IAC: On the Feasibility of Utilizing Neural Signals for Access Control

Md Lutfor Rahman
Computer Science and Engineering
University of California Riverside
mrahm011@ucr.edu

Ajaya Neupane
Computer Science and Engineering
University of California Riverside
ajaya@ucr.edu

Chengyu Song
Computer Science and Engineering
University of California Riverside
csong@cs.ucr.edu

ABSTRACT

Access control is the core security mechanism of an operating system (OS). Ideally, the access control system should enforce context integrity, *i.e.*, an application can only access security and privacy sensitive resources expected by users. Unfortunately, existing access control systems, including the permission systems in modern OS like iOS and Android, all fail to enforce context integrity thus allow apps to abuse their permissions. A naive approach to enforce context integrity is to prompt users every time a sensitive resource is accessed, but this will quickly lead to habituation. The state-of-art solutions include (1) user-driven access control, which binds a predefined context to protected GUI elements and (2) predicting users' authorization decision based on their previous behaviors and privacy preferences. However, previous studies have shown that the first approach is vulnerable to attacks (*e.g.*, clickjacking) and the second approach is challenging to implement as it is difficult to infer the context. In this work, we explore the feasibility of a novel approach to enforce the context integrity—by inferring what task users want to do under the given context from their neural signals; then automatically authorizes access to a predefined set of sensitive resources that are necessary for that task. We conducted a comprehensive user study including 41 participants where we collected their neural signals when they were performing tasks that required access to sensitive resources. After preprocessing and features extraction, we trained machine learning classifier to infer what kind of tasks a user wants to perform. The experiment results show that the classifier was able to infer the high-level intents like take a photo with a weighted average precision of 88%.

CCS CONCEPTS

• **Security and privacy** → **Systems security**; *Usability in security and privacy*; • **Human-centered computing** → Empirical studies in ubiquitous and mobile computing;

KEYWORDS

brain-computer interface, intent-driven access control, machine learning

ACSAC'18, December 3–7,2018, San Juan, PR, USA

ACM ISBN 978-1-4503-6569-7/18/12…$15.00
https://doi.org/10.1145/3274694.3274713

ACM Reference Format:
Md Lutfor Rahman, Ajaya Neupane, and Chengyu Song. 2018. IAC: On the Feasibility of Utilizing Neural Signals for Access Control. In *Proceedings of 2018 Annual Computer Security Applications Conference (ACSAC'18).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3274694.3274713

1 INTRODUCTION

Access control is the core security mechanism of an operating system (OS). It decides what resources a subject can access and in what way the access can be performed (*e.g.*, read, write, execute). Classic access control models include Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-based Access Control, Attribute-based Access Control, etc. An important property of all these models is that a subject is not the human user, but a process/thread that operates on behalf of the human user (*i.e.*, a proxy). Therefore, the effectiveness of these models heavily relies on the assumption that *the software truly operates as the user intended.* This assumption generally held in the early era of computing history when software was either written by users themselves or by a trusted authority (*e.g.*, an administrator). However, with the boom of the software industry, this assumption no longer holds—as users, we usually do not fully understand what a piece of software truly does. Consequently, numerous security and privacy issues arise. For example, ransomware can abuse our credentials to encrypt our files and spyware can easily steal our private information.

Modern operating systems like iOS and Android use sandbox and permission system to mitigate this threat. In these systems, apps are no longer trusted—by default, they can only access to their own files and limited system resources. Accesses to user-owned data and privacy sensitive sensors are mediated by the permission system through which user can decide either to allow the accesses or deny them. While this is a step forward, the problem of these systems (iOS and Android M+) is that they only ask users to authorize the first access to the protected resources, *a.k.a.*, *ask-on-first-use* (AOFU). Any subsequent access to the same resource will be automatically allowed unless users manually revoke the permission. However, since an app can have different functionalities and the resources may be used under quite different *context*, recent research results have shown that AOFU failed to protect users' privacy over half of the time [65].

A straightforward idea to solve this problem is to prompt user *every* time a protected resource is about to be accessed. However, as the number of accessing requests can be huge (*e.g.*, Wijesekera *et al.* found that a single app can make tens of hundreds of requests per day [65]), this approach can easily cause habituation and loose its effectiveness. So, the real challenge is *how to reduce the number of prompts without sacrificing users' privacy.*

A general idea to solve this challenge is to *infer* what decision a user is likely to make thus avoiding redundant prompts [33, 38, 41, 47, 55, 57, 66]. Existing solutions can be divided into two directions. Solutions in the first direction associate GUI gadget with predefined context, then extract user's authorization from their interactions with the gadget, *a.k.a. user-driven access control* [33, 38, 41, 51, 55–57, 59, 67]. For example, a downloaded file is allowed to be executed only if the user has clicked the "Save" button to save it [38]; an email is allowed to be sent only if the user has clicked the "Send" button and its content must match what is displayed on screen [33]; and only when the user clicks the "Camera" button can an app access the camera device [41, 55, 57]. However, this associating user's authorization to GUI gadgets has two major drawbacks. First, there are many GUI attacks that can mislead the user, such as clickjacking attacks [30]. For this reason, existing user-driven access control models have to employ additional steps to prevent such attacks, *e.g.*, by isolating the gadgets from the application and letting the OS to render [57]. Secondly and more importantly, not all legitimate resources accesses are initiated from user interaction [23].

The second direction is to *predict* users' authorization decision based on their privacy preference [37], privacy profile [36], or previous authorization decisions and other behaviors [47, 66]. Because the decisions are usually context-sensitive, the biggest challenge for this direction is how to infer the *context*. Olejnik *et al.* used 32 raw features to define a unique context but admitted that they are not exhaustive [47]. Wijesekera *et al.* believed that the problem of inferring the exact context in an automated way is likely to be intractable thus focused on inferring when context has *changed* [66].

In this study, we explore the feasibility of a new way to infer users' authorization decisions—*by directly inferring their intent through the brain-computer interface*. Our observation is that the notion of contextual integrity [46] suggests that each unique context will setup a set of corresponding *social norms* on how users would expect their privacy information to be used. Whenever the information is used in ways that defy the users' expectations, a privacy violation occurs. Applying this notion to the access control systems (permission models) implies that we can automate the authorization process by (1) associating each context of an app with a functionality it appears to perform; (2) associating each functionality with a set of *expected* sensitive resources that are necessary (*i.e.*norms); and (3) limiting the requested resources to the expected set. However, as discussed earlier, the first step—inferring functionality from a context is very difficult. The key idea behind our approach is that we can actually avoid solving this challenging problem by utilizing our brain as a "magic" inference engine to directly output the result: *what is the intended functionality the user wants to perform under the given context*. Once we can infer intents from the user's brain signals, we can easily follow step (2) and (3) to make authorization decisions.

As the first footprint towards this direction, this work studies the feasibility of constructing such a decision-making system based on non-invasive electroencephalography (EEG) headset. Recent advances of the EEG sensor technology have enabled us to use consumer-grade headset to capture brain signals that used to be only available to clinical settings with invasive probes. Utilizing these EEG sensors, researchers have shown it is possible to recognize simple mental tasks as well as playing games. In this study, we aim to explore the feasibility of utilizing these sensors to infer user's intent through answering the following research questions:

- **Q1**: Is it possible to extract high-level intents (*e.g.*, taking a photo) from the neural signals with a machine learning classifier?
- **Q2**: Is the accuracy of the classifier high enough to support automated authorization?

To answer these questions, we designed and conducted a user study with 41 participants. Experiment over the collected data indicates that the answers to the above research questions are mostly positive. Specifically, our classifier is able to distinguish four different high-level intents (taking a photo, taking a video, choosing a photo from the gallery, and cancel) with a weighted average *Precision* of 88.34%, while the weighted average *Recall* is 86.52%, and the weighted average $F-measure$ is 86.92%.

Contributions. In brief, our contributions in this paper are:

- We designed a new intent-driven access control model that relies on inferring of user's high-level intents through the brain-computer interface (BCI).
- We experimentally validated the feasibility of constructing such a system with consumer-grade EEG headset via a user study of 41 participants. Our experimental results show the feasibility of intent-driven access control. To our best knowledge, this is the first study of utilizing brain signals to protect users' privacy.

