



# Techno-regulation and intelligent safeguards

## Analysis of touch gestures for online child protection

Rocco Zaccagnino<sup>1</sup> · Carmine Capo<sup>1</sup> · Alfonso Guarino<sup>1</sup> · Nicola Lettieri<sup>2</sup> · Delfina Malandrino<sup>1</sup>

Received: 9 October 2019 / Revised: 9 September 2020 / Accepted: 22 December 2020 /

Published online: 6 February 2021

© The Author(s) 2021

### Abstract

The growth of Internet and the pervasiveness of ICT have led to a radical change in social relationships. One of the drawbacks of this change is the exposure of individuals to threats during online activities. In this context, the *techno-regulation* paradigm is inspiring new ways to safeguard legally interests by means of tools allowing to hamper breaches of law. In this paper, we focus on the exposure of individuals to specific online threats when interacting with smartphones. We propose a novel techno-regulatory approach exploiting machine learning techniques to provide safeguards against threats online. Specifically, we study a set of touch-based gestures to distinguish between underages or adults who is accessing a smartphone, and so to guarantee protection. To evaluate the proposed approach's effectiveness, we developed an Android app to build a dataset consisting of more than 9000 touch-gestures from 147 participants. We experimented both *single-view* and *multi-view* learning techniques to find the best combination of touch-gestures able of distinguishing between adults and underages. Results show that the multi-view learning combining scrolls, swipes, and pinch-to-zoom gestures, achieves the best ROC AUC (0.92) and accuracy (88%) scores.

**Keywords** Online child protection · Touch-based gesture analysis · Machine learning · Multi-view learning · Techno-regulation

---

✉ Rocco Zaccagnino  
rzaccagnino@unisa.it

Alfonso Guarino  
alguarino@unisa.it

Nicola Lettieri  
n.lettieri@inapp.org

Delfina Malandrino  
dmandrino@unisa.it

<sup>1</sup> Dipartimento di Informatica, Università degli Studi di Salerno, Via Giovanni Paolo II, I-84084, Fisciano, SA, Italy

<sup>2</sup> National Institute for Public Policy Analysis (INAPP), Corso d'Italia 33, 00198 Rome, Italy

## 1 Introduction

Over the last decades, scientific developments have led deep change at all levels of human societies. In the world of *planetary-scale computation* [6], where ICT shape economies and jurisdictions, new regulatory mechanisms are increasingly needed to provide innovative solutions fitting to a reality in which technologies and social activities, illegal and legal, melt in an inextricable whole.

As for the *privacy by design paradigm* [10, 26], legal safeguards need to be designed together with the technological ones. Beyond interesting issues on the legal level, the idea to adopt regulation strategies mediated by digital technologies puts challenges on the computer science perspective and on the computational legal studies. In particular, the design of legal safeguards for children interacting in digital environments gives rise to problems which regard the regulation of all the contexts in which humans act through ICT.<sup>1</sup>

*Techno-regulation*, i.e., “*the intentional influencing of individuals behavior by building norms into technological devices*” [50], is one of the issues with which computer scientists and legal experts will have to deal with in the future. An example of techno-regulation can be seen in Digital Right Management systems [43], which incorporate copyright law into technological safeguards by limiting the use of copyrighted artifacts. Over the years, more advanced ways of integrating regulatory processes and ICT have been conceived, shedding light on two interesting aspects. First, instead of relying on *ex post* regulations by legal bodies, the *techno-rules* can be applied *ex ante*, making their violation impossible or still very difficult. Second, in contrast to traditional legal rules, often open to interpretation, techno-rules leave little room for ambiguity, thus reducing the likelihood of misunderstanding. The techno-regulation has so led to the creation of two categories of tools somehow capable of pursuing the same goals normally pursued by traditional legal instruments:

- *Detection/classification*: tools aimed at “identifying” facts and individuals to which particular legal consequences must be applied; for example, identification of individuals responsible for illegal/prohibited conducts [8].
- *Enforcement*: tools aimed at “safeguarding” interests protected by norms and regulations, designed to “*determine what people can or cannot do in the first place*” [21], or that can be used to nudge individuals promoting awareness and the compliance with rules [22, 27, 28, 50].

Techno-regulatory tools have already reached a high degree of technical heterogeneity ranging from cloud architectures and platforms design to plug-ins and software agents. In this scenario, machine learning and other Artificial Intelligence (AI) techniques will be increasingly able to support the development of more “smart” techno-regulatory solutions. Among the possible scenarios, an example could be developing tools capable of an intelligent identification of threats, illegal behaviors, and people responsible for illegal activities.

According to a 2018 Pew Research study [41], smartphones and social media are now an almost universal feature of teenage life in the United States, with more than nine-in-ten U.S. teens ages 13 to 17 accessing them. Apart from advantages and opportunities, the Internet exposes numerous threats for children/teenagers: from access to inappropriate content to exposition to dangerous behaviors. The same study also states that parents use a wide array

---

<sup>1</sup><https://www.unicef.org/endviolence/endviolenceonline/>

of strategies to monitor their teens' technology use, including 52% of parents who install parental control applications on their teens' mobile devices to filter and block inappropriate online activities.

In this paper, we present a novel approach aiming to protect children when interacting with smartphones. Specifically, we face the problem of classifying mobile users into two groups: *underages* and *adults*. The *age-threshold* used to distinguish between the two types of subjects is 16, as stated by the art. 8 of the *EU's General Data Protection Regulation*.<sup>2</sup> *Touch gestures* such as swipes, taps, and keystrokes, are common modes of interaction with smart touchscreen-enabled devices [9]. Major platforms including Android OS and iOS provide a variety of APIs to help developers detect gestures aiming to enhance apps' quality of experience. Access to these APIs allows apps to collect raw gesture data from different sensors available on the smart device. The fine-grained nature of this data became appealing for research, indeed touch gestures have been used for person recognition [45], for user authentication when combined with sensor data [47], and for other applications such as the complex task of bio-cryptography [16] and to foster social communication in children with autism [4, 55]. Thus, by starting with the observation that underages and adults perform commonly used touch-gestures in a different way on mobile devices, we developed an Android app in order to collect, in an experimental study involving 147 participants, more than 9000 heterogeneous touch-based gestures. We carried out several experiments to find, by exploiting machine learning techniques like in [13, 14, 25, 38, 45], the best combination of touch-based gestures capable of distinguishing between adults and underages.

The main *contributions* of our work can be summarized as follows:

- Proposal of a novel techno-regulatory approach exploiting machine learning techniques to provide safeguards against threats online. We study a set of touch-based gestures, to determine whether it is possible to distinguish who is accessing a smartphone, i.e., underage or adult, to guarantee protection.
- Evaluation of the effectiveness of our approach, on a large dataset including more than 9000 touch-gestures from 147 participants. We experimented both *single-view* and *multi-view* learning techniques to find the best combination of touch-gestures capable of distinguishing between adults and underages. Results show that the multi-view learning combining just three touch gestures, that is, scrolls, swipes, and pinch-to-zoom gestures, achieves the best ROC AUC (0.92) and accuracy (88%) scores.
- Several improvements against related works available in the literature which in general: (i) rely on just three touch gestures with fewer features to compute (ten for each gesture), (ii) do not consider some relevant multi-touch gestures, such as the pinch-to-zoom, which as proven, carry loads of information, (iii) do not experiment their approaches on different types of smartphones.

The rest of the paper is organized as follows. In Section 2, we describe some relevant works in the field of child protection and user identification based on the analysis of touch-based gestures performed on mobile devices. In Section 3, we provide an overview of the motivations of our research. In Section 4, we describe our approach to distinguish between underages and adults when analyzing touch-based gestures. In Section 5, we discuss the results and, finally, in Section 6 we conclude with some future directions.

