe and J. Li, 'Fast implementation of sparse iterative covariance-based estimation for source localization,' *The Journal of the Acoustical Society of America*, vol. 131, no. 2, pp. 1249–1259, Feb. 2012.

[189] J. H. DiBiase, 'A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays,' en, p. 122,

[190] D. Diaz-Guerra, A. Miguel and J. R. Beltran, 'Robust Sound Source Tracking Using SRP-PHAT and 3D Convolutional Neural Networks,' *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 300–311, 2021.

[191] E. Sejdić, I. Djurović and J. Jiang, 'Time–frequency feature representation using energy concentration: An overview of recent advances,' en, *Digital Signal Processing*, vol. 19, no. 1, pp. 153–183, Jan. 2009.

[192] P. Svaizer, M. Matassoni and M. Omologo, 'Acoustic source location in a three-dimensional space using crosspower spectrum phase,' in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Apr. 1997, 231–234 vol.1.

[193] H. Do, H. F. Silverman and Y. Yu, 'A Real-Time SRP-PHAT Source Location Implementation using Stochastic Region Contraction(SRC) on a Large-Aperture Microphone Array,' in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 1, Apr. 2007, pp. I–121–I–124.

[194] M. Basiri, F. Schill, P. Lima and D. Floreano, 'On-Board Relative Bearing Estimation for Teams of Drones Using Sound,' *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 820–827, Jul. 2016.

[195] J. Read, B. Pfahringer, G. Holmes and E. Frank, 'Classifier chains for multi-label classification,' en, *Machine Learning*, vol. 85, no. 3, pp. 333–359, Dec. 2011.

[196] B. Bashari Rad, H. Bhatti and M. Ahmadi, 'An Introduction to Docker and Analysis of its Performance,' *IJCSNS International Journal of Computer Science and Network Security*, vol. 173, p. 8, Mar. 2017.

[197] V. Gonzalez-Pacheco, A. Ramey, F. Alonso-Martin, A. Castro-Gonzalez and M. A. Salichs, 'Maggie: A Social Robot as a Gaming Platform,' *International Journal of Social Robotics*, vol. 3, no. 4, pp. 371–381, Sep. 2011.

[198] D. Tang, B. Yusuf, J. Botzheim, N. Kubota and C. S. Chan, 'A novel multimodal communication framework using robot partner for aging population,' en, *Expert Systems with Applications*, vol. 42, no. 9, pp. 4540–4555, Jun. 2015.

[199] R. C. B. Madeo, S. M. Peres and C. A. d. M. Lima, 'Gesture phase segmentation using support vector machines,' en, *Expert Systems with Applications*, vol. 56, pp. 100–115, Sep. 2016.

[200] Y. Kim, S. Koo, J. G. Lim and D. Kwon, 'A robust online touch pattern recognition for dynamic human-robot interaction,' *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1979–1987, Aug. 2010.

[201] C. Van Rijsbergen, *Information Retrieval*. Butterworths, 1979.

[202] A. Kent, M. M. Berry, F. U. Luehrs Jr. and J. W. Perry, 'Machine literature searching VIII. Operational criteria for designing information retrieval systems,' en, *American Documentation*, vol. 6, no. 2, pp. 93–101, 1955.

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

[204] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY: Springer, 2009.

[205] R. Caruana, N. Karampatziakis and A. Yessenalina, 'An empirical evaluation of supervised learning in high dimensions,' in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08, New York, NY, USA: Association for Computing Machinery, Jul. 2008, pp. 96–103.

[206] J. Platt, 'Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines,' Microsoft, Tech. Rep. MSR-TR-98-14, Apr. 1998.

[207] M. Fernández-Delgado, E. Cernadas, S. Barro and D. Amorim, 'Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?' *Journal of Machine Learning Research*, vol. 15, no. 90, pp. 3133–3181, 2014.

[208] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald and E. Muharemagic, 'Deep learning applications and challenges in big data analytics,' *Journal of Big Data*, vol. 2, no. 1, p. 1, Feb. 2015.

[209] D. Palaz, M. Magimai.-Doss and R. Collobert, 'Analysis of CNN-based Speech Recognition System using Raw Speech as Input,' in *Proceedings of Interspeech*, Dresden: ISCA, 2015, pp. 11–15.