The rest of the paper is organized as follows: §2 provides the background on Electroencephalography (EEG), Event-related potential (ERP), Emotiv Epoc + headset and Brain Computer Interface (BCI) which are required to understand our study, §3 introduces the threat model of our new access control design and how it works, §4 presents the experiments design and experimental procedures, §5 provides the details of how raw EEG data is processed before feeding into a machine learning algorithm, §6 empirically answers the two research questions, §7 discusses the limitations of our existing design and possible future work, §8 compares our work with related research, and §9 concludes the paper.

2 BACKGROUND

In this section, we give the background of Electroencephalography (EEG), event-related potential (ERP), Emotiv Epoc + headset and Brain Computer Interface (BCI).

EEG. Electroencephalography (EEG) is a monitoring technique that records the brain's electrical activities. The recorded EEG data is a time series data. Voltage fluctuations generated from neurons inside the brain are captured by electrodes and amplified. The electrodes are usually placed in a non-invasive way (*i.e.*, attached to the skin of the head scalp), but they can also be used invasively. For this study, we used non-invasive EEG sensors.

Event-Related Potentials. Event-related potentials (ERPs) are small but measurable (with an EEG sensor) voltages changes generated by the brain in response to a stimulus event. The stimulus events include a wide range of cognitive, sensory, or motor activities, such as showing different letters to the participants, or in our experiments, performing a given task with mobile apps. ERPs

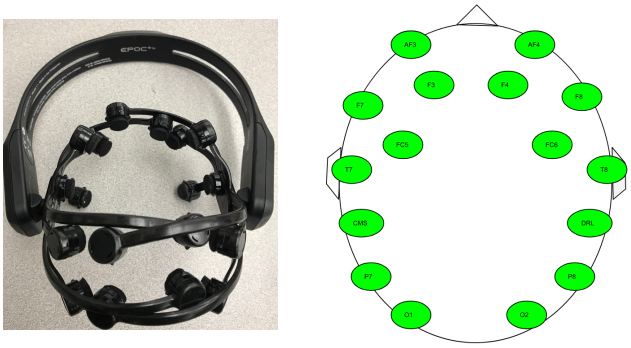


Figure 1: A 14-channel Emotiv Epoc+ headset used to collect data in this study (left) and electrode positions on the headset (right).

are time-locked to the stimulus, *i.e.*, given stimuli, an EEG voltage change is always expected in a known time frame. Because the voltage changes are small, ERPs are calculated by averaging multiple trials of the time-locked EEG samples. This procedure will filter out the background EEG noise and extract ERPs. The resulting ERP waveforms consist of a sequence of positive and negative voltage deflections, which are called *components*. So far, researchers have discovered more than a dozen ERP components [61]. Among them, the most well-studied ERP component is P300 or P3 wave.

Emotiv EPOC+ Headset. The Emotiv EPOC+ wireless headset (Figure 1) [20] is a consumer-grade EEG device that is widely used in gaming and entertainment industry. It allows gamers to control computer game based on their thoughts or facial expression [54]. We used this device in our study because it is significantly less expensive than other clinical-grade EEG devices and is more portable. For this reason, it is also widely used in research projects [40, 44]. The headset consists 14 data collecting electrodes (AF3, AF42, F3, F4, F7, F8, FC5, FC6, O1, O2, P7, P8, and T7, T8) and two reference electrodes (CMS/DRL). The electrodes are placed according to the International 10-20 system (Figure 1).

Getting good quality signal from the Emotiv headset requires pressing the two references for 5s or more before data collection. The Emotiv headset collects EEG data at 128 sample per second [1]. The captured EEG signals are then converted to digital form. The digital data are then processed and transmitted as encrypted data to the stimuli computer via USB dongle receiver. This proprietary USB dongle communicates with Emotiv headset in 2.4 GHz frequency. Emotiv also provides companion software for its device. EmoEngine is a software component for post-processing data. This software exposes data to BCI applications via the Emotiv Application Programming Interface (Emotiv API). Pure.EEG is a software component for data collection, which is used in this study. Pure.EEG collects data from the Emotiv device independently via the USB dongle and can upload data to the cloud and download from the cloud recorded sessions.

BCI. Brain-Computer Interface (BCI) is a new type of user interface where our neural signals are interpreted into machine understandable commands. Here, it converts brainwaves into digital commands which instruct machine to conduct various tasks. For example, researchers have shown it is possible to use BCI to allow patients who suffer from neurological diseases like locked-in syndrome to spell words and move computer cursors [9, 62] or allow patients to move a prosthesis [60]. With BCI, instead of using physical interactions human can use mind interaction. In our study, we choose this interface as it can directly reveal the user's intent thus is resistant to some perception manipulation attacks (*e.g.*, clickjacking [30]).

[1] The device internally collected data at a frequency of 2048 Hz, then down-sampled to 128 Hz before sending it to the computer.

3 INTENT-DRIVEN ACCESS CONTROL

In this section, we introduce how our new access model would work. We start with the threat model and assumptions. Then we show how to realize the model with BCI.

3.1 Threat Model and Assumptions

We make following assumptions for constructing a BCI-based intent inference engine and use it to authorize access to user-owned sensitive resources and sensors. We assume the OS is trusted. Attacks that exploit OS vulnerabilities to gain illegal access to the protected resources and sensors are out-of-scope. We also assume the OS already employs a permission model that considers context integrity (*e.g.*, an ask-on-every-use model). Our goal is not to replace the existing access control system, but to make it more user-friendly.

We assume our adversary is skilled application developer aiming to gain access to the user-owned resources/sensors without user's consent and abuse such access. Attackers are allowed to launch UI attacks (*e.g.*, clickjacking) to mislead users. With one exception, to correctly identify which app the user is interacting with, the OS should not allow transparent overlay [26]. We consider phishing-style attacks (*e.g.*, explicitly instructing users to perform sensitive operations) and side-channel attacks (that leak protected information) out-of-scope.

Regarding access to the raw EEG data, we envision a restricted programming model. Specifically, existing platforms like Emotiv expose raw EEG data to any applications build against their APIs. This programming model has been proven to be vulnerable to side-channel attacks that can infer user's sensitive and private information [10, 25, 40, 44]. To prevent such attacks, we assume a programming model that is similar to the voice assistants [4, 42]. That is, the raw EEG data is exclusively accessed by a trusted module, which will interpret the data and translate into app understandable events. We assume our inference engine to be part of this module and is implemented correctly. We also do not consider physical attacks against the EEG sensors.

3.2 IAC via BCI

Our BCI-based intent-driven access control system works similarly to the systems proposed in [47, 66]. In particular, the baseline access control system will prompt the user to authorize every access to protected resources. The goal of IAC is to minimize the number of the prompts by checking whether the access is intended by the user. Specifically, a legitimate access to protected resource should be (1) initiated by user's intent to perform a certain task under the

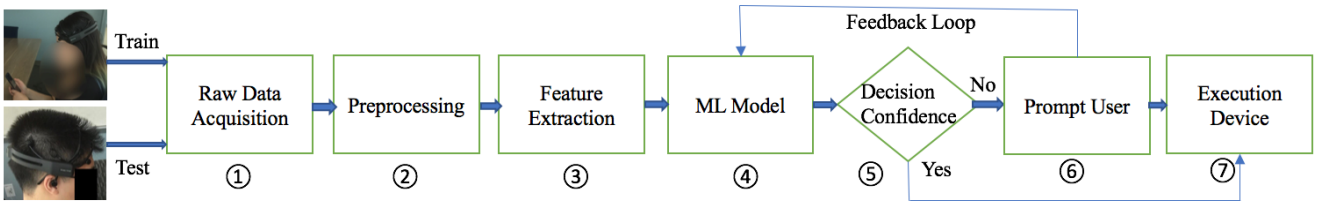


Figure 2: Overview of IAC's Architecture. IAC will ① continuously monitor the brain signals using the EEG sensor and user interaction with the system. Upon an input event, IAC will create an ERP, ② preprocess the raw EEG data to get purer signals, ③ extract feature vector from the purified signals, ④ feed the extracted features to a ML model to infer the user's intent. In step ⑤, if the ML gives enough decision confidence, IAC will directly ⑦ authorize access to protected resources. Otherwise, it will ⑥ prompt users to authorize the access and improve the ML model with the feedback loop.

presented app context[2] and (2) within the expected set of necessary resources for that task. Therefore we can create intent-based access control mechanism based on ERPs and use them as the inputs to a machine learning classifier. The data flow diagram for IAC is given in Figure 2.