---

<sup>2</sup><http://www.privacy-regulation.eu/it/8.htm>

## 2 Related work

Despite of several advantages and opportunities discussed in the previous section, Internet exposes children/teenagers to numerous threats, ranging from access to inappropriate content to exposition to dangerous behaviors. Several child protection systems, categorized as parental control systems, provide parents with numerous instruments to protect children from such threats.<sup>3</sup>

In recent years, several works [31–33] proposed systems for continuous authentication primarily based on data streams coming from gyroscope and accelerometer sensors. They integrate different techniques of deep learning, such as Convolutional Neural Networks and features fusion, obtaining encouraging results in users recognition. Differently from these works, we used only the information extracted from touch gesture data.

Other works available in literature are based on the analysis of touch-based gestures performed on smartphones, arguing that user information could be extracted and used to control smartphones' interactions. This information could improperly be used to track users and distinguish between them to provide access and functionalities diversified on a per-user basis.

In [46], the authors describe a novel multi-touch gesture based authentication technique, defined on a set of five-finger touch gestures. The authors built a classifier to recognize unique biometric gesture characteristics of an individual, achieving an accuracy rate of 90% with single gestures. In [19], the authors analyzed a set of 30 behavioral touch features, that can be extracted from raw touchscreen logs, and demonstrate that different users populate distinct subspaces of this feature space. The authors collected touch data from users interacting with a smartphone by using *only* up-down and left-right scroll gestures. They proposed a classification framework that after learning the touch-based user behavior, can proceed by accepting or rejecting the current user. In [5], the authors present *SilentSense*, a framework to authenticate smartphone users. The main idea is to exploit biometrics information obtained from the touches and leverage on the sensors to capture the device's micro-movements caused by user's actions. Conversely to the described approaches whose main goal is to preserve security on smartphones by avoiding intruders' access, our main goal is to recognize a specific category of users and adapt behaviors accordingly.

In [35], authors argued that touch-based gestures on touchscreen devices constitute a privacy threat. They show how the combination of swipe, tap, and handwriting gestures reveals up to 98.5% of information about users. It is worth noting that as explained in [37, 53], handwriting is not one of the most common touch gestures, and in literature, several studies acknowledge handwriting as a biometric [3]. In addition, the experiment they performed involved 89 participants but only 30 of them used all the envisioned games and hence provided samples for all gestures. Participants were free to join or leave the experimental phase whenever they wanted; therefore, the number of performed touch gestures considerably varied from participant to participant; conversely, in our study, all participants interacted with all games and therefore provided samples for all gestures.

In [52], authors present a technique to classify the users' age group from touch gestures. In their work, a child is a person having 6 years at most, and the dataset collected from 119 participants (89 children ages 3 to 6) included 587 samples. Using a Bayes' rule classifier, their technique delivered 86.5% accuracy when classifying each touch event one at a time, and 99% accuracy with a window of 7 or more consecutive taps. Differently from our work,

---

<sup>3</sup>For the under 16s, see the *Family Link* <https://families.google.com/familylink/>

the authors analyzed a dataset composed of instances associated with actions performed by children who are actually very young (3 to 6 years old). With these data, therefore, is very easy to classify individuals correctly as children have a very different tactile behavior compared to that of adults. Furthermore, it is clear that children's input performance and touch accuracy improve with age [2]. In our study the collected dataset included 9983 touch-based gestures among which 2942 gestures were taps. Our evaluation phase involved 147 participants with a better age distribution, and more than 30% of the participants were in the range 7–16 years old while more than 25% in the range 17–21. Finally, in our work, we do not consider touch gesture windows but combinations of single gestures performed by the same participant.

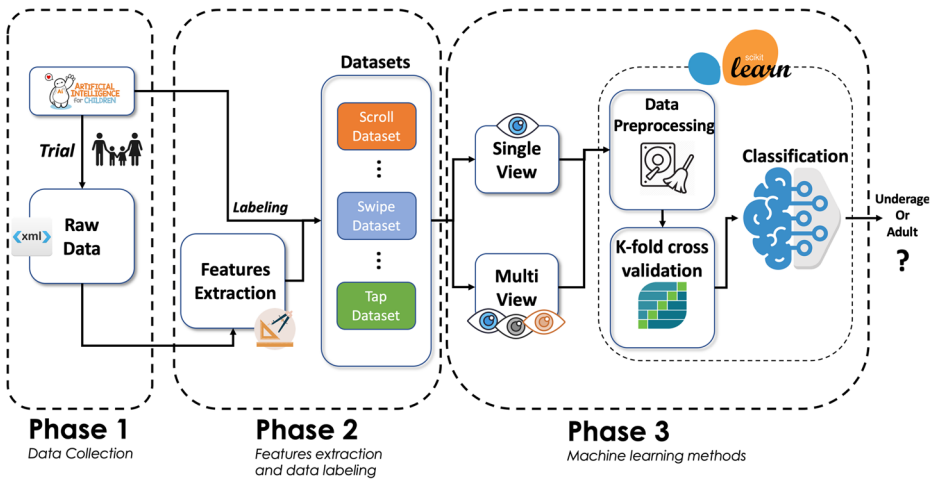
In another similar work [49], the authors present techniques to detect whether a child is on a mobile phone. They analyzed touch-based gestures as well as sensors features (and a combination of thereof). Fifty subjects (25 children and 25 adults) were recruited, with a clear gap between ages of underage (range 3–12) and adults (range 24–66). They evaluated the Random Forest classifier when using tap and stroke features and when bundling multiple gestures together. The results show good performance on the age group detection task with over 0.99 AUC for all the three approaches investigated. We differ from this work in several ways. First of all, our sample was larger, with 147 individuals and with a better age distribution. Second, we investigated a multi-view learning approach for the age classification problem, while authors in [49], clearly state that they had “*not examined other types of gestures like multi-finger gestures and the possibility of fusing different classifiers of different gestures for better and faster detection*”. Finally, they had not evaluated their models across different devices and different vendors.

### 3 Child protection: why?

Beyond motivational issues, one thing that clearly emerges by the related works is the need for cross-disciplinary experimental activities leading to more and more efficient techno-regulatory approaches binding regulatory priorities with opportunities provided by technological safeguards. From this perspective, experiments appear to be essential as there is still a lack of the expertise, the experiences, and the hybrid skills (computer science, law, interaction design) needed to achieve adequate regulatory results from all points of view (formal compliance with existing legal standards, effectiveness, scalability and technical feasibility). In this direction, we propose a techno regulated-based approach that exploits machine learning techniques to classify individuals, specifically underages, while they interact with technology, with the final goal of protecting them against specific online threats.

The explosion of information and communication technology, as emphasized in the 2015 edition of Guidelines for Industry on Child Online Protection released by International Telecommunication Union (ITU)<sup>4</sup> and by UNICEF, has created unprecedented opportunities for children and young people to communicate and access information but, at the same time, significant challenges to children's safety [12]. Online threats are numerous [34]: cyberbullying, grooming, hidden advertising, non-illicit contents that are still harmful to psychological well-being, and kiddie porn material.

<sup>4</sup><https://www.itu.int/en/cop/Pages/default.aspx>



**Fig. 1** A sketch of the phases envisioned in the proposed approach for the age detection. Phase 1: data collection, through the use of *AI4C* app. Phase 2: feature extraction, labeling an dataset building. Phase 3: application of machine learning methods (single and multi-view) and results

The latest Google Transparency Report<sup>5</sup> gives a rough idea of the scale of issues at stake. In the period from July to September 2018, about 1.7 million videos (more than the 22% of the whole number of the videos removed) have been removed just from YouTube because unsuitable for children (videos containing adult themes, nudity, violence). Despite that, there is still some danger because contents must be removed proactively or in real-time with an ad-hoc and customized solution: indeed, more than 25% of the content has been deleted after at least one visualization.