[210] S. Lawrence, C. Giles, A. C. Tsoi and A. Back, 'Face recognition: A convolutional neural-network approach,' *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[211] A. Krizhevsky, I. Sutskever and G. E. Hinton, 'ImageNet Classification with Deep Convolutional Neural Networks,' in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012.

[212] M. S. Sorower, 'A literature survey on algorithms for multi-label learning,' *Oregon State University, Corvallis*, vol. 18, 2010.

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

[214] J. Rosa and M. Basiri, 'Cooperative Audio-Visual System for Localizing Small Aerial Robots,' in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 6064–6069.

[215] S. Roweis, 'EM Algorithms for PCA and SPCA,' in *Advances in Neural Information Processing Systems*, vol. 10, MIT Press, 1997.

[216] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, 'SMOTE: Synthetic minority over-sampling technique,' *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, Jun. 2002.

[217] R. Beale and C. Peter, 'The Role of Affect and Emotion in HCI,' en, in *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*, ser. Lecture Notes in Computer Science, C. Peter and R. Beale, Eds., Berlin, Heidelberg: Springer, 2008, pp. 1–11.

[218] A. Henschel, G. Laban and E. S. Cross, 'What Makes a Robot Social? A Review of Social Robots from Science Fiction to a Home or Hospital Near You,' en, *Current Robotics Reports*, vol. 2, no. 1, pp. 9–19, Mar. 2021.

[219] E. K. Diekhof, H. E. Kipshagen, P. Falkai, P. Dechent, J. Baudewig and O. Gruber, 'The power of imagination–how anticipatory mental imagery alters perceptual processing of fearful facial expressions,' eng, *NeuroImage*, vol. 54, no. 2, pp. 1703–1714, Jan. 2011.

[220] J. Redondo, I. Fraga, I. Padrón and A. Piñeiro, 'Affective ratings of sound stimuli,' en, *Behavior Research Methods*, vol. 40, no. 3, pp. 784–790, Aug. 2008.

[221] M. Vasconcelos, M. Dias, A. P. Soares and A. P. Pinheiro, 'What is the Melody of That Voice? Probing Unbiased Recognition Accuracy with the Montreal Affective Voices,' en, *Journal of Nonverbal Behavior*, vol. 41, no. 3, pp. 239–267, Sep. 2017.

[222] M. Y. Tsalamlal, M. Amorim, J. Martin and M. Ammi, 'Combining Facial Expression and Touch for Perceiving Emotional Valence,' *IEEE Transactions on Affective Computing*, vol. 9, no. 4, pp. 437–449, Oct. 2018.

[223] Y. Huang, J. Yang, S. Liu and J. Pan, 'Combining Facial Expressions and Electroencephalography to Enhance Emotion Recognition,' en, *Future Internet*, vol. 11, no. 5, p. 105, May 2019.

[224] C. Breazeal and L. Aryananda, 'Recognition of Affective Communicative Intent in Robot-Directed Speech,' en, *Autonomous Robots*, vol. 12, no. 1, pp. 83–104, Jan. 2002.

[225] J. C. Castillo, A. Fernández-Caballero, Á. Castro-González, M. A. Salichs and M. T. López, 'A Framework for Recognizing and Regulating Emotions in the Elderly,' en, in *Ambient Assisted Living and Daily Activities*, L. Pecchia, L. L. Chen, C. Nugent and J. Bravo, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 320–327.

[226] A. Fernández-Caballero *et al.*, 'Smart environment architecture for emotion detection and regulation,' en, *Journal of Biomedical Informatics*, vol. 64, pp. 55–73, Dec. 2016.

[227] J. C. Castillo *et al.*, 'Software Architecture for Smart Emotion Recognition and Regulation of the Ageing Adult,' en, *Cognitive Computation*, vol. 8, no. 2, pp. 357–367, Apr. 2016.

[228] I. Ahmed, V. J. Harjunen, G. Jacucci, N. Ravaja, T. Ruotsalo and M. Spape, 'Touching virtual humans: Haptic responses reveal the emotional impact of affective agents,' *IEEE Transactions on Affective Computing*, pp. 1–1, 2020.

[229] P. Ekman, 'Basic emotions,' *Handbook of cognition and emotion*, vol. 98, no. 45-60, p. 16, 1999.

[230] M. G. Calvo and D. Lundqvist, 'Facial expressions of emotion (KDEF): Identification under different display-duration conditions,' en, *Behavior Research Methods*, vol. 40, no. 1, pp. 109–115, Feb. 2008.