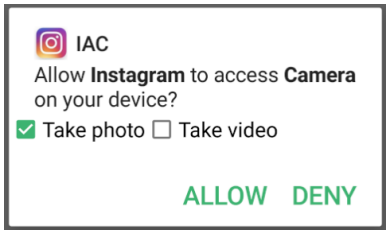


Figure 3: Example permission request. Compare to existing permission request, the biggest difference is IAC also asks for intended task (*e.g.*, taking a photo).

To train the classifier, we use user's explicit answers to the ask-on-every-use prompt as the ground truth. However, instead of just asking the user to authorize the access, IAC will also list a set of tasks that rely the requested resource for user to choose (e.g., Figure 3). If the access is authorized, we label the ERP with the task user has chosen; otherwise the event is discarded.

During the normal operations, the OS will continuously monitor neural signals through the BCI device as well as user's interaction with the system to create and cache most recent ERPs. ERPs are bound to the app to which the input event is delivered (*e.g.*, the most foreground app at that moment) and will expire after a context switch. This prevents one app from "stealing" another app's ERP. Upon an application's request to access a protected resource (*e.g.*, camera), the access control system will retrieve the most recent ERP. The ERP will then be fed into the trained classifier to infer whether the user intended to *perform a task that requires access to that resource*. If so, permission is automatically granted to that request; if the intended task does not require the permission or the confidence of the classification result is not high enough, IAC will fall back to prompt the user to make the decision. The ground truth collected from the prompt window is then to update the ML model. As demonstrated in previous works [47, 66] and our own experiment, this feedback is important for fine tuning the ML model to improve the precision of the prediction.

[2] Note that unlike access control gadget, we do not require the intent to be expressed through certain interactions with the app's GUI.

Applicable Scenarios. Apparently, using BCI-based access control for existing systems like desktop and mobile devices is impractical; users need to wear the device all the time. However, this field is advancing fast and companies like Facebook and Neuralink are laying out projects to decode users' intents into machine readable commands to scroll menus, select items, launch applications, and manipulate objects [13]. BCI has also been used in manufacturing to control machines [1, 58] or to monitor workers' mental status in order to avoid over-stressing [16]. With the rapid progress in neural imaging and signal processing, in not so distant future, BCI-based applications can be far beyond gaming and entertainment. Hence, we believe BCI could become ubiquitous and a practical way to interact with digital systems and our IAC be easily integrated into such systems to protect users' privacy.

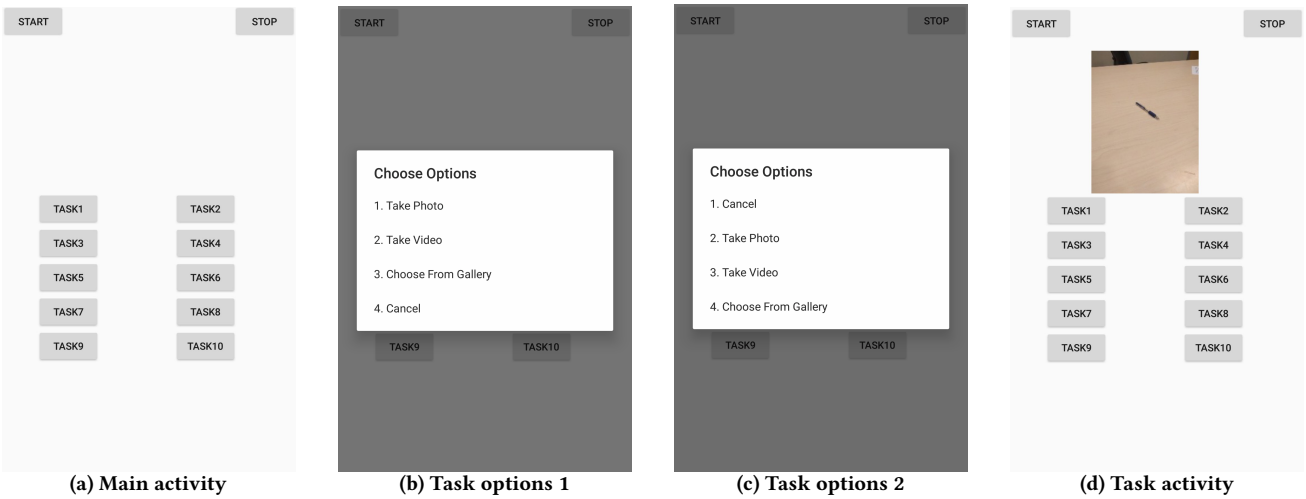
4 EXPERIMENT DESIGN

The goal of our experiment is to study the feasibility of inferring user's high-level intents through the brain-computer interface (BCI) and use user's intents to authorize access to protected resources. More specifically, we want to assess whether the event-related potentials (ERPs) recorded using a consumer-based EEG headset could be used to infer three types of high-level common tasks: (1) taking a photo, (2) taking a video, and (3) pick a photo from library. The hypothesis to be tested is:

Hypothesis. *Visual and mental processing of each unique intention has distinguishable patterns in event-related potentials that can be extracted with a supervised machine learning algorithm.*

4.1 Single App Experiment

We designed a special Android app (Figure 4) to test our hypothesis. This app consists of three steps. The main activity (Figure 4a) contains 10 `TASK` buttons to start 10 sets of tasks. The tasks are randomized in each set. Before starting each session, participants will click the `START` button to begin logging all the click events into a text file. In each session, participants are asked to go through all 10 sets of tasks. Clicking on each `TASK` button will lead to the task option screen (Figure 4b and Figure 4c). Here participants are asked to perform 4 actions. When an action is finished, participants will return to the same task option screen and continue to the next task until all 4 actions are done. Then they move on to the next task set. When all 10 sets of tasks are completed, participants will click the `STOP` button to stop the session and take a break before starting



(a) Main activity **(b) Task options 1** **(c) Task options 2** **(d) Task activity**

Figure 4: Android app for data collection.

another session. Among these 4 tasks, three involve accessing user-owned privacy sensitive sensors (camera and microphone) and files (photo gallery). The order of these four tasks is different between different task sets. Details of the 4 tasks are listed below.

- **Take Photo** (`Photo`) Clicking this button will send a `MediaStore.ACTION_IMAGE_CAPTURE` intent, to start the camera app. As the name suggests, participants are then asked to take a photo of a target object (*e.g.*, a pen Figure 4d) with the camera app. This task requires access to the camera device.
- **Take Video.** (`Video`) Similar to taking a photo, clicking this button will send a `MediaStore.ACTION_VIDEO_CAPTURE` intent and invoke the camera app. Participants are then asked to take a short video of the target object. The differences from taking a photo are (1) taking a video will access both the camera and the microphone device and (2) accessing to both devices are continuous.
- **Choose from Gallery.** (`Gallery`) Clicking this button will send an `Intent.ACTION_GET_CONTENT` intent with `image/*` type. Participants are then asked to pick the photo of the target object (*e.g.*, a pen) from the photo gallery of the Android device. To make sure the photo is always available, we do not use this task as the first option of the first task set. This task requires access to the privacy-sensitive files.
- **Cancel.** Cancel is a unique task, it does not perform particularly interesting operations or access any privacy-sensitive resources. Its sole purpose is to ask the participants to click a button on the touchscreen of the phone.

Alternative Explanations. An important part of this experiment design is to rule out a few alternative explanations (AE). Specifically, as our experiment involves asking participants to perform a task using the smartphone, we want to rule out the possibility that what we captured from the neural signals is *not* the user's intent to perform the given task but

- **AE1**: The intent to interact with the phone (*e.g.*, click a button).
- **AE2**: The intent to click a specific position of the touch screen (*e.g.*, a button at a fixed position).
- **AE3**: The reaction of seeing similar pictures.

We added the `Cancel` task so if AE1 is true, we will not be able to distinguish the `Cancel` task from the rest tasks. We randomize the order of the tasks on the options activity so if AE2 is true, we will not be able to distinguish between randomized tasks. We deliberately choose three visually similar tasks so if AE3 is true, we will not be able to distinguish between these tasks that involve the same photo.