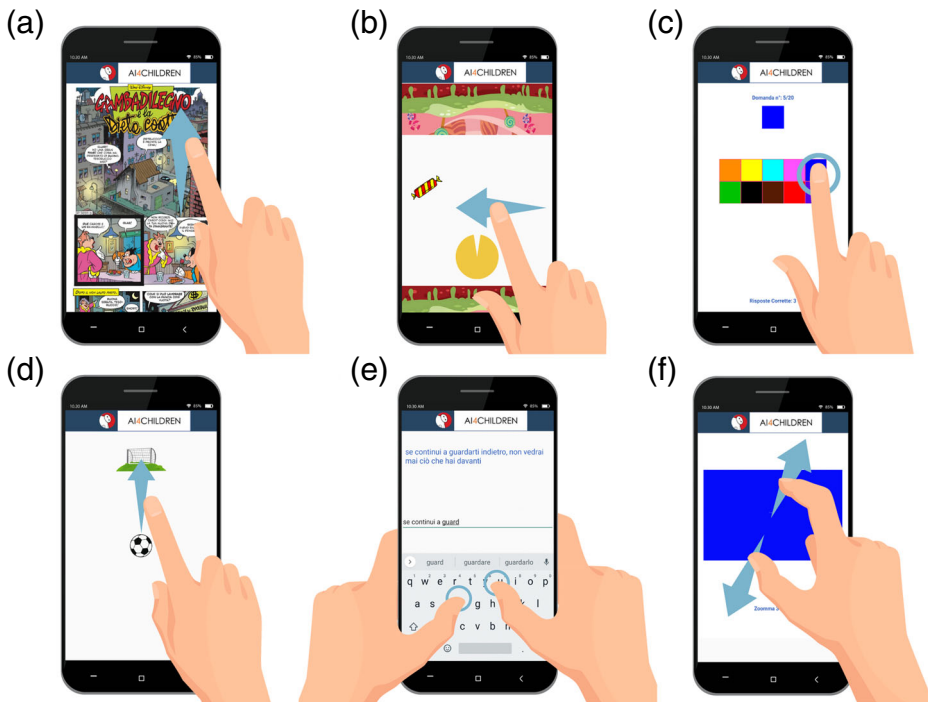
For children risks are even higher since they often circumvent or uninstall parental controls by lying about their age. At the same time, parents do not always understand the potential risks their children may encounter since they often underestimate teenagers' exposure to sexual content or overestimate it due to mass media messages [36, 42]. Against this background, it is easy figuring out how the issue led to national and international initiatives and regulatory actions, among which the most recent is the GDPR,<sup>6</sup> which limits the contents that can be shown to 13 to 15-year-old users. What emerges is that alongside traditional protections, it is necessary to develop other types of protection able to hinder use by protected parties.

## 4 Our approach for age detection

In this section we describe the approach we proposed for the classification of mobile users in *underages* or *adults*. The approach, as proposed in [13] and shown in Fig. 1, envisions different phases to (i) collect data, (ii) build the data sets through feature extraction and data labeling, (iii) apply machine learning methods to derive the best combination of gestures and the best machine learning technique for the age group classification.

<sup>5</sup><https://transparencyreport.google.com/netzdg/youtube>

<sup>6</sup><https://gdpr-info.eu>



**Fig. 2** The six micro-games of AI4C app: **a** Reading, **b** Candy Pacman, **c** Color Matching, **d** Score a goal, **e** Writing, **f** Calculation

#### 4.1 Phase 1: data collection

In order to collect data, we implemented an Android app that allows to capture and analyze user interactions with a mobile device. Such app, named *Artificial Intelligence for Children (AI4C app<sup>7</sup>)*, is essentially a simple game consisting of a series of tests, or *micro-games*. Each micro-game allows to capture a specific type of touch gesture. According to [37, 53], we consider the following touch-based gestures: *scroll*, *swipe*, *tap*, *drag & drop*, *pinch-to-zoom*.

AI4C app provides the following micro-games:

- *Reading (scroll gestures)*: it allows to read a Disney cartoon composed of a sequence of 6 pages, that have to be scrolled down (see Fig. 2a).
- *Candy Pacman (swipe gestures)*: it has been implemented to capture lateral swipes gestures performed to move Pacman and allow it to eat a candy (see Fig. 2b).
- *Color Matching (tap gestures)*: it shows a single color on the top of the window and a grid of colors on the bottom. When a color appears on the top, the user has to select the same color, by tapping it on the grid (see Fig. 2c).
- *Score a goal (drag & drop gestures)*: it allows the user to select the ballon, positioned in a random position on the screen, and drag it to score a goal (see Fig. 2d).

<sup>7</sup><https://bit.ly/2M4sE2G>



**Table 1** Description of the raw data captured through the *AI4C* app for each touch-based gesture and for each micro-game

Game	Gesture	Information
Reading	Scroll	<i>start time, start point, end point, duration, touch dimension touch pressure, velocity along x and y axis</i>
Pacman	Swipe	<i>start time, start point, end point, duration, touch dimension, touch pressure, velocity along axis</i>
Color Matching	Tap	<i>start time, start point, end point, touch dimension, touch pressure</i>
Score a goal	Drag & Drop	<i>start time, start point, end point, duration, touch dimension, touch pressure, velocity along x and y axis</i>
Writing	Keystroke	<i>time, number of characters, number of deletions</i>
Calculation	Pinch to zoom	<i>start time, finger1 start point, finger1 start dimension, finger1 start pressure, finger2 start point, finger2 start dimension, finger2 start pressure, length, fingers motion points, fingers motion velocity along x and y axis</i>

- *Writing (keystroke gestures)*: a short sentences from Disney cartoons is displayed on the top of the screen and the user has to write the same text at the bottom; it has been implemented to capture keystrokes events performed by the user while writing on the keyboard (see Fig. 2e).
- *Calculation (Pinch to zoom)*: it shows a blue rectangle with a small text inside. To read the text, the user has to pinch and zoom. The text is an arithmetic operation, for which the user has to write the right solution at the bottom of the screen (see Fig. 2f).

In Table 1 we provide details about the information we captured for each analyzed gesture when executing the 6 micro-games provided by our app.

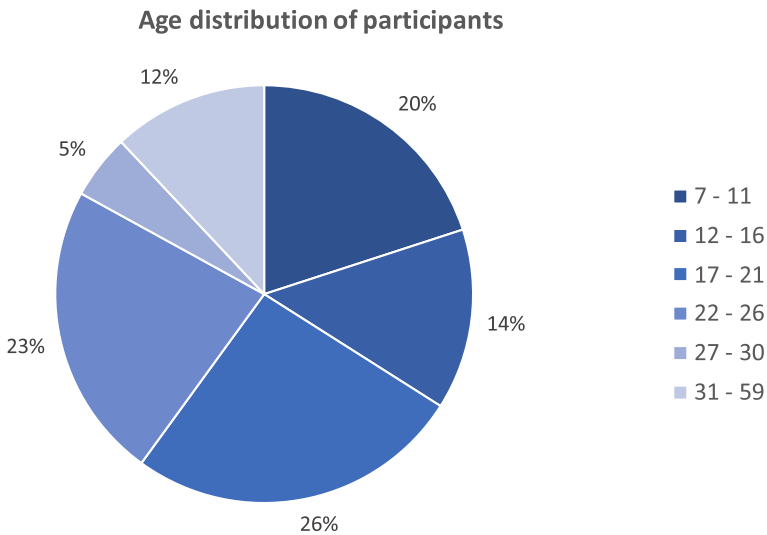
Before starting the collection phase we explained to subjects what they were expect to do in the study. For children, their parents completed written parental permission forms. All participants had to sign consent forms. We explained that we did not collected personal information during the game process except for username, device ID, and age. Raw data associated to gestures performed by users are tracked and saved for the subsequent analysis. We also explained that this data were kept confidential and used only for the period of the experimentation. Moreover, subjects used smartphones provided by us, in order to avoid to use personal devices and to be sure to use devices with the characteristics needed in our study.