[231] S. S. Shapiro and M. B. Wilk, 'An Analysis of Variance Test for Normality (Complete Samples),' *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[232] S. Gobron, J. Ahn, G. Paltoglou, M. Thelwall and D. Thalmann, 'From sentence to emotion: A real-time three-dimensional graphics metaphor of emotions extracted from text,' en, *The Visual Computer*, vol. 26, no. 6, pp. 505–519, Jun. 2010.

[233] G. Paltoglou and M. Thelwall, 'Seeing Stars of Valence and Arousal in Blog Posts,' *IEEE Transactions on Affective Computing*, vol. 4, no. 1, pp. 116–123, Jan. 2013.

[234] M. Donnermann *et al.*, 'Social robots and gamification for technology supported learning: An empirical study on engagement and motivation,' en, *Computers in Human Behavior*, vol. 121, p. 106 792, Aug. 2021.

[235] A. Tapus, M. J. Mataric and B. Scassellati, 'Socially assistive robotics [Grand Challenges of Robotics],' *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 35–42, Mar. 2007.

[236] A. E. Raake Sebastian, *Quality of Experience - Quality and Quality of Experience*. 2014, vol. NA.

[237] K. u. R. Laghari, N. Crespi, B. Molina and C. Palau, 'QoE Aware Service Delivery in Distributed Environment,' in *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, Mar. 2011, pp. 837–842.

[238] K. U. Rehman Laghari and K. Connelly, 'Toward total quality of experience: A QoE model in a communication ecosystem,' *IEEE Communications Magazine*, vol. 50, no. 4, pp. 58–65, Apr. 2012.

[239] H. L. O'Brien and E. G. Toms, 'What is user engagement? A conceptual framework for defining user engagement with technology,' en, *Journal of the American Society for Information Science and Technology*, vol. 59, no. 6, pp. 938–955, 2008.

[240] E. L. Deci, R. Koestner and R. M. Ryan, 'Extrinsic Rewards and Intrinsic Motivation in Education: Reconsidered Once Again,' en, *Review of Educational Research*, vol. 71, no. 1, pp. 1–27, Mar. 2001.

[241] G. Tisza, S. Zhu and P. Markopoulos, 'Fun to Enhance Learning, Motivation, Self-efficacy, and Intention to Play in DGBL,' en, in *Entertainment Computing – ICEC 2021*, J. Baalsrud Hauge, J. C. S. Cardoso, L. Roque and P. A. Gonzalez-Calero, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 28–45.

[242] H. O'Brien and P. Cairns, *Why Engagement Matters: Cross-Disciplinary Perspectives of User Engagement in Digital Media*, en. Springer, May 2016.

[243] H. L. O'Brien and E. G. Toms, 'The development and evaluation of a survey to measure user engagement,' en, *Journal of the American Society for Information Science and Technology*, vol. 61, no. 1, pp. 50–69, 2010.

[244] H. L. O'Brien, P. Cairns and M. Hall, 'A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form,' en, *International Journal of Human-Computer Studies*, vol. 112, pp. 28–39, Apr. 2018.

[245] J. Hall, T. Tritton, A. Rowe, A. Pipe, C. Melhuish and U. Leonards, 'Perception of own and robot engagement in human–robot interactions and their dependence on robotics knowledge,' en, *Robotics and Autonomous Systems*, Advances in Autonomous Robotics — Selected extended papers of the joint 2012 TAROS Conference and the FIRA RoboWorld Congress, Bristol, UK, vol. 62, no. 3, pp. 392–399, Mar. 2014.

[246] C. Rich, B. Ponsler, A. Holroyd and C. L. Sidner, 'Recognizing engagement in human-robot interaction,' in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2010, pp. 375–382.

[247] C. L. Sidner, C. D. Kidd, C. Lee and N. Lesh, 'Where to look: A study of human-robot engagement,' in *Proceedings of the 9th international conference on Intelligent user interfaces*, ser. IUI '04, New York, NY, USA: Association for Computing Machinery, 2004, pp. 78–84.

[248] C. L. Sidner, C. Lee, C. D. Kidd, N. Lesh and C. Rich, 'Explorations in engagement for humans and robots,' en, *Artificial Intelligence*, vol. 166, no. 1, pp. 140–164, Aug. 2005.