Table 1: The list of Apps used in testing phase: we test the performance of the model built on the neural data collected from the *in-house* android app in correctly identifying the intention of the users when they interact with these real apps.

App Name	Actions
Facebook Messenger	`Photo`, `Gallery`
Google Hangouts	`Photo`, `Gallery`
WhatsApp	`Photo`, `Gallery`
Instagram	`Photo`, `Gallery`
Camera	`Photo`, `Video`
VideoCamDirect	`Video`
QuickVideo	`Video`
SnapChat	`Gallery`

4.2 Multiple Apps Experiment

For testing the "portability" of the learned model (*i.e.*, the model can identify the same intent across different apps and contexts), we designed a second experiment with eight popular real-world apps (Table 1). All of them have more than 500k downloads in the Google Play Store. We created testing accounts for WhatsApp, Hangouts, Messenger, Snapchat and Instagram. The other three apps Camera,

QuickVideo, and VideoCamDirect did not need any account to take photos or videos.

We instructed participants to browse these apps as they use it in their real-life (e.g., they might be taking a photo, or writing texts). However, in this study, we just focus on the participants' interaction events related to the following three tasks: (1) taking a photo, (2) taking video, and (3) select and upload a photo from the gallery. This experiment has more realistic and ecologically valid settings as the participants were browsing these popular apps and performing the common tasks (*e.g.*, take photo, take video and upload photo) as per their own choice.

4.3 Experimental Procedures

Ethical and Safety Considerations. Our study involved human subjects, and our experimental and recruiting procedures were approved by the Office of Research Integrity (ORI) at the University of California, Riverside under UCR IRB-HS 16-210. All participants were given the option to withdraw from the study at any point of the time. Devices involved in the study were sanitized after each session to avoid skin problems (*e.g.*, irritation). The standard best practices were followed to protect the confidentiality and privacy of the participants data. Compensation of $30 was provided to the participants whether they withdrew or not.

Participants Recruitment. After obtaining the IRB approval, we recruited a total of 41 healthy participants for our experiments. Among the 41 participants, 33 participants were for single app experiment and 8 participants were for multiple app experiment. Participants were recruited by word of mouth, flyers, and social media (Facebook) advertising. Informed consent and some non-personally-identifiable data (gender, age, and major) were obtained from all participants. Twenty-seven (65.85%) of the participants were male, and Fourteen (34.15%) were female. The details on the participants' demographics are provided in Table 2.

Experiment Setup. The experiment consists a consumer-grade EEG headset (Emotiv EPOC+), an Android phone (Google Nexus 5X), an experiment app (§4.1), a laptop, and the Emotiv software package [21]. Participants are asked to use the app on the Android phone while wearing the lightweight EEG headset. The EEG headset connects to laptop and sends EEG data via a Bluetooth dongle. The Android phone connects to the laptop via USB. To construct the ERPs, the Android app records the timestamp of the task. Clocks of the phone and the laptop are synchronized with network time to precisely align the event time stamps and the EEG data. EEG data is recorded using the Emotiv Pure.EEG software.

Testbed. Our testbed is based on Android. To ease the creation of ERP, in the experiments, we use touch events as the anchor to distinguish different ERPs. In particular, we developed a standalone monitoring app which uses the accessibility service in Android to capture all the touch events (using the `flagRetrieveInteractiveWindows` flag) [26] and log the timestamps of the events and the target GUI element. The logged timestamps are then used to synchronize with the neural signals captured by the BCI device and generate ERPs corresponding to the touch events. To label ERPs, we manually label GUI controls with corresponding intents (similar to access control gadget). If a monitored touch event triggers a labeled GUI control, we tag the ERP with the corresponding intent.

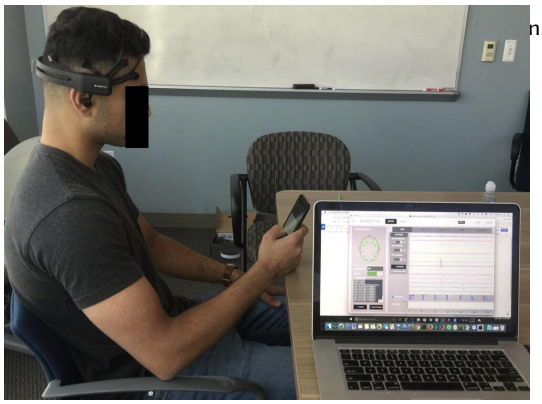


Figure 5: Experiment setup user is playing android apps while wearing the Emotiv Epoc+ BCI headset. The sensors of headset captured neural signals, converted to digital form and transmitted encrypted data to the neural data collection computer via USB dongle receiver.

Preparation Phase. The first step of the preparation is to inform participants that their brain signals would be collected while playing our app on our test Android device and will be used to improve the access control model. Next, we sanitize the electrodes of the EEG headset and apply gel on them to improve their connectivity with the skin. Then we set-up the EEG headset by putting it on the head of the participant. Because the signal-to-noise ratio is lower in raw EEG data, additional preparation steps are followed to ensure the quality of the data. First, all experiments were conducted in a quiet meeting room reserved for one participant only (Figure 5).

Table 2: Participants Demographic Distribution Summary

Gender (%)	
Male	65.85
Female	34.15
Age (%)	
18-21 years	39.03
22-25 years	24.39
26-29 years	29.26
≥ 30 years	7.32
Background (%)	
Computer Science	31.70
Bioengineering	9.74
Biology	4.87
Psychology	7.32
Linguistic	2.44
Business	7.32
Political Science	7.32
Mechanical Engineering	2.44
Economics	7.32
Public Policy	2.44
Anthropology	2.44
Gender and Sexuality	2.44
Toxicology	2.44
Medical Science	2.44
Undeclared	7.32

Second, a preprocessing step is carried out on the raw EEG data to increase their signal-to-noise ratio. During preprocessing, noise reduction is applied to each of the raw EEG channels. To ensure all the signals from the electrodes were properly channeled, we checked the Pure.EEG control panel [21]. With the help of this tool, we can validate the signal strength of each channel (electrodes). The color green against the channel in the control panel meant good strength while black meant no signal.

Task Execution Phase. Before starting the data collection, the operator verbally instructed to the participants about the procedure of experiments. For the single app experiment, all participants performed the same set of tasks for 5 sessions, where each session includes performing all 10 sets of tasks (Figure 4a); so a total number of 200 actions (trials) were performed by each participant if without doing any mistake. All sessions were performed on the same day and in the same room. A break of 2-4 minutes was given to participant between each session. Users were instructed to stay calm and relax in the entire session of the experiment. In real life, participants may not face close to 40 actions within a short time (∼ 5 min). However, multiple trials are the fundamental requirement of most ERP-related study [40, 58]. We conducted this single app for proving the ground truth of IAC. For multiple app experiments, participants interacted with 8 popular apps for the entire time of the experiments. They were instructed to play those apps for approximately 25 minutes. The operator notified the participants to stop the browsing after 25 minutes. However, the participants were allowed to stop the session if they were feeling uneasy or bored. On average, the session duration for this experiment was 21 minutes. After finished the experiment, if the participant is interested about our study, we explained the details of our experiment to those curious participants.

5 DATA PROCESS AND ANALYSIS

Figure 2 depicts the work flow of our system. First, we acquire the neural data using the EEG device. Then the raw EEG data is preprocessed to make it usable for the classifiers. Next, we apply Independent Component Analysis (ICA) to recover original signals from unknown mixtures of sources and extract features using autoregressive coefficients. Finally, we utilize machine learning (ML) techniques to get the intent.

Raw Data Acquisition. We collected raw EEG data using the Emotiv Pure.EEG software [21]. We synchronize the EEG data with actions (*i.e.*, click events received by the app) using calibrated clocks on the phone and the laptop. Based on the study of Martinovic *et al.* [40] and Neupane *et al.* [44], we epochize the signals with 938 ms window which starts at 469 ms before a touch event and 469 ms after the event. We chose this window size as it provides the best results during our analyses. Similar to the previous works [40, 44], we also consider the window before the touch event because participants know beforehand which action they will perform; so the stimuli session actually starts before the event is recorded.