Data collection, done through the *AI4C* app was mainly conducted at the “University of Salerno” and in collaboration with the “Gino Landolfi Primary School” in Agropoli (SA). The models of smartphones used for the experiments were an LG Nexus 5X, an ASUS ZenFone 2, and a HTC Desire 820. Data were collected from 147 participants, with the age distribution shown in Fig. 3. The age varies in the range from 7 to 59 years. The sample was composed of 46 underages and 101 adults. 80% of adults falls into the range 16–29 and 20% into the age range 30–59. Moreover, 61% of the participants were males and 39% females.

## 4.2 Phase 2: features extraction and data labeling

The aim of this phase is to identify significant features within the raw data, and therefore build the data sets that will be used by the machine learning algorithms tested during the





**Fig. 3** Age distribution of the participants at the study. 34% of participants was under the threshold of 16 years

classification phase (Section 4.3). As explained before, such data sets were labeled (0 for underages and 1 for adults) according to the *age-threshold* used to distinguish adults and underages by the GDPR and whose value is 16. Table 2 shows the data sets generated for each type of gesture, with the number of calculated features.

**Scroll down data set** As shown in Table 2, we collected 2594 scroll down gestures, where 31% were performed by underages and 69% by adults. To build this data set, for each scroll down gesture we calculated 50 features. As we can see in Table 3, there exist several types of features. Some of these features are returned “directly” by the Android Touch API; examples include: the number of fragments the scroll down gesture is composed of (*fragments number*), the duration of the scroll down gesture (*duration*), the coordinates of the initial point of the scroll down gesture ( $X_s, Y_s$ ), and so on. Other features regard the “geometric” properties of the scroll down gesture, and have to be specifically calculated.

**Table 2** Number of occurrences for each touch-based gesture with information about the category and the calculated features

Gesture/Dataset	Size	#Underages	#Adults	#Features
<i>Scroll down</i>	2594	807	1787	50
<i>Swipe right</i>	972	306	666	13
<i>Swipe left</i>	1005	315	690	13
<i>Tap</i>	2942	921	2021	29
<i>Drag &amp; Drop</i>	735	230	505	13
<i>Writing</i>	645	166	479	5
<i>Pinch-to-zoom</i>	1090	319	771	48

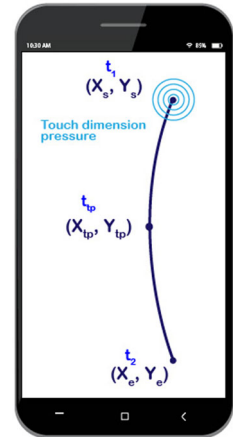
**Table 3** Details about the features calculated for each touch-based gesture

Gesture	Features
Scroll down	<i>fragments number, duration, start point <math>(X_s, Y_s)</math>, end point <math>(X_e, Y_e)</math>, length, total velocity, area, turning point <math>(X_{tp}, Y_{tp})</math>, turning point angle, acceleration<sub>s,tp</sub>, acceleration<sub>tp,e</sub>, start dimension, middle dimension, end dimension, min dimension, max dimension, mean dimension, 25% dimension, 50% dimension, 75% dimension, start pressure, middle pressure, end pressure, min pressure, max pressure, mean pressure, 25% pressure, 50% pressure, 75% pressure, start velocity X, middle velocity X, end velocity X, min velocity X, max velocity X, mean velocity X, 25% velocity X, 50% velocity X, 75% velocity X, start velocity Y, middle velocity Y, end velocity Y, min velocity Y, max velocity Y, mean velocity Y, 25% velocity Y, 50% velocity Y, 75% velocity Y</i>
Swipe right	<i>duration, start point <math>(X_s, Y_s)</math>, end point <math>(X_e, Y_e)</math>, dimension, pressure, length, velocity, acceleration, area, X velocity, Y velocity</i>
Swipe left	<i>duration, start point <math>(X_s, Y_s)</math>, end point <math>(X_e, Y_e)</math>, dimension, pressure, length, velocity, acceleration, area, X velocity, Y velocity</i>
Tap	<i>fragments number, X shift, Y shift, start dimension, middle dimension, end dimension, min dimension, max dimension, mean dimension, 25% dimension, 50% dimension, 75% dimension, start pressure, middle pressure, end pressure, min pressure, max pressure, mean pressure, 25% pressure, 50% pressure, 75% pressure</i>
Drag & Drop	<i>duration, start point <math>(X_s, Y_s)</math>, end point <math>(X_e, Y_e)</math>, dimension, pressure, length, velocity, acceleration, area, X velocity, Y velocity</i>
Writing	<i>time, number of characters, frequency, number of deletions, jaccard similarity</i>
Pinch-to-zoom	<i>finger1 start point <math>(X_1, Y_1)</math>, finger1 dimension, finger1 pression, finger1 start point <math>(X_2, Y_2)</math>, finger2 dimension, finger2 pression, fragments number, length, area, pinch grade, start dimension, middle dimension, end dimension, min dimension, max dimension, mean dimension, 25% dimension, 50% dimension, 75% dimension, start pressure, middle pressure, end pressure, min pressure, max pressure, mean pressure, 25% pressure, 50% pressure, 75% pressure, start velocity X, middle velocity X, end velocity X, min velocity X, max velocity X, mean velocity X, 25% velocity X, 50% velocity X, 75% velocity X, start velocity Y, middle velocity Y, end velocity Y, min velocity Y, max velocity Y, mean velocity Y, 25% velocity Y, 50% velocity Y, 75% velocity Y</i>

As an example, let  $(X_s, Y_s)$  and  $(X_e, Y_e)$  the coordinates of the start point and the end point of a scroll down gesture respectively, then the *length* of the scroll down gesture is defined as:

$$\text{length} = \sqrt{(X_e - X_s)^2 + (Y_e - Y_s)^2} \quad (1)$$

**Fig. 4** Scroll down gesture: turning point and touch information



Also, let  $X_{max}$  and  $X_{min}$  (resp.  $Y_{max}$  e  $Y_{min}$ ) be the maximum and the minimum coordinates on the  $x$  axis (resp. coordinates on the  $y$  axis) of the scroll down gesture respectively, then the covered *area* is defined as:

$$area = (X_{max} - X_{min}) * (Y_{max} - Y_{min}) \quad (2)$$

Other types of features include information about the velocity, touch dimension, and touch pressure. Specifically, for each of this information we considered: (i) the value with respect to the whole gesture running, (ii) the maximum, minimum, mean value, and (iii) the values in correspondence of the *quartiles* of the gesture (the value at start, 25%, 50%, 75% and at end of the gesture running).

Finally, the last features regard the *turning points*. Given a scroll down gesture, the turning point  $(X_{tp}, Y_{tp})$  is the point where the gesture changes direction respect the  $x$  axis. The information about the acceleration, is related to the turning point, and is captured by two features, i.e., the acceleration of the scroll down gesture from  $(X_s, Y_s)$  to  $(X_{tp}, Y_{tp})$  and the acceleration of the scroll down gesture from  $(X_{tp}, Y_{tp})$  to  $(X_e, Y_e)$ . Likewise, we included information about the touch dimension and the touch pressure (see Fig. 4) in correspondence of  $(X_{tp}, Y_{tp})$  (indicated as *middle dimension* and *middle pressure* in Table 3).