[249] R. Ishii and Y. I. Nakano, 'Estimating User's Conversational Engagement Based on Gaze Behaviors,' en, in *Intelligent Virtual Agents*, H. Prendinger, J. Lester and M. Ishizuka, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2008, pp. 200–207.

[250] S. Ivaldi, S. Anzalone, W. Rousseau, O. Sigaud and M. Chetouani, 'Robot initiative in a team learning task increases the rhythm of interaction but not the perceived engagement,' *Frontiers in Neurorobotics*, vol. 8, 2014.

[251] J. Sanghvi, G. Castellano, I. Leite, A. Pereira, P. W. McOwan and A. Paiva, 'Automatic analysis of affective postures and body motion to detect engagement with a game companion,' in *Proceedings of the 6th international conference on Human-robot interaction*, ser. HRI '11, New York, NY, USA: Association for Computing Machinery, Mar. 2011, pp. 305–312.

[252] S. Boucenna *et al.*, 'Interactive Technologies for Autistic Children: A Review,' en, *Cognitive Computation*, vol. 6, no. 4, pp. 722–740, Dec. 2014.

[253] S. Boucenna, S. Anzalone, E. Tilmont, D. Cohen and M. Chetouani, 'Learning of Social Signatures Through Imitation Game Between a Robot and a Human Partner,' *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 3, pp. 213–225, Sep. 2014.

[254] B. Scassellati, 'Quantitative metrics of social response for autism diagnosis,' in *RO-MAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, Aug. 2005, pp. 585–590.

[255] B. Scassellati, 'How Social Robots Will Help Us to Diagnose, Treat, and Understand Autism,' en, in *Robotics Research*, S. Thrun, R. Brooks and H. Durrant-Whyte, Eds., ser. Springer Tracts in Advanced Robotics, Berlin, Heidelberg: Springer, 2007, pp. 552–563.

[256] S. M. Anzalone, S. Boucenna, S. Ivaldi and M. Chetouani, 'Evaluating the Engagement with Social Robots,' en, *International Journal of Social Robotics*, vol. 7, no. 4, pp. 465–478, Aug. 2015.

[257] M. Lalmas, H. O'Brien and E. Yom-Tov, *Measuring User Engagement*, en, ser. Synthesis Lectures on Information Concepts, Retrieval, and Services. Cham: Springer International Publishing, 2015.

[258] M. Saerbeck, T. Schut, C. Bartneck and M. D. Janse, 'Expressive robots in education: Varying the degree of social supportive behavior of a robotic tutor,' in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10, New York, NY, USA: Association for Computing Machinery, 2010, pp. 1613–1622.

[259] S. Lee *et al.*, 'On the effectiveness of Robot-Assisted Language Learning,' *ReCALL*, vol. 23, no. 1, pp. 25–58, 2011.

[260] A. Deublein, A. Pfeifer, K. Merbach, K. Bruckner, C. Mengelkamp and B. Lugrin, 'Scaffolding of motivation in learning using a social robot,' en, *Computers & Education*, vol. 125, pp. 182–190, Oct. 2018.

[261] S. Zörner *et al.*, 'An Immersive Investment Game to Study Human-Robot Trust,' *Frontiers in Robotics and AI*, vol. 8, 2021.

[262] R. Vallerand, L. Pelletier, M. Blais, N. Brière, C. Senécal and E. Vallieres, 'The Academic Motivation Scale: A Measure of Intrinsic, Extrinsic, and Amotivation in Education,' *Educational and Psychological Measurement*, vol. 52, pp. 1003–1003, Dec. 1992.

[263] E. L. Deci and R. M. Ryan, 'Self-determination theory,' in *Handbook of theories of social psychology, Vol. 1*, Thousand Oaks, CA: Sage Publications Ltd, 2012, pp. 416–436.

[264] R. M. Ryan, 'Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory,' *Journal of Personality and Social Psychology*, vol. 43, pp. 450–461, 1982.

[265] R. M. Ryan, R. Koestner and E. L. Deci, 'Ego-involved persistence: When free-choice behavior is not intrinsically motivated,' en, *Motivation and Emotion*, vol. 15, no. 3, pp. 185–205, Sep. 1991.

[266] E. L. Deci, H. Eghrari, B. C. Patrick and D. R. Leone, 'Facilitating internalization: The self-determination theory perspective,' *Journal of personality*, vol. 62, no. 1, pp. 119–142, 1994.