Data Preprocessing. Neural activities of human involve a huge number of neuronal-membrane potentials. EEG records the voltage change of cerebral tissues and the state of brain function. However, these signals are weak, non-stationary and nonlinear in nature [6]. For this reason, EEG signals can easily be contaminated by external noises like the frequency of the power supply and noise generated by the human body, such as eye movements, eye blinks, cardiac signals, muscles noise, etc. The most significant and common artifact produced by eye movements and blinks is known as electrooculogram (EOG). Electromyography (EMG) is another type of contaminating artifact, which is a measurement of the electrical activity in muscles as a byproduct of contraction. EMG artifacts are much more complex than EOG artifacts due to the movement of muscles, particularly those of the neck, face, and scalp. Both EMG and EOG seriously degrade the extraction of the EEG signals and lead to incorrect analyses. Hence they must be removed from the raw data. Similar to previous work [3, 53], we used the AAR (Automatic Artifact Removal) toolbox [27], which utilizes the Blind Source Separation (BSS) algorithm to remove both EOG and EMG [35]. After removing the EOG and EMG artifacts, we applied an 8th order Butterworth band pass filter with a cutoff frequency of 3-60 Hz to remove all other useless signals. The band pass filter keeps signals within the specified frequency range and rejects the rest. The selected frequency range covers all five major frequency bands in EEG signal, namely delta (0.1 to 4 Hz), theta (4.5 to 8 Hz), alpha (8.5 to 12 Hz), beta (12.5 to 36 Hz), and gamma (36.5 Hz and higher) [19]. This preprocessing step extracts quality signals with good SNR (signal-to-noise-ratio).

ICA. Independent Component Analysis (ICA) is standard method to recover original signals from known observations where each observation is an unknown mixture of the original signals. EEG device has 14 electrodes for receiving the brain signals from different regions of the brain. Typically, each sensor will receive signals from a mixture of regions. ICA can be applied to separate independent sources from a set of simultaneously received signals from different regions of human brain [31, 32, 64]. In this study, we used ICA to separate multi-channel EEG data into independent sources.

Feature Extraction. The features from neural signals are extracted using autoregressive (AR) model. This model is a popular feature extraction method for biological signals, especially for time series data [15]. It can estimate the current values $x(t)$ of a time series from the previous $x(t-1)$ observations of the same time series. The current term $x(t)$ of the series can be estimated by a linear weighted sum of previous term $x(t-1)$. A generic formula for representing the time series data (*e.g.*, EEG) is

$$x(t) = \sum_{i=1}^{n} \alpha_i x(t-i) + e(t) \quad (1)$$

Where α_i, is weight which also known as the autoregressive coefficients, $x(t)$ is the EEG signal, and n is the order of the model, indicating the number of previous data points used for estimation. $e(t)$ is called noise or residual term which is assumed to be Gaussian white noise. $x(t)$ measured in time period t.

The selection of order in AR is the crucial step for getting a successful application. We chose AR order six like previous studies [5, 44, 50]. All these studies used the 128Hz Emotiv EPOC device. We calculated AR coefficients using the Yule-Walker method [22]. We consider all 14 channels data for our analysis. Therefore, six AR coefficients were obtained for each electrode channel, resulting in

84 (14x6) features for each action of data. The total process of extracting feature applied all the actions for both of the experiments.

Classification Models and Evaluation Metrics. In this study, we used random forest (RF) [12] because our extracted features (autoregressive coefficients) are suitable for RF algorithms [8, 24]. For implementation, we used the Weka classification software package [29].

We evaluate IAC using the weighted average of *Precision*, *Recall* and $F-Measure$. A higher weighted average *Precision* value indicates less false positives (*i.e.*incorrectly authorize access to sensitive data and sensors). A higher weighted average *Recall* value indicate less false negatives (*i.e.*unnecessarily prompt users for authorization). The weighted average $F-Measure$ is the weighted average of *Precision* and *Recall* which takes both false positives and false negatives into account and gives the balance of our machine learning model. Finally, we used k-fold cross validation to validate our results, where $k = 10$. This is a broadly used technique for calculating test accuracy in the classification problem for small sample which can prevent overfitting. The goal of our study is to train a classifier which can be used to predict user's intent based on features that extracted using earlier step.

6 FEASIBILITY TEST

In this section, we aim to answer the research questions through analyzing the data we collected from the two *different* experiments described in §4. We start from **Q1**—*is it possible to distinguish the three high-level intent based on neural signals using machine learning algorithm.*

6.1 Single App Analysis

Recall that our single app experiment includes 5 sessions for each participant, where each session includes 10 sets of tasks and each task set includes 4 actions. Therefore, each participant has 50 instances per action (5 sessions x 10 task sets). In total, we have 1650 instances (50 instances x 33 users) per action from all 33 participants in the single app experiment. We then extracted features from these instances using the methodology discussed in §5 and labeled the feature vectors with the following four actions as classes:

- ***Camera*** for the task of taking photo action,
- ***Video*** for the task of taking video action,
- ***Gallery*** for the task of choosing a photo from gallery, and
- ***Cancel*** for canceling the pop-up.

Global Model. In this model, we consider dataset of all the users with all the sessions. We have total 6600 (1650 instances x 4 actions) ERP events for this model. The experiment results of this model are shown in Table 3. As shown in the table, the weighted average of *Precision* is 70.70%. This implies that our IAC can correctly detect human intention for 70.70% of time, which is not very good for automated authorization. The reason behind this relatively low accuracy is that even for the same task, different people are likely to have different ERPs patterns, which actually has been used to build authentication systems [5, 63]. For this reason, we would like to know how the classifier performs when *only* consider actions belong to the same participant.

Table 3: Classification result of global model.

Metrics		
Precision	*Recall*	$F-Measure$
70.70%	70.70%	70.70%

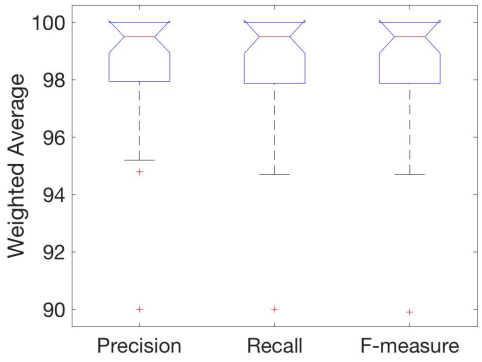


Figure 6: Boxplot of *Precision*, *Recall*, and $F-measure$ of individual model. The red line indicates the median value and + symbol indicates the outliers.

Individual Model. In the individual model, we train and test the model with data from a single user across all sessions of single app experiment. The results for the *individual model* are reported in Figure 6. Overall, the results were much better than when considering all segments across all participants (*i.e.*, the global model). From the boxblot, we observed that the median of weighted average of *Precision* and *Recall* are 99.50% and 99.50%, respectively. The median of weighted average $F-measure$ is 99.50% also. These results imply that IAC correctly detect human intent for 99.50% of the time. The results also indicate that IAC works well when the ML model is trained and tested with a single user and a single app.

6.2 Cross-app Portability Analysis

Through the single app experiment, we partially verified that it is possible to infer users' high-level intents based on their brain signals. In terms of app context, this implies that our classifier can distinguish different app contexts. However, since it only involves one app, the remaining questions is: *can the learned model work across different apps?* That is, in terms of app context, we want to know whether our classifier can identify *similar* context from different apps (*i.e.*, cross-app portability).

We answer this question using the multiple real-world apps experiment where 8 participants interacted with 8 real world apps with a duration of 21 minutes on average. However, we had to discard 3 participants data due to the device error caused data loss. So we only consider those 5 participants whose data is sufficient. On average, the 5 participants performed 22 actions for video, 47 actions for camera, and 27 actions for gallery. In total, we have 484 ERPs from 5 users.