**Swipe data set** As shown in Table 2, we collected 972 swipe right gestures (resp. 1005 swipe left gestures), of which 31% belongs to underages and 69% to adults (resp. 315 belong to underages and 690 to adults). To build this data set (resp. *Swipe left data set*), for each swipe gesture we calculated 13 features, shown in Table 3. Specifically, we considered: duration, coordinates of the start point, coordinates of the end point, dimension, pressure, velocity along  $x$  axis, velocity along  $y$  axis, length, acceleration and, finally, the area.

**Tap data set** We collected 2942 tap gestures, of which 921 belongs to underages and 2021 to adults. To build this data set, for each tap gesture we have calculated 29 features (see Table 3). Specifically, we have considered the start point  $(X_s, Y_s)$  and the end point  $(X_e, Y_e)$ . As seen for the scroll down dataset, we have considered the number of fragments of the tap gesture, and both for velocity, dimension and pressure, we calculated maximum, minimum, mean values, values in correspondence of the quartiles, and maximum shift respect the  $x$  axis (resp.  $y$  axis).

**Drag & drop data set** We collected 735 drag & drop gestures, of which 230 belongs to underages and 505 to adults. To build this data set, for each gesture we calculated the same features calculated for the swipe gesture (see Table 3).

**Writing data set** We collected 645 writing gestures, where 166 belongs to underages and 479 to adults. To build this dataset, for each keystroke we calculated 5 features (see Table 3): time, number of characters, number of deletions, writing frequency of  $\bar{S}$  and the Jaccard similarity [24] between  $S$  and  $\bar{S}$ , where  $S$  is the sequence of “characters to write” (proposed by the micro-game), and  $\bar{S}$  is the sequence of “characters written” by the user performing the test.

**Pinch to zoom data set** We collected 1090 pinch-to-zoom gestures, of which 771 belongs to underages and 48 to adults. To build this data set, for each pinch-to-zoom gesture we calculated 48 features (see Table 3). Among these, the finger1 start point  $(X_1, Y_1)$ , the finger2 start point  $(X_2, Y_2)$ , the finger1 (resp. finger2) dimension, pression, length, covered area and pinch grade. As seen for the scroll down dataset, we have considered the number of fragments of the pinch-to-zoom gesture, and both for velocity, dimension and pressure, we calculated maximum, minimum, mean values and values in correspondence of the quartiles.

### 4.3 Phase 3: machine learning methods

In this section, first we provide some preliminary statistical information obtained by analyzing the data sets and the features calculated in the previous phase, and then we describe the machine learning-based methods we proposed.

#### 4.3.1 Some preliminary statistical observations

By analyzing features about pressure, dimension, duration and length, some differences between adults and underages have been found. As an example, to perform a scroll gesture, the underages required a 12% longer duration respect to the adults. Furthermore, the length of gestures performed by underages was, in pixel, 4% greater than that of the adults. The dimension of the adults’ touch dimension, instead, is 23% greater than that of the underages.

Overall, by observing the features: (i) for each type of gesture the dimension of the adults’ touch dimension is about 20% larger than that of the underages, (ii) the duration of swipe and drag & drop gestures of underages is greater than that of adults, (iii) the frequency of writing of the adults is about 116% higher than that of the underages, and (iv) the pressure of the touch is almost similar.

#### 4.3.2 Preprocessing and validation

The aim of this phase is to prepare the data set for the validation and testing phases. The classical approach in literature for problems of touch-based gesture classification is the *single-view learning*, in which each item of the data sets contains the features associated to one single gesture [5, 19, 52]. Other approaches aim to find the best *combination* of touch gestures which can be used to classify underages and adults. Such combinations are based on the *multi-view learning* techniques [29, 30, 39], in particular *early*, *intermediate* and *late integration*. The “early integration” consists in concatenating the features associated to different gestures (single-views) performed by the same individual; in this way each combination (concatenation of two or more single-view features-vector sample) represents one

sample in the data sets; this approach has the downside of considering large space features vectors. The “intermediate integration” consists in performing a features selection for each type of gesture [29, 30] (single-view), and then by combining the features selected; thus, for each individual, we combine (concatenate) such features in order to obtain the samples for the integrated datasets; the advantages of such a technique are: (a) the heterogeneous nature of the gestures’ features can be better used by separating the data, (b) the size of the output is reduced, and (c) the separate extraction of features for different type of gestures implements the *divide-et-impera* principle, reducing the complexity of the operations. With the “late integration” we train a classifier for each type of gesture (single-view) and then we use the outputs obtained by these models as input for a new model used for the final classification [20]. This method has the advantage to be easily implemented in parallel, because each model is fitted on a single view in an independent fashion but, as downside, it does not account interactions that could exist among single views.

In this phase we tested both the single-view learning techniques and the multi-view learning techniques for combinations of pairs, triples and quadruples of gestures (Section 4.3.3). For each of these models, the data set built was split into: (i) *the training set*, obtained by including the 80% of the elements (randomly chosen), and (ii) *the testing set*, obtained including the remaining 20% of the elements. Due to the different size of the adults and underages groups, we applied to the training set the SMOTE algorithm [11], applied in several studies [1, 18, 51], to do over-sampling on the data of the smaller group. In this way, for each learning technique (single-view and multi-view), we obtained balanced datasets.

In this work we used the most popular machine learning models available in literature and implemented by the *scikit-learn* Python library [40], that is, Random Forest (RF) [7], Support Vector Machines (SVM) [54], MultiLayer Perceptron (MLP) [44], and Logistic Regression (LR) [15]. Finally, in order to validate the machine learning models we perform a 10-fold cross-validation by using the *GridSearchCV* method, as proposed in [45, 49]. The performance of the classifiers have been evaluated with popular metrics: AUC—Area Under the Receiver Operating Characteristic (ROC) curve, and accuracy [17, 23, 48]. ROC curve, the most common way evaluate the performance of a binary classifier (also used in [49]), is created by plotting True Positive Rate against False Positive Rate. The ROC AUC value ranges from 0 to 1, where 1 is the perfect accuracy score.

For each model, we evaluated the following hyper-parameters (shown in *italic*):

- Random Forest (RF): its performance rely mainly on the *number of estimators*, therefore we tested from 20 to 200 estimators. Best results were found between 100 and 200 estimators.
- Support Vector Machines (SVM): it was tested on different *kernels* (polynomial, sigmoid, radial) and optimized with respect the *penalty parameter C* (from 0.1 to 100). Best results were found with radial kernel and C from 1 to 100.
- MultiLayer Perceptron (MLP): it mainly relies on *hidden layers size*. The number of hidden layers size was tested from 5 to the size of input layer (according to the touch gesture considered). Best results were found between  $\frac{1}{2}$  and  $\frac{4}{5}$  of input layer size. We also adopted the *lbfgs optimizer* in the family of quasi-Newton methods proved to converge faster and perform better on small datasets [56].
- Logistic Regression (LR): as *optimization algorithm* we used *liblinear* (good for small datasets), and *lbfgs*, and the *penalty norms l1* and *l2*. In addition, as for SVM, we tested *C* parameter from 1 to 100. Best results were found with *liblinear*, *l1* penatly norm, and *C* between 10 and 100.

**Table 4** Single-view: accuracy (*acc*) and ROC AUC (*auc*) values for each classifier and for each analyzed gesture (dataset)

	Gesture	RF		MLP		SVM		LR	
		acc.	auc	acc.	auc	acc.	auc	acc.	auc
	<i>scrollD</i>	<b>0.86</b>	<b>0.93</b>	0.81	0.87	0.85	n/a	0.74	0.73
	<i>swipeR</i>	0.72	0.79	0.73	0.79	0.71	n/a	0.65	0.71
	<i>swipeL</i>	0.71	0.79	0.68	0.71	0.71	n/a	0.66	0.69
	<i>tap</i>	0.66	0.66	0.56	0.65	0.63	n/a	0.59	0.61
SVM's auc is n/a because SVM is a non probabilistic classifier	<i>dad</i>	0.70	0.76	0.68	0.79	0.68	n/a	0.68	0.73
	<i>writing</i>	0.72	0.69	0.47	0.66	0.72	n/a	0.70	0.65
Best results are highlighted in bold	<i>pinch</i>	0.77	0.83	0.75	0.80	0.79	n/a	0.68	0.73

For more details on the hyper-parameters see Scikit-learn Library [40].