[267] K. S. Ostrow and N. T. Heffernan, 'Testing the Validity and Reliability of Intrinsic Motivation Inventory Subscales Within ASSISTments,' en, in *Artificial Intelligence in Education*, C. Penstein Rosé *et al.*, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 381–394.

[268] G. Tisza and P. Markopoulos, 'FunQ: Measuring the fun experience of a learning activity with adolescents,' en, *Current Psychology*, Mar. 2021.

[269] T. W. Malone and M. R. Lepper, 'Making Learning Fun: A Taxonomy of Intrinsic Motivations for Learning,' in *Aptitude, Learning, and Instruction*, Routledge, 1987.

[270] C. Bisson and J. Luckner, 'Fun in Learning: The Pedagogical Role of Fun in Adventure Education,' en, *Journal of Experiential Education*, vol. 19, no. 2, pp. 108–112, Aug. 1996.

[271] J. C. Read, 'Validating the Fun Toolkit: An instrument for measuring children's opinions of technology,' en, *Cognition, Technology & Work*, vol. 10, no. 2, pp. 119–128, Apr. 2008.

[272] A. Tasci and Y. J. Ko, 'A Fun-Scale for Understanding the Hedonic Value of a Product: The Destination Context,' *Journal of Travel & Tourism Marketing*, vol. 33, pp. 1–22, May 2015.

[273] M. Austin, *Music Video Games: Performance, Politics, and Play*, en. Bloomsbury Publishing USA, Jul. 2016.

[274] E. B. Mackamul and A. Esteves, 'A Look at the Effects of Handheld and Projected Augmented-reality on a Collaborative Task,' in *Proceedings of the Symposium on Spatial User Interaction*, ser. SUI '18, New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 74–78.

[275] R. Speakman, M. M. Hall and D. Walsh, 'User Engagement with Generous Interfaces for Digital Cultural Heritage,' en, in *Digital Libraries for Open Knowledge*, E. Méndez, F. Crestani, C. Ribeiro, G. David and J. C. Lopes, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 186–191.

[276] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi and Z. Lin, 'When Machine Learning Meets Privacy: A Survey and Outlook,' *ACM Computing Surveys*, vol. 54, no. 2, 31:1–31:36, Mar. 2021.

[277] H. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y. Arcas, 'Communication-Efficient Learning of Deep Networks from Decentralized Data,' *arXiv:1602.05629 [cs]*, Feb. 2017.

[278] A. Galakatos, A. Crotty and T. Kraska, 'Distributed Machine Learning,' en, in *Encyclopedia of Database Systems*, L. Liu and M. T. Özsu, Eds., New York, NY: Springer, 2018, pp. 1196–1201.

[279] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen and J. S. Rellermeyer, 'A Survey on Distributed Machine Learning,' *ACM Computing Surveys*, vol. 53, no. 2, 30:1–30:33, Mar. 2020.

[280] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh and D. Bacon, *Federated Learning: Strategies for Improving Communication Efficiency*, Oct. 2017.

[281] Q. Yang, Y. Liu, T. Chen and Y. Tong, *Federated Machine Learning: Concept and Applications*, Feb. 2019.

[282] L. Li, Y. Fan, M. Tse and K.-Y. Lin, 'A review of applications in federated learning,' en, *Computers & Industrial Engineering*, vol. 149, p. 106 854, Nov. 2020.

[283] M. Aledhari, R. Razzak, R. M. Parizi and F. Saeed, 'Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications,' *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[284] Y. LeCun, Y. Bengio and G. Hinton, 'Deep learning,' en, *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[285] X. Lu, Y. Liao, P. Lio and P. Hui, 'Privacy-Preserving Asynchronous Federated Learning Mechanism for Edge Network Computing,' *IEEE Access*, vol. 8, pp. 48 970–48 981, 2020.

[286] Y. Lu, X. Huang, Y. Dai, S. Maharjan and Y. Zhang, 'Differentially Private Asynchronous Federated Learning for Mobile Edge Computing in Urban Informatics,' *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.

[287] I. Kholod *et al.*, 'Open-Source Federated Learning Frameworks for IoT: A Comparative Review and Analysis,' en, *Sensors*, vol. 21, no. 1, p. 167, Jan. 2021.

[288] Martín Abadi *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015.

[289] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[290] S. Hernández Juan and F. Herrero Cotarelo, 'Multi-master ROS systems,' eng, 2015.