Because these 5 participants have *not* participated in the single app experiment, this experiment resembles a more practical scenario. With this setup, we have two options to bootstrap the

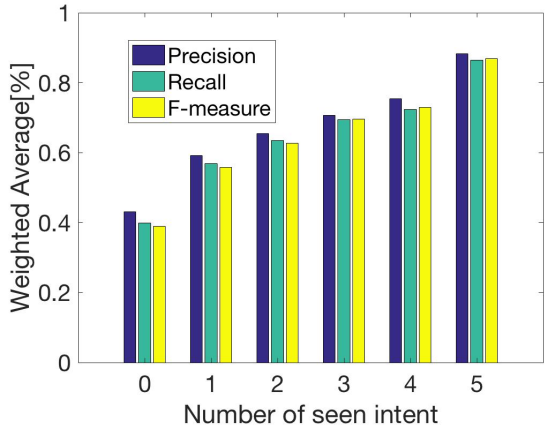


Figure 7: How classification metrics varies with the number of seen intents? The first bar represents the $Precision$, $Recall$, and $F-measure$ without adding any new intents from multiple real world apps experiment to the global model from single app experiment. The second bar represents results with adding new intents to the global model, The third bar represents the results after adding two intents to the global model, and so on and so forth. We observed the upward trends of $Precision$, $Recall$, and $F-measure$ with the addition of more new intents to the global model.

individual model: (1) we can start with an *empty* model can completely rely on the feedback loop (in Figure 2) to collect training data; or (2) we start with a *half-baked* model and use the feedback loop to improve it. In this experiment, we chose the second option as it requires less training and the global model we tested in §6.1 still showed reasonable accuracy.

With Initial Model. We used the global model learned from all participants in the single app experiment as the initial model (*i.e.*, train the model with all data in the single app experiment) and tested it with all data collected from the multiple app experiments. The classification results of *Precision*, *Recall*, and $F-measure$ of the initial model are presented in the first bar diagram in Figure 7. From this figure, we can observe that we can only correctly infer the user intention with the precision of 43.16%.

Adding Feedback Loop. When we gradually add new training intents collected from the user when he/she is using real world apps, the improvement on *Precision*, *Recall*, and $F-measure$ are shown in Figure 7. All newly added intents were from the multiple app experiment and we have to stop at 5 so we can have enough data for the testing phase. As we can see, after adding 5 intents from real world apps, the weighted average *Precision* improved from 43.16% to 88.34%, the weighted average *Recall* improved from 39.82% to 86.52%, and the weighted average $F-measure$ improved from 38.94% to 86.92%. The results imply that in real world context, IAC can correctly infer the user intention 86.92% of time by adding only 5 intents to re-train the ML model. Again, the precision is expected to continue improving and the only reason we stop at 5 is due to lack of data.

6.3 Results Analysis

Based on the classification results from above experiments, we decided to accept our hypothesis. That is, *it is possible to identify high-level intents based on neural signals using a machine learning algorithm.* In terms of app context, our classifier can both distinguish different contexts from the same app and identify similar contexts from different apps. Hence, the answer to **Q1** is positive.

6.4 Authorization Accuracy

In the above analysis, we have shown that it is possible to identify user's high-level intent through the brain-computer interface. However, whether the classification result can be used for automated access authorization for user-owned sensitive sensors and resources still faces the question: is it accurate enough (**Q2**). In this subsection, we analyze the classification results to answer this question. From the analysis of multiple app experiment data, we observed that our classifier can achieve a weighted average of *Precision* 88.34% with the weighted average of $F-measure$ 86.92% for the completely unknown scenarios. Based on this, we think the answer to **Q2** is positive.

7 DISCUSSION

IAC and Contextual Integrity. Access control system is a mechanism to protect user's privacy. Modern OS, including Android (M+), iOS, and Windows (8+) uses an ask-on-first-use permission system to guard access to sensitive data and sensors. This approach provides some context cues but only at the first time when the permission is requested. Researchers have argued that permission should be requested under the context that matches user's expectations, *i.e.*, contextual integrity [46]. IAC enforces contextual integrity in the way that user would only have an intent in her mind when the context is relevant to the intent. In other word, if an app violates contextual integrity, then the user will not express the intent and IAC will block the access.

Learning Strategy. As demonstrated in §6, the classification accuracy can vary based on the learning strategy. Overall, since different people may exhibit different brain signals even when thinking about the same thing (which has been used for neural-signal-based authentication); it is preferable to use individual models. However, bootstrapping such a model require users to go through a calibration phase. An alternative approach, as used in [66] and our own experiment, is to use a half-baked model (*e.g.*, the generalized model learned from all participants in the single app experiment), then personalized it by adding feedbacks from explicit prompts, especially for newly installed apps. Once the model has seen enough feedback, we can start using it to make real authorization decisions. Our multiple app experiment has partially validated the effectiveness of this strategy.

Limitations. Similar to other previous studies on BCI [40, 45], our study also has several limitations. First, the study was conducted in the controlled environment so whether unwanted artifacts like EOG and EMG can be reliably removed in an uncontrolled environment is still unclear. However, since this is a common problem for BCI, we believe future techniques will be able to address it. Second, despite that our sample set is relatively larger (41 participants) than previous studies (*e.g.*, 5 participants [5, 49], 9 participants [39], 16 participants [7]) and have diverse demography background, it is still much smaller than data set in other machine learning applications, such as computer vision, voice recognition, and natural

language processing. Third, we used only popular apps for testing our feasibility and the number of apps is only 8. This could be a bias scenario as participants are more familiar with popular apps. Finally, our classifier is likely to be vulnerable to phishing-style attacks. That is, similar to following our instructions to perform actions that would allow an app to access protected resources, a phishing-style attack might also be able to trick users into willing to perform operations that would compromise the security and privacy of their data.

Future Work. There are many unexplored areas along this research direction. First, we would like to explore other machine learning algorithms like deep neural network (DNN) to see if it can help improve the classification accuracy. Second, we would like to see if the classifier can scale to support more types of tasks and how the accuracy would look like. Third, we would also like to explore if it is possible to improve the classifier by including other behavioral information, such as eye gazing information. Moreover, although our current design might be vulnerable to phishing-style attacks, previous study [45] has shown that even though at conscious level, users may not realize the difference between phishing and non-phishing websites, their neural signals still differs. Based on this observation, we would like to explore the possibility of defending against phishing-style attacks at brain signal level. Finally, recent research has shown machine-learning-based classifiers may be subject to adversarial examples [28], so might be our classifier. However, it is unclear that under our threat model, how attackers can tamper with the collected EEG data to inject their malicious perturbations. So we would also like to explore this direction.

8 RELATED WORK

In this section, we briefly discuss related work on neural signals and permission model.

BCI-based security studies. Neural signals have used for user authentication [17, 34, 43, 63] and identification [52, 68]. Ashby *et al.* [5] proposed an EEG-based authentication system using a consumer grade 14-sensor Emotiv Epoc headset. Abdullah *et al.* [2] discussed the possibility of the EEG-based biometric system using 4 or fewer electrodes. Chuang *et al.* [17] developed a user authentication model using one single-sensor Neurosky headset. Campbell *et al.* [14] developed a neurophone which is based upon ERP of brain signal. They implemented a brain-controlled address book dialing app, which shows a sequence of photos of contacts from address book to users. Thorpe *et al.* [63] suggested pass-thoughts to authenticate users. In their study, they used EEG signal to replace password typing. The EEG-based authentication system overcomes the weakness of current authentication protocol which suffers from several types of attacks including dictionary attack, password guessing, etc. However, there are some drawbacks to this approach like non-pervasiveness of EEG equipment and lack of feedback to the users during the authentication process.

Exposing user's neural signals to third-party apps via the brain computer interfaces introduced new security and privacy issues [11, 25, 40, 44]. Martinovic *et al.* [40] introduced a side-channel attack which they referred to as "brain spyware" using commercially available headset Emotiv EPOC. The authors extracted private information like familiar banks, ATMs, PIN digits, and month of birth using only brain signal. Their work is similar to Guilty-KnowledgeTest (GKT) [18] where familiar items evoked a different response than unfamiliar items. In their experiment, users are shown images of banks, digits, known people images. The users' ERP responses will be different for their very known banks as that information stored their memory beforehand. However, their attack is intrusive and can be easily detectable as the users may notice the abnormality in the application when it displays some of their familiar information sequentially. Frank *et al.* [25] proposed a subliminal attack in which attacker can learn relevant private information from the victim at the levels below his cognitive perception. Bonaci *et al.* [11] showed how non-invasive BCI platforms used in games or web navigation, can be misused to extract user's private information. Neupane *et al.* [44] showed the feasibility of stealing users' PIN from their brain signals.