Results in validation show accuracy from 73% for tap to 92% for scroll down, and ROC AUC from 0.74 for writing to 0.98 for scroll down, when using the best performing classifier i.e., Random Forest.

### 4.3.3 Classification

Here we provide details about the results of the application of different machine learning methods, taking into account different classifiers and combining different types of gestures. We analyzed both single-view and multi-view learning techniques. We want to emphasize that in the following we will indicate with *scrollD*, *swipeL*, *swipeR*, *tap*, *dad*, *writing* and *pinch*, the Scroll down data set, the Swipe Left data set, the Swipe Right data set, the Tap data set, the Drag & Drop data set, the Writing data set and the Pinch to zoom data set, respectively.

**Single-view** In this model each sample in the data sets contains the features associated to a single touch gesture. Table 4 shows the ROC AUC and the accuracy score for each classifier on the test set. Best results have been obtained with the Random Forest classifier and when using the scroll down gesture, with ROC AUC of 0.93 and accuracy of 86%. For the other data sets ROC AUC values range from 0.76 to 0.83 (with Random Forest).

**Multi-view** Each multi-view learning technique (early, intermediate and late integration) has been tested on gestures pairs, triples, and quadruples, respectively. In the following we show the results obtained for each of these cases. We have to emphasize that we show the results only for gestures that exhibited the best results in the previous analysis (when applying the single-view learning approach). We also remark that for the *intermediate integration* we used a Random Forest classifier for the feature selection, and for the *late integration* method the strategy is the following: in all the experiments (pairs, triples and quadruples) we used the best single classifier for each single-view, and then the outputs were conveyed as input of a final Random Forest classifier.

- *Gestures pairs*. As shown in Table 5, for the *early integration*, the best result is achieved when combining the scroll down and the pinch-to-zoom data sets, with the best ROC AUC score of 0.91. This result is slightly worse with respect to the single-view learning when using only the scroll down data set.

**Table 5** Early, Intermediate, and late integration: accuracy (*Acc*) and ROC AUC (*auc*) values for the Random Forest classifier

Gesture pair	Early		Intermediate		Late	
	acc.	auc	acc.	auc	acc.	auc
<i>scrollD_swipeR</i>	0.78	0.85	0.79	0.87	0.81	0.82
<i>scrollD_swipeL</i>	0.79	0.86	0.76	0.86	0.79	0.81
<i>scrollD_pinch</i>	0.83	<b>0.91</b>	0.83	<b>0.91</b>	0.80	<b>0.86</b>
<i>swipeR_swipeL</i>	0.76	0.83	0.82	0.87	0.75	0.78
<i>swipeR_pinch</i>	0.86	0.89	0.80	0.87	0.80	0.81
<i>swipeL_pinch</i>	0.79	0.86	0.81	0.87	0.73	0.78

Best results are highlighted in bold

With the *intermediate integration*, for each dataset the Random Forest classifier selected the 10 most significant features. As an example, for the multi-view experiment *scrollD\_pinch* (combination of the scroll down and pinch-to-zoom datasets) we have shrunk down the features-vector space from a (50+48)-dimension space (of the early integration) to a (10+10)-dimension space. In Table 5 we can see that the best results have been obtained for this combination of gestures (*scrollD\_pinch*) with 0.91 and 83% for ROC and accuracy scores, respectively. As we can see in Table 5, the performance obtained with the late integration are lower than those obtained with the other two integration methods. This deterioration could be due to the nature of the integration technique, i.e., the error made on one of the two single-views spreads in the final classification. The deterioration has the same effect in all experiments, with the best result (ROC AUC score 0.86) obtained when combining scroll down and pinch-to-zoom, thus on the dataset *scrollD\_pinch*.

- *Gestures triples*. Similarly to the analysis about gesture pairs, for these experiments we report the best results only. The combinations that we do not show exhibited results lower than 0.85 for ROC AUC score and 83% for the accuracy score. As we can see in Table 6, by combining three type of gestures, the ROC AUC scores obtained with the *early integration* were always greater than 0.90, and the accuracy values are always greater than 85%. When applying the *intermediate integration*, for each dataset the Random Forest classifier selected the 10 most significant features. For instance, for the multi-view experiment *scrollD\_swipeR\_pinch* (combination of the scroll down, swipe right and pinch-to-zoom single-view datasets) the features-vector space has shrunk down from a (50+13+48)-dimension space (of the early integration) to a (10+10+10)-dimension space. Table 6 shows the results obtained with the intermediate integration, with the best ROC AUC score of 0.92 and accuracy of 88%, obtained when combining

**Table 6** Early, Intermediate, and Late integration: accuracy (*acc*) and roc auc (*auc*) values for RF classifier and for each triple of gesture datasets

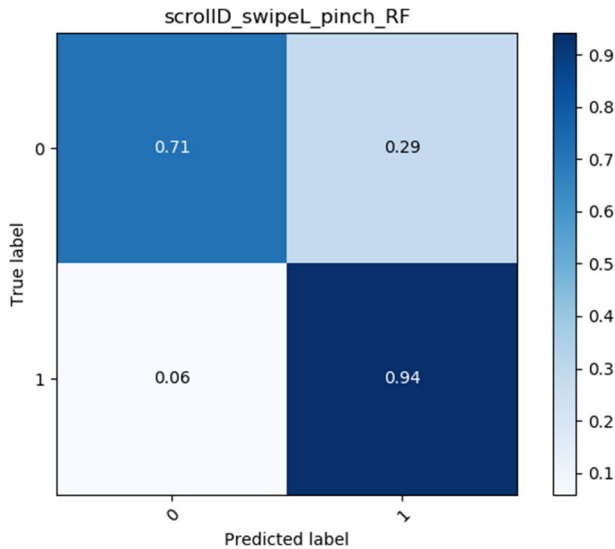
Gesture triple	Early		Intermediate		Late	
	acc.	auc	acc.	auc	acc.	auc
<i>scrollD_swipeR_pinch</i>	0.85	0.93	0.85	0.90	0.86	0.90
<i>scrollD_swipeL_pinch</i>	0.86	0.92	<b>0.88</b>	<b>0.92</b>	0.86	0.89

Best results are highlighted in bold





**Fig. 5** The 10 most relevant features for scroll down, swipe right and pinch-to-zoom datasets, selected by Random Forest classifier



**Fig. 6** Confusion matrix for gestures triples experiment, including scroll down, swipe left, and pinch-to-zoom single-views, when applying intermediate integration technique

the scroll down, the swipe left and the pinch-to-zoom datasets. Figure 5 shows the 10 most relevant features for scroll down, swipe right and pinch-to-zoom datasets. As we can see, in general the most important ones regard dimension, pressure and area of the touch gesture.

Figure 6 shows the normalized confusion matrix with true positive, false positive, true negative and false negative obtained results. In terms of ROC AUC scores (probabilistic measure) the classifier is very accurate (0.92), while in terms of accuracy (discrete measure) the classifier shows slightly lower performance when classifying underages (0.71). The problem is that errors (0.29) occur when classifying individuals with age near to the 16-years-old threshold. We also performed experiments in which samples did not included such individuals, obtaining results comparable to [35, 49]. This result suggest to further investigate the classification of individuals in the middle of puberty. Regarding the *late integration* (see in Table 6), the ROC AUC scores obtained are similar each other, in particular around 0.90. Compared to the results obtained in the experiments involving late integration and gestures pairs, by adding another single-view (another touch gesture) we obtain the same results. Whereas the late integration technique applied to gestures triples, compared with the other integration techniques, shows slightly lower performances. This indicate a correlation between single-views.

