UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"AUTENTICAZIONE ORIENTATA ALLO SMARTWATCH UTILIZZANDO MULTIPLI SENSORI"

Relatore: Prof. / Dott Migliardi Mauro

Laureando/a: Marcato Francesco

ANNO ACCADEMICO 2021 – 2022
Data di laurea 17/03/2022

Sommario

La presente tesi descrive l'analisi di un nuovo metodo di autenticazione utilizzando i sensori di uno smartwatch, esaminando il modo in cui un soggetto apre una porta. Il lavoro svolto dal laureando Francesco Marcato ha come obiettivo quello di raccogliere i dati sensoristici, in particolare dell'accelerometro e del giroscopio, che lo smartwatch rileva eseguendo l'azione in un certo numero di ripetizioni. Vengono fatte le opportune valutazioni, sulla base di questa raccolta, per stabilire se questa determinata azione può diventare un nuovo metodo di autenticazione da parte dell'utente. L'esperimento viene effettuato su diversi soggetti rendendo possibile l'analisi ed il confronto tra i vari movimenti. Il raggiungimento di tale obiettivo è stato effettuato tramite due linguaggi: Java e MATLAB, rispettivamente per la raccolta ed analisi dei dati.

Ringraziamenti

Un sentito grazie a tutte le persone che mi hanno permesso di arrivare fin qui e di portare a termine questo lavoro di tesi.

Un sentito grazie al mio relatore prof. Mauro Migliardi per la sua infinita disponibilità e tempestività ad ogni mia richiesta. Grazie per avermi fornito ogni materiale utile alla stesura dell'elaborato.

Non posso non menzionare la mia famiglia che da sempre mi sostiene nella realizzazione dei miei progetti. Non finirò mai di ringraziarvi per avermi permesso di arrivare fin qui.

Grazie ai miei amici per essere stati sempre presenti anche durante questa ultima fase del mio percorso di studi. Grazie per aver condiviso incertezze e fatiche e per tutti i momenti di spensieratezza.

Padova, Marzo 2022

Francesco Marcato

Tabella dei Contenuti

1. Introduzione 8
2. Autenticazione 9
2.1. Classificare un'Autenticazione 10
2.2. Valutazione comparativa di un'autenticazione 12
3. Captcha 13
3.1. Cos'è un captcha 13
3.2. Esempi più comuni 14
3.3. Invisible biometric captcha 15
4. Utilizzo dello smartwatch come identity proxy 16
4.1. Importanza del Machine Learning e dell'intelligenza Artificiale 18
5. Il dispositivo utilizzato 20
5.1. Scelta dello smartwatch 20
5.2. Sensori presenti sul Fossil Gen 5 21
6. Ambiente di sviluppo e raccolta dati dei sensori 22
6.1. Ambiente di sviluppo 22
6.2. Raccolta dati 25
6.2.1. Campionamento dei dati dai sensori 26
6.2.2. Problema dell'Ambient Mode 27
6.2.3. Il Dataset 28
6.2.4. Movimenti registrati 28
7. Analisi movimento come possibile autenticazione 29
7.1. Spiegazione dell'esperimento 29
7.2. Risultati 32
7.3. Conclusioni e studio sulla fattibilità 34
8. Analisi del movimento come possibile captcha 34
8.1. Spiegazione dell'esperimento 35
8.2. Risultati 36
8.3. Conclusioni e studio sulla fattibilità 37
9. Conclusioni 38
10. Bibliografia 40

1. Introduzione

Nonostante i continui annunci di nuove soluzioni destinate ad eliminare le password, queste sono ancora la soluzione più diffusa che porta all'autenticazione di un utente.
Le prime password informatiche ed i primi metodi di autenticazione risalgono al 1960 [2], dove servivano ad accedere ad informazioni contenute in sistemi centralizzati.
L'autenticazione è il processo attraverso il quale viene verificata l'identità di un utente che vuole accedere ad un dispositivo o ad una rete. È il sistema che verifica, effettivamente, che un individuo è chi sostiene di essere. Ogni autenticazione suppone che la metodologia utilizzata sia affidabile e che l'identità fornita sia degna di fiducia per tutta la sessione di utilizzo.
Le password sono, come detto prima, ancora oggi lo strumento di autenticazione più diffuso perché nonostante a volte siano disagevoli da utilizzare, non esiste ancora un metodo più semplice ed efficiente. Tuttavia, le password sono considerate molto spesso vulnerabili, con 158 account compromessi ogni secondo nel mondo, l'80% di questi attacchi provengono dalla vulnerabilità delle password.[1]
Negli anni sono stati molti i metodi tramite i quali si è cercato di sostituire le password, è infatti possibile definire tre metodi tramite i quali un soggetto può autenticarsi.
Un'autenticazione può quindi essere basata su:[9]

- qualcosa che so (es. password, frase chiave o pin)
- qualcosa che ho (es. tesserino identificativo)
- qualcosa che sono (es. impronte digitali, impronta vocale, modello retinico, sequenza del DNA, calligrafia o altri identificatori biometrici)

La scelta dei diversi metodi di autenticazione è condizionata da diversi fattori, tra cui l'usabilità, l'importanza delle informazioni da proteggere ed il costo del sistema.
Autenticazioni basate su valori biometrici come quelle che utilizzano l'impronta digitale o riconoscimento dell'iride sono autenticazioni che indentificano un'utente basandosi sulle caratteristiche biometriche del corpo umano, e stanno guadagnando molta popolarità negli ultimi anni.
Tuttavia, esistono metodi di autenticazione che si basano sul comportamento e sui movimenti che vengono rilevati attraverso i sensori presenti in un dispositivo, riuscendo a identificare un utente senza richiedere alcuna operazione extra, tra le autenticazioni più studiate che rientrano in questa categoria sono quelle basate sulla battitura a tastiera [11] o su modo in cui una persona cammina [12].

In questa tesi viene studiata una possibile autenticazione di un soggetto dal modo in cui esso apre la porta, l'esatto movimento che fa utilizzando la maniglia. Per rilevare ed analizzare il movimento, il soggetto in questione sarà dotato di uno smartwatch.
L'idea è quindi quella di analizzare i movimenti che vengono rilevati dai sensori di uno smartwatch quando si apre una porta, e vedere se è possibile usarli per autenticare un utente. Capita infatti molte volte che solo alcuni utenti abbiano accesso ad alcune stanze e per autenticare questi utenti viene solitamente usato un badge.
L'implementazione di un'autenticazione basata sul movimento fatto per aprire una porta potrebbe essere più sicura e ridurre l'accesso di soggetti indesiderati a stanze dove non sono autorizzati, passando quindi da un'autenticazione basata su qualcosa che ho(badge) a qualcosa che sono, poiché vengo identificato univocamente dal modo in cui apro la porta.
Successivamente si verificherà anche se l'atto di apertura di una porta è sostanzialmente diverso da altri movimenti che possono essere effettuati, riuscendo quindi a riconoscere se un utente con lo smartwatch al polso stia o meno aprendo una porta. Quest'ultima analisi potrebbe servire a sventare tutti quegli attacchi hacker che aprono le porte di qualche locale da remoto implementando un sistema di sicurezza che riesca a riconoscere se sia un uomo quello che sta aprendo la porta.
Concludendo, in questa tesi verranno introdotti i concetti di autenticazione e di captcha, di come uno smartwatch può essere usato come un identity-proxy, assieme al dispositivo e all'ambiente di sviluppo utilizzato per svolgere nello specifico gli esperimenti sopra spiegati.

2. Autenticazione

L'autenticazione è il processo che