[291] Y. Wang, Y. Tong and D. Shi, 'Federated Latent Dirichlet Allocation: A Local Differential Privacy Based Framework,' en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 6283–6290, Apr. 2020.

[292] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin and V. Chandra, 'Federated Learning with Non-IID Data,' *arXiv:1806.00582 [cs, stat]*, Jun. 2018.

[293] H. Zhu, J. Xu, S. Liu and Y. Jin, 'Federated Learning on Non-IID Data: A Survey,' *arXiv:2106.06843 [cs]*, Jun. 2021.

[294] N. Shoham *et al.*, 'Overcoming Forgetting in Federated Learning on Non-IID Data,' *arXiv:1910.07796 [cs, stat]*, Oct. 2019.

[295] C. Briggs, Z. Fan and P. Andras, 'Federated learning with hierarchical clustering of local updates to improve training on non-IID data,' *arXiv:2004.11791 [cs, stat]*, May 2020.

[296] F. Sattler, S. Wiedemann, K.-R. Müller and W. Samek, ‘Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data,’ *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[297] J. S. Ng *et al.*, ‘A Hierarchical Incentive Design Toward Motivating Participation in Coded Federated Learning,’ *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 359–375, Jan. 2022.

[298] W. Y. B. Lim *et al.*, ‘Hierarchical Incentive Mechanism Design for Federated Machine Learning in Mobile Networks,’ *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9575–9588, Oct. 2020.

[299] X. Wu, Z. Liang and J. Wang, ‘FedMed: A Federated Learning Framework for Language Modeling,’ en, *Sensors*, vol. 20, no. 14, p. 4048, Jan. 2020.

[300] M. Chen *et al.*, ‘Federated Learning of N-Gram Language Models,’ in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 121–130.

[301] X. Ouyang, Z. Xie, J. Zhou, J. Huang and G. Xing, ‘ClusterFL: A similarity-aware federated learning system for human activity recognition,’ in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’21, New York, NY, USA: Association for Computing Machinery, Jun. 2021, pp. 54–66.

[302] F. Liu, X. Wu, S. Ge, W. Fan and Y. Zou, ‘Federated Learning for Vision-and-Language Grounding Problems,’ en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 11 572–11 579, Apr. 2020.

[303] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck and S. Srikanteswara, ‘Energy Demand Prediction with Federated Learning for Electric Vehicle Networks,’ in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6.

[304] B. Liu, L. Wang, M. Liu and C.-Z. Xu, ‘Federated Imitation Learning: A Novel Framework for Cloud Robotic Systems With Heterogeneous Sensor Data,’ *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3509–3516, Apr. 2020.

[305] S. Yohanan and K. E. MacLean, ‘The Haptic Creature Project : Social Human-Robot Interaction through Affective Touch,’ 2008.

[306] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam and C. Ré, ‘Data Programming: Creating Large Training Sets, Quickly,’ in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016.

[307] P. Nodet, V. Lemaire, A. Bondu, A. Cornuéjols and A. Ouorou, 'From Weakly Supervised Learning to Biquality Learning: An Introduction,' in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, pp. 1–10.

[308] X. Zhu and A. B. Goldberg, 'Overview of Semi-Supervised Learning,' en, in *Introduction to Semi-Supervised Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning, X. Zhu and A. B. Goldberg, Eds., Cham: Springer International Publishing, 2009, pp. 9–19.

[309] J. E. van Engelen and H. H. Hoos, 'A survey on semi-supervised learning,' en, *Machine Learning*, vol. 109, no. 2, pp. 373–440, Feb. 2020.

[310] D.-H. Lee, 'Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks,' in *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, vol. 3, Jul. 2013, p. 896.

Appendix 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
MultilayerPerceptronCS	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

Appendix B

Questionnaires

B.1. User Experience Questionnaire - English version

This is the English version of the questionnaire used during our User Experience Experiments, shown in Section 6.2. We have highlighted the items that were similar enough to be merged into a single item using the same colour.