Runtime Permission Models. Requesting access to sensitive resources at runtime—the moment they will be used provide more context information thus can help users better understanding the nature of these requests and make more optimal decisions [23]. The challenge is how to avoid habituation caused by high frequency of resource access [65].

User-driven access control. The first approach to reduce the number of prompts is to automatically authorize the requests based on users' intent. Existing user-driven access control systems [33, 38, 41, 48, 51, 55, 57] all utilize the same way to infer the intent—by capturing *authentic* user interaction with *trusted* GUI gadgets (*i.e.*, access control gadgets), *e.g.*, the "camera" button. Our approach also tries to infer the intent of an user. However, as we directly infer the intent from the neural signals, our system is not vulnerable to GUI attacks [30, 51] thus do not require additional protection for GUI gadgets. Please note that although we only used user-initiated actions in our experiment, unlike existing user-driven access control systems, our approach *is not* limited to user-initiated events. Because any external stimulus, including viewing an app's foreground GUI context can be used to create event-related potentials (ERPs) and drive our system.

Decision prediction. The second approach is to use machine learning (ML) to predict users' privacy decisions [36, 47, 65, 66]. Liu*et al.* [36] proposed using user's answers to a few privacy related questions to build a personalized privacy profile. They then create a Privacy Assistant that offer recommendations for future permission settings based on the profile, apps category, requested permission, and purposes associated with the permission. While they found that 78.8% of the recommendations were adopted by users, the biggest limitation is that they used the ask-on-install model so the recommendations were made without considering context. Recognizing the importance of context integrity, Wijesekera *et al.* [65] pioneered the work on predicting user's privacy decisions based on the context. In their first attempt, they used a one-size-fits-all logistic regression model which can provide 40%-60% better accuracy than random guessing. In [66], they further extended this idea by building a SVM-based classifier based on when context has changed and user's past decisions and behavior. This new approach

improved the accuracy to 96.8% across all users. However, the accuracy drops to 80% among users who truly make different decisions based on context. Around the same time, Olejnik *et al.* [47] also propose using context information and ML technique to predict user's privacy decisions. In this work, they used 32 raw contextual features (*e.g.*, app name, foreground app, method, time, semantic location) to train a linear regression model based on users' previous decisions under different contexts. The mean correct classification rate of their model is 80%. Our approach also relies on ML techniques and our learning strategy is very close to [66]. However, instead of trying to encode context as a set of features to the ML techniques, we rely on users to interpret the context and aim to infer what they want to do under the given context.

9 CONCLUSION

In this work, we proposed a new direction to protect user-owned, security and privacy sensitive sensors and resources—by inferring user's intents and use it to automate authorization decisions. As a first step, we studied the feasibility of leveraging the brain-computer interface to infer the intents. Our experiment with 41 participants showed that neural signals can be utilized to train a machine learning classifier to recognize high-level intents like taking a photo. The accuracy of the classifier was also good enough for this security and privacy sensitive task.

10 ACKNOWLEDGMENT

This research was supported, in part, by NSF award CNS-1718997 and ONR under grant N00014-17-1-2893. The authors like to thank Yue Duan and Ali Mohammadkhan for their feedback on early version of this paper. We also acknowledge Sri Shaila G, Ali Davanian, and Sankha Dutta for the proofreading of the final version of this paper. We also extend thanks to the ACSAC'18 anonymous reviewers for their constructive feedback and comments.

REFERENCES

[1] 2018. Mind-controlled robots: the factories of the future? https://www.youtube.com/watch?v=wXYvuhH_4Uw. Accessed: 02-10-2018.
[2] Muhammad Kamil Abdullah, Khazaimatol S Subari, Justin Leo Cheang Loong, and Nurul Nadia Ahmad. 2010. Analysis of effective channel placement for an EEG-based biometric system. In *IEEE EMBS Conference, Biomedical Engineering and Sciences (IECBES)*.
[3] Mohammad H Alomari, Aya Samaha, and Khaled AlKamha. 2013. Automated classification of L/R hand movement EEG signals using advanced feature extraction and machine learning. *arXiv preprint arXiv:1312.2877* (2013).
[4] Amazon.com, Inc. 2027. Alexa Skill Kit. https://developer.amazon.com/alexa-skills-kit.
[5] Corey Ashby, Amit Bhatia, Francesco Tenore, and Jacob Vogelstein. 2011. Low-cost electroencephalogram (eeg) based authentication. In *International IEEE/EMBS Conference on Neural Engineering (NER)*.
[6] H Aurlien, IO Gjerde, JH Aarseth, G Eldøen, B Karlsen, H Skeidsvoll, and NE Gilhus. 2004. EEG background activity described by a large computerized database. *Clinical Neurophysiology* 115, 3 (2004), 665–673.
[7] Louise Barkhuus and Anind K Dey. [n. d.]. Location-based Services for Mobile Telephony: a Study of Users' Privacy Concerns. In *International Conference on Human-Computer Interaction*.
[8] Maouia Bentlemsan, ET-Tahir Zemouri, Djamel Bouchaffra, Bahia Yahya-Zoubir, and Karim Ferroudji. 2014. Random forest and filter bank common spatial patterns for eeg-based motor imagery classification. In *International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*.
[9] Niels Birbaumer, Nimr Ghanayim, Thilo Hinterberger, Iver Iversen, Boris Kotchoubey, Andrea Kübler, Juri Perelmouter, Edward Taub, and Herta Flor. 1999. A spelling device for the paralysed. *Nature* 398, 6725 (1999), 297–298.
[10] Tamara Bonaci, Ryan Calo, and Howard Jay Chizeck. 2014. App stores for the brain: Privacy & security in Brain-Computer Interfaces. In *IEEE International Symposium on Ethics in Science, Technology and Engineering*.
[11] TLBMT Bonaci, J Herron, and HJ Chizeck. 2015. How susceptible is the brain to the side-channel private information extraction. *American Journal of Bioethics, Neuroscience* 6, 4 (2015).
[12] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
[13] Ahier Brian. 2017. Neuralink, Facebook, and Kernel Compete on Direct Brain-Computer Interface. https://www.linkedin.com/pulse/direct-brain-interface-brian-ahier/. Accessed: 05-10-2017.
[14] Andrew Campbell, Tanzeem Choudhury, Shaohan Hu, Hong Lu, Matthew K Mukerjee, Mashfiqui Rabbi, and Rajeev DS Raizada. 2010. NeuroPhone: brain-mobile phone interface using a wireless EEG headset. In *ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds*.
[15] Chris Chatfield. 2016. *The analysis of time series: an introduction*. CRC press.
[16] Stephen Chen. 2018. China is mining data directly from workers' brains on an industrial scale. http://www.scmp.com/news/china/society/article/2143899/forget-facebook-leak-china-mining-data-directly-workers-brains. Accessed: 04-30-2018.
[17] John Chuang, Hamilton Nguyen, Charles Wang, and Benjamin Johnson. 2013. I think, therefore i am: Usability and security of authentication using brainwaves.
[18] National Research Council et al. 2003. The polygraph and lie detection. Committee to review the scientific evidence on the Polygraph. Division of Behavioral and Social Sciences and Education. *Washington, DC: The National Academic Press. Retrieved* 7, 7 (2003), 09.
[19] Jan C de Munck, Sonia I Gonçalves, R Mammoliti, Rob M Heethaar, and FH Lopes Da Silva. 2009. Interactions between different EEG frequency bands and their effect on alpha–fMRI correlations. *Neuroimage* 47, 1 (2009), 69–76.
[20] EMOTIV Inc. 2017. Emotiv EEG Headset. https://www.emotiv.com. Accessed: 5-17-2017.
[21] EMOTIV, Inc. 2017. EMOTIV PureEEG Software. https://www.emotiv.com/product/emotiv-pure-eeg/. Accessed: 5-17-2017.
[22] Gidon Eshel. 2003. The yule walker equations for the AR coefficients. *Internet resource* 2 (2003), 68–73.
[23] Adrienne Porter Felt, Serge Egelman, Matthew Finifter, Devdatta Akhawe, David Wagner, et al. 2012. How to Ask for Permission.
[24] Luay Fraiwan, Khaldon Lweesy, Natheer Khasawneh, Heinrich Wenz, and Hartmut Dickhaus. 2012. Automated sleep stage identification system based on time–frequency analysis of a single EEG channel and random forest classifier. *Computer methods and programs in biomedicine* 108, 1 (2012), 10–19.
[25] Mario Frank, Tiffany Hwu, Sakshi Jain, Robert Knight, Ivan Martinovic, Prateek Mittal, Daniele Perito, and Dawn Song. 2013. Subliminal probing for private information via EEG-based BCI devices. *arXiv preprint arXiv:1312.6052* (2013).
[26] Yanick Fratantonio, Chenxiong Qian, Simon P Chung, and Wenke Lee. 2017. Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop.
[27] Germán Gómez-Herrero, Wim De Clercq, Haroon Anwar, Olga Kara, Karen Egiazarian, Sabine Van Huffel, and Wim Van Paesschen. 2006. Automatic removal of ocular artifacts in the EEG without an EOG reference channel. In *Signal Processing Symposium (NORSIG)*.
[28] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.
[29] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.
[30] Lin-Shung Huang, Alexander Moshchuk, Helen J Wang, Stuart Schecter, and Collin Jackson. 2012. Clickjacking: Attacks and Defenses.
[31] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. 2004. *Independent component analysis*. Vol. 46. John Wiley & Sons.
[32] Aapo Hyvärinen and Erkki Oja. 2000. Independent component analysis: algorithms and applications. *Neural networks* 13, 4 (2000), 411–430.
[33] Yeongjin Jang, Simon P Chung, Bryan D Payne, and Wenke Lee. 2014. Gyrus: A Framework for User-Intent Monitoring of Text-based Networked Applications.
[34] Benjamin Johnson, Thomas Maillart, and John Chuang. 2014. My thoughts are not your thoughts.
[35] Carrie A Joyce, Irina F Gorodnitsky, and Marta Kutas. 2004. Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology* 41, 2 (2004), 313–325.
[36] Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, SA Zhang, Norman Sadeh, Alessandro Acquisti, and Yuvraj Agarwal. 2016. Follow my recommendations: A personalized privacy assistant for mobile app permissions.
[37] Bin Liu, Jialiu Lin, and Norman Sadeh. 2014. Reconciling mobile app privacy and usability on smartphones: Could user privacy profiles help?
[38] Long Lu, Vinod Yegneswaran, Phillip Porras, and Wenke Lee. 2010. Blade: an attack-agnostic approach for preventing drive-by malware infections.
[39] Sebastien Marcel and José del R Millán. 2007. Person authentication using brainwaves (EEG) and maximum a posteriori model adaptation. *IEEE transactions*