- **Gestures quadruples.** The last experiment considers a broader touch gestures combination. In our approach, the single-view used for the experiment has been chosen among the ones which showed the best results in the previous combinations (pairs and triples). Specifically we integrated scroll down, swipe right, swipe left, and pinch-to-zoom. As result of the with the *early integration*, by combining these four type of gestures, the best ROC AUC score is 0.90, while the best accuracy score is 0.84. As conclusion, this result does not improve the one obtained with gestures triples. When applying the *intermediate integration*, for each dataset the Random Forest classifier selected the 10 most significant features. Thus the features-vector space has shrunk down from

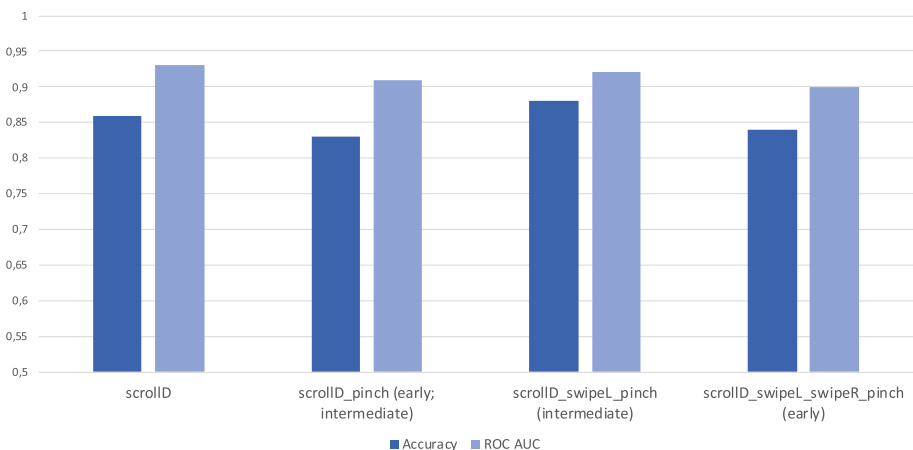
a  $(50+13+13+48)$ -dimension space (of the early integration) to a  $(10+10+10+10)$ -dimension space. By combining these four type of gestures, the ROC AUC score is 0.89 while the accuracy score is 83%. Such a result does not improve the one obtained when applying the early integration method. Finally, with *late integration* we obtained results comparable with the ones obtained with intermediate integration techniques applied to gestures triples, i.e., 0.89 as ROC AUC and 88% as accuracy score.

#### 4.4 Results

In this section we summarize the results obtained during our experiments. As we can notice in Fig. 7, in terms of ROC AUC score the single-view learning technique (*scrollID*) shows the best result (0.93). In the multi-view learning setting, when combining three gestures with the intermediate integration (*scrollID\_swipeL\_pinch*), the accuracy increases up to 88% (our best result) and the ROC AUC score reaches almost the same result (0.92) as in the single-view setting. We also observe that further increasing the number of gestures to consider in the multi-view learning technique does not improve the accuracy of the methods. In Section 5, we will discuss the results obtained and we will provide further explanations/intuitions.

### 5 Discussion

Looking at the results, we can conclude that scroll down is the type of touch gesture that best allows us to classify among underages and adults, with a ROC AUC score of 0.93 and an accuracy score of 86%, followed by pinch-to-zoom, swipe left, and swipe right. This is largely due to the rich set of information that can be derived as features from scrolls and pinch-to-zoom. In contrast, other touch gestures are simpler; thus, only a few characteristic features can be derived. Specifically, features based on the dimension, area, and pressure of the gesture are the most informative. This shows that there is a significant variation between underages and adults in the running of touch gestures like swipes and scrolls. We remark



**Fig. 7** Comparison between the best single-view and multi-view learning methods proposed

that our data collection procedure did not impose any condition on how users needed to interact with smartphones, such as sitting, standing, or walking.

Using strategies that combine different touch gestures allows us to improve the performance of the several machine learning classifiers considered. Indeed, for each multi-view learning (early, intermediate, late integration) based experiment, the classifiers' average performance is better than that reached during the single-view experiments. In particular, the touch gesture triple combining scroll down, pinch-to-zoom, and swipe left, with the intermediate integration technique, showed a ROC AUC score of 0.92 and an accuracy score of 88%. This indicates that it is preferable to adopt the intermediate integration strategy, with the advantage of using fewer features (30 compared to the 50 in single-view) without losing performances. Furthermore, this means that the way an underage/adult performs a scroll gesture is correlated with how he performs a swipe or a pinch-to-zoom gesture. As further proof of this correlation, the early and intermediate integration strategies always show better results than the late integration, which works with single-views separated and in parallel.

## 6 Conclusion

In this paper, we studied touch gestures to derive whether it is possible to distinguish who is accessing a mobile device (underages or adults) to provide safeguards against threats online. Existing protection solutions are not user-friendly. Additionally, it is challenging for parents to provide continuous monitoring of children using the smartphone without impacting on their privacy and autonomy.

Therefore, our result is a new regulatory approach that exploits machine learning techniques to provide automatic safeguards against threats online. The idea is to relieve parents from the challenging task of configuring protecting tools and constantly monitoring children and trusting an automatic mechanism that identifies the user accessing a smartphone and provides the right protection.

The experiments have shown positive results in terms of age group classification when analyzing touch gestures on different types of smartphones. The outcome is the applicability of machine learning to protect children through the use of automatic approaches. As best result, the intermediate integration technique in multi-view learning method with a combination of scroll down, pinch-to-zoom, and swipe left touch gestures and Random Forest, allowed us to reach an accuracy score of 88% and a ROC AUC score of 0.92.