Questionnaire	Dimension	ID	Question
UES-SF	**Focused Attention**	FA-S.1	I lost myself in this experience.
		FA-S.2	**The time I spent using Application X just slipped away.**
		FA-S.3	**I was absorbed in this experience.**
FunQ	**Immersion**	I1	**During the activity, I felt that time flew.**
		I2	**During the activity, I forgot about my surroundings.**
		D9	During the activity, I felt good.
FunQ	**Delight**	D1	**During the activity, I had fun.**
		D2	I want to do something like this again.
		D3	During the activity, I was happy.
IMI	**Interest/ Enjoyment**	E1	I enjoyed doing this activity very much
		E2	**This activity was fun to do.**
		E3	I thought this was a boring activity. (R)
		E4	This activity did not hold my attention at all. (R)
		E5	I would describe this activity as very interesting.
		E6	I thought this activity was quite enjoyable.
		E7	While I was doing this activity, I was thinking about how much I enjoyed it.
FunQ	**Challenge**	C1	**During the activity, I felt I was good at this activity.**
		C2	During the activity, I did something new.
		C3	During the activity, I was curious.
IMI	**Perceived Competence**	C1	**I think I am pretty good at this activity.**
		C2	I think I did pretty well at this activity, compared to other users.
		C3	After working at this activity for awhile, I felt pretty competent.
		C4	I am satisfied with my performance at this task.
		C5	I was pretty skilled at this activity.
		C6	This was an activity that I couldn't do very well. (R)
UES-SF	**Reward Factor**	RW-S.1	Using Application X was worthwhile.
		RW-S.2	My experience was rewarding.
		RW-S.3	I felt interested in this experience.
UES-SF	**Aesthetic Appeal**	AE-S.1	This Application X was attractive.
		AE-S.2	This Application X was aesthetically appealing.
		AE-S.3	This Application X appealed to my senses.
UES-SF	**Perceived Usability**	PU-S.1	I felt frustrated while using this Application X.
		PU-S.2	I found this Application X confusing to use.
		PU-S.3	Using this Application X was taxing.

B.2. User Experience Questionnaire - Spanish version

The following is the final questionnaire after being translated into Spanish, as explained in Section 6.2.3. We followed the next steps to adapt the questionnaire: (i) each of the items used in the final questionnaire was translated into this language, trying to preserve the original meaning of the question in the translation, (ii) backward translation from Spanish, and finally, (iii) comparison of the original and the backward translated English text, solving the discrepancies. For each item in the final form, we used a 5-point Likert scale, ranging from 1-'strongly disagree' to 5-'strongly agree'.

Questionnaire	Dimension	ID	Question
UES-SF	**Focused Attention**	FA-S.1	Me sumergí en esta experiencia.
		FA-S.3	Estuve absorto en esta experiencia.
FunQ	**Immersion**	I1	Durante la actividad, sentí que el tiempo volaba.
		D9	Durante la actividad, me sentí bien.
FunQ	**Delight**	D1	Durante la actividad, me divertí.
		D2	Quiero volver a hacer algo así.
		D3	Durante la actividad, fui feliz.
IMI	**Interest/ Enjoyment**	E1	He disfrutado haciendo esta actividad mucho
		E3	Me pareció una actividad aburrida. (R)
		E4	Esta actividad no mantuvo mi atención en absoluto. (R)
		E5	Yo describiría esta actividad como muy interesante.
		E6	Esta actividad me pareció bastante agradable.
		E7	Mientras estaba haciendo esta actividad, estuve pensando en lo mucho que la disfrutaba.
FunQ	**Challenge**	C2	Durante la actividad, hice algo nuevo.
		C3	Durante la actividad, sentí curiosidad.
IMI	**Perceived Competence**	C1	Creo que soy bastante bueno en esta actividad.
		C2	Creo que en esta actividad lo hice bastante bien, en comparación con otros usuarios
		C3	Después de trabajar en esta actividad durante un tiempo, me sentí bastante competente.
		C4	Estoy satisfecho con mi rendimiento en esta tarea.
		C5	Fuí bastante habilidoso en esta actividad.
		C6	Esta era una actividad que no pude hacer muy bien. (R)
UES-SF	**Reward Factor**	RW-S.1	Usar la aplicación X ha merecido la pena.
		RW-S.2	Mi experiencia fue gratificante.
		RW-S.3	Me sentí interesado en esta experiencia.
UES-SF	**Aesthetic Appeal**	AE-S.1	Esta aplicación X era atractiva.
		AE-S.2	Esta Aplicación X era estéticamente atrayente.
		AE-S.3	Esta Aplicación X atraía a mis sentidos.
UES-SF	**Perceived Usability**	PU-S.1	Me sentí frustrado mientras usaba esta Aplicación X.
		PU-S.2	Encontré esta Aplicación X confusa de usar.
		PU-S.3	Usar esta Aplicación X fue agotador.