on pattern analysis and machine intelligence 29, 4 (2007).
[40] Ivan Martinovic, Doug Davies, Mario Frank, Daniele Perito, Tomas Ros, and Dawn Song. 2012. On the feasibility of side-channel attacks with brain-computer interfaces. USENIX.
[41] Kristopher Micinski, Daniel Votipka, Rock Stevens, Nikolaos Kofinas, Michelle L Mazurek, and Jeffrey S Foster. 2017. User Interactions and Permission Use on Android.
[42] Microsoft. 2017. Cortana Skill Kit. https://developer.microsoft.com/en-us/windows/projects/campaigns/cortana-skills-kit.
[43] Fabian Monrose and Aviel Rubin. 1997. Authentication via keystroke dynamics.
[44] Ajaya Neupane, Md Lutfor Rahman, and Nitesh Saxena. 2017. Peep: Passively eavesdropping private input via brainwave signals. In *International Conference on Financial Cryptography and Data Security*. Springer, 227–246.
[45] Ajaya Neupane, Md Lutfor Rahman, Nitesh Saxena, and Leanne Hirshfield. 2015. A multi-modal neuro-physiological study of phishing detection and malware warnings. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 479–491.
[46] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Wash. L. Rev.* 79 (2004), 119.
[47] Katarzyna Olejnik, Italo Ivan Dacosta Petrocelli, Joana Catarina Soares Machado, Kévin Huguenin, Mohammad Emtiyaz Khan, and Jean-Pierre Hubaux. 2017. SmarPer: Context-Aware and Automatic Runtime-Permissions for Mobile Devices.
[48] Kaan Onarlioglu, William Robertson, and Engin Kirda. 2016. Overhaul: Input-Driven Access Control for Better Privacy on Traditional Operating Systems.
[49] Ramaswamy Palaniappan. 2006. Electroencephalogram signals from imagined activities: A novel biometric identifier for a small population. In *International Conference on Intelligent Data Engineering and Automated Learning*. Springer.
[50] Ramaswamy Palaniappan. 2008. Two-stage biometric authentication method using thought activity brain waves. *International Journal of Neural Systems* 18, 01 (2008), 59–66.
[51] Giuseppe Petracca, Ahmad-Atamli Reineh, Yuqiong Sun, Jens Grossklags, and Trent Jaeger. 2017. Aware: Preventing Abuse of Privacy-Sensitive Sensors via Operation Bindings.
[52] M Poulos, M Rangoussi, V Chrissikopoulos, and A Evangelou. 1999. Person identification based on parametric processing of the EEG. In *IEEE International Conference on Electronics, Circuits and Systems*.
[53] Md Lutfor Rahman, Sharmistha Bardhan, Ajaya Neupane, Evangelos Papalexakis, and Chengyu Song. 2018. Learning Tensor-based Representations from Brain-Computer Interface Data for Cybersecurity. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer.
[54] Rijin Raju, Chenguang Yang, Chunxu Li, and Angelo Cangelosi. 2016. A video game design based on Emotiv Neuroheadset. In *Advanced Robotics and Mechatronics (ICARM)*.
[55] Talia Ringer, Dan Grossman, and Franziska Roesner. 2016. AUDACIOUS: User-Driven Access Control with Unmodified Operating Systems.
[56] Franziska Roesner and Tadayoshi Kohno. 2013. Securing Embedded User Interfaces: Android and Beyond.
[57] Franziska Roesner, Tadayoshi Kohno, Alexander Moshchuk, Bryan Parno, Helen J Wang, and Crispin Cowan. 2012. User-driven access control: Rethinking permission granting in modern operating systems.
[58] Andres F Salazar-Gomez, Joseph DelPreto, Stephanie Gil, Frank H Guenther, and Daniela Rus. 2017. Correcting robot mistakes in real time using eeg signals. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.
[59] Marc Stiegler, Alan H Karp, Ka-Ping Yee, Tyler Close, and Mark S Miller. 2006. Polaris: virus-safe computing for Windows XP. *Commun. ACM* 49, 9 (2006), 83–88.
[60] Md Sohel Parvez Sumon. 2016. First man with two mind-controlled prosthetic limbs. *Bangladesh Medical Journal* 44, 1 (2016), 59–60.
[61] Shravani Sur and VK Sinha. 2009. Event-related potential: An overview. *Industrial psychiatry journal* 18, 1 (2009), 70.
[62] Desney Tan and Anton Nijholt. 2010. Brain-computer interfaces and human-computer interaction. In *Brain-Computer Interfaces*. Springer, 3–19.
[63] Julie Thorpe, Paul C van Oorschot, and Anil Somayaji. 2005. Pass-thoughts: authenticating with our minds. In *Workshop on New Security Paradigms*.
[64] M Ungureanu, C Bigan, R Strungaru, and V Lazarescu. 2004. Independent component analysis applied in biomedical signal processing. *Measurement Science Review* 4, 2 (2004), 18.
[65] Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. 2015. Android Permissions Remystified: A Field Study on Contextual Integrity.
[66] Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David Wagner, and Konstantin Beznosov. 2017. The Feasibility of Dynamically Granted Permissions: Aligning Mobile Privacy with User Preferences.
[67] Zhemin Yang, Min Yang, Yuan Zhang, Guofei Gu, Peng Ning, and X Sean Wang. 2013. Appintent: Analyzing sensitive data transmission in android for privacy leakage detection.
[68] Qinglin Zhao, Hong Peng, Bin Hu, Quanying Liu, Li Liu, YanBing Qi, and Lanlan Li. 2010. Improving individual identification in security check with an EEG based biometric solution. In *International Conference on Brain Informatics*.