Given these promising results, we are currently working along two different directions. Firstly, by performing experiments with more massive and more balanced datasets to improve results and by studying other types of gestures and sequences of the same touch gesture run by the same user sequentially, as proposed in [52]. Second, in terms of technology solutions, we aim at integrating our approach inside the *AI4C app*, with a parental control system designed to support in various ways the enforcement of the safeguards resulting from the normative framework in the field of online child protection. Specifically, we are planning to extend *AI4C app* functionalities by developing different safeguards: (i) an awareness-enhancing software allowing to display/send alerts in case of inappropriate behaviors; (ii) an intelligent browser capable of filtering harmful content, according to the user accessing to the mobile device. Finally, we will perform an evaluation study in order to assess both effectiveness and user (parents) satisfaction.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Agrawal A, Menzies T (2018) Is better data better than better data miners?: on the benefits of tuning smote for defect prediction. In: Proceedings of the 40th international conference on software engineering. ACM, pp 1050–1061
2. Anthony L, Brown Q, Tate B, Nias J, Brewer R, Irwin G (2014) Designing smarter touch-based interfaces for educational contexts. *Pers Ubiquitous Comput* 18(6):1471–1483
3. Ballard L, Lopresti D, Monrose F (2006) Evaluating the security of handwriting biometrics. In: Tenth international workshop on frontiers in handwriting recognition. Suvisoft
4. Bernardini S, Porayska-Pomsta K, Smith TJ (2014) Echoes: an intelligent serious game for fostering social communication in children with autism. *Inf Sci* 264:41–60
5. Bo C, Zhang L, Li XY, Huang Q, Wang Y (2013) Silentsense: silent user identification via touch and movement behavioral biometrics. In: Proceedings of the 19th annual international conference on Mobile computing & networking. ACM, pp 187–190
6. Bratton BH (2016) *The stack: on software and sovereignty*. MIT press
7. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
8. Brownsword R, Yeung K (2008) *Regulating technologies: legal futures, regulatory frames and technological fixes*. Bloomsbury Publishing
9. Buzzi MC, Buzzi M, Leporini B, Trujillo A (2017) Analyzing visually impaired people's touch gestures on smartphones. *Multimed Tools Appl* 76(4):5141–5169
10. Cavoukian A (2011) *Privacy by design in law, policy and practice*. A white paper for regulators decision-makers and policy-makers
11. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
12. Choo BJ, Joo HJ, Kim BS, Cha SH, Myung DH (2019) Using multimedia based big data for child abuse prevention system (based on the establishment of e-child welfare support system). *Multimed Tools Appl* 78(20):28805–28814
13. Cosimato A, De Prisco R, Guarino A, Lettieri N, Malandrino D, Sorrentino G, Zaccagnino R (2019) The conundrum of success in music: playing it or talking about it? *IEEE Access* 7:123289–123298. <https://doi.org/10.1109/ACCESS.2019.2937743>
14. Cozza F, Guarino A, Isernia F, Malandrino D, Rapuano A, Schiavone R, Zaccagnino R (2020) Hybrid and lightweight detection of third party tracking: design, implementation, and evaluation. *Comput Netw* 167(106):993
15. Dreiseitl S, Ohno-Machado L (2002) Logistic regression and artificial neural network classification models: a methodology review. *J Biomed Inform* 35(5–6):352–359
16. Eskander GS, Sabourin R, Granger E (2014) A bio-cryptographic system based on offline signature images. *Inf Sci* 259:170–191
17. Fawcett T (2006) An introduction to roc analysis. *Pattern Recognit Lett* 27(8):861–874
18. Fernández A, García S, Herrera F, Chawla NV (2018) Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *J Artif Intell Res* 61:863–905
19. Frank M, Biedert R, Ma E, Martinovic I, Song D (2013) Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Trans Inf Forensics Secur* 8(1):136–148
20. Fratello M, Caiazza G, Trojsi F, Russo A, Tedeschi G, Tagliaferri R, Esposito F (2017) Multi-view ensemble classification of brain connectivity images for neurodegeneration type discrimination. *Neuroinformatics* 15(2):199–213
21. Hassan S, De Filippi P (2017) The expansion of algorithmic governance: from code is law to law is code. *Field actions science reports*. *J Field Actions (Special Issue 17)*:88–90

22. Hildebrandt M (2011) Legal protection by design: objections and refutations. *Legisprudence* 5(2):223–248
23. Hossin M, Sulaiman M (2015) A review on evaluation metrics for data classification evaluations. *Int J Data Min Knowl Manag Process* 5(2):1
24. Jaccard P (1901) Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat* 37:547–579
25. Klempous R, Nikodem J, Jacak W, Chaczko Z (2013) Advanced methods and applications in computational intelligence, vol 6. Springer Science & Business Media
26. Langheinrich M (2001) Privacy by design—principles of privacy-aware ubiquitous systems. In: International conference on ubiquitous computing. Springer, pp 273–291
27. Leenes R (2011) Framing techno-regulation: an exploration of state and non-state regulation by technology. *Legisprudence* 5(2):143–169
28. Lettieri N, Guarino A, Malandrino D, Zaccagnino R (2019) Platform economy and techno-regulation—experimenting with reputation and nudge. *Future Internet* 11(7):163
29. Li Y, Ngom A (2015) Data integration in machine learning. In: 2015 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, pp 1665–1671
30. Li Y, Wu FX, Ngom A (2016) A review on machine learning principles for multi-view biological data integration. *Brief Bioinform* 19(2):325–340
31. Li Y, Hu H, Zhou G (2018) Using data augmentation in continuous authentication on smartphones. *IEEE Internet Things J* 6(1):628–640
32. Li Y, Hu H, Zhu Z, Zhou G (2020) Scanet: sensor-based continuous authentication with two-stream convolutional neural networks. *ACM Trans Sens Netw (TOSN)* 16(3):1–27
33. Li Y, Zou B, Deng S, Zhou G (2020) Using feature fusion strategies in continuous authentication on smartphones. *IEEE Internet Comput* 24(2):49–56
34. Livingstone S, Haddon L (2009) Eu kids online. *Z Psychol/J Psychol* 217(4):236
35. Masood R, Zhao B, ZH, Asghar HJ, Kaafar MA (2018) Touch and you're trapp (ck) ed: quantifying the uniqueness of touch gestures for tracking. *Proc Privacy Enhanc Technol* 2018(2):122–142
36. Mitchell KJ, Finkelhor D, Wolak J (2005) Protecting youth online: family use of filtering and blocking software. *Child Abuse Neglect* 29(7):753–765
37. Nacher V, Jaen J, Navarro E, Catala A, González P (2015) Multi-touch gestures for pre-kindergarten children. *Int J Hum-Comput Stud* 73:37–51
38. Narudin FA, Feizollah A, Anuar NB, Gani A (2016) Evaluation of machine learning classifiers for mobile malware detection. *Soft Comput* 20(1):343–357
39. Pavlidis P, Weston J, Cai J, Noble WS (2002) Learning gene functional classifications from multiple data types. *J Comput Biol* 9(2):401–411
40. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
41. PewResearchCenter: Teens socialmedia&technology2018 (2018) <https://www.pewinternet.org/2018/05/31/teens-social-media-technology-2018/>
42. Richardson CR, Resnick PJ, Hansen DL, Derry HA, Rideout VJ (2002) Does pornography-blocking software block access to health information on the internet? *JAMA* 288(22):2887–2894
43. Rosenblatt W, Mooney S, Trippe W (2001) Digital rights management: business and technology. Wiley, New York
44. Rumelhart DE, Hinton GE, Williams RJ et al (1988) Learning representations by back-propagating errors. *Cognit Model* 5(3):1
45. Rzecki K, Pławiak P, Niedźwiecki M, Sośnicki T, Leśkow J, Ciesielski M (2017) Person recognition based on touch screen gestures using computational intelligence methods. *Inf Sci* 415:70–84
46. Sae-Bae N, Ahmed K, Isbister K, Memon N (2012) Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 977–986
47. Shen C, Chen Y, Guan X (2018) Performance evaluation of implicit smartphones authentication via sensor-behavior analysis. *Inf Sci* 430:538–553
48. Swets JA (1988) Measuring the accuracy of diagnostic systems. *Science* 240(4857):1285–1293
49. Toan N, Aditi R NasirM (2019) Kid on the phone! Toward automatic detection of children on mobile devices. *Comput. Secur.* 84: 334–348
50. Van den Berg B, Leenes RE (2013) Abort, retry, fail: scoping techno-regulation and other techno-effects. In: Human law and computer law: comparative perspectives. Springer, pp 67–87
51. Van Hulse J, Khoshgoftaar TM, Napolitano A (2007) Experimental perspectives on learning from imbalanced data. In: 24th International conference on machine learning, pp 935–942

52. Vatavu RD, Anthony L, Brown Q (2015) Child or adult? Inferring Smartphone users' age group from touch measurements alone. In: IFIP conference on human-computer interaction. Springer, pp 1–9
53. Villamor C, Willis D, Wroblewski L (2010) Touch gesture reference guide. Touch Gesture Reference Guide
54. Wang L (2005) Support vector machines: theory and applications, vol 177. Springer Science & Business Media
55. Wojciechowski A, Al-Musawi R (2017) Assisstive technology application for enhancing social and language skills of young children with autism. *Multimed Tools Appl* 76(4):5419–5439
56. Zhao R, Haskell WB, Tan VY (2018) Stochastic l-bfgs: improved convergence rates and practical acceleration strategies. *IEEE Trans Signal Process* 66(5):1155–1169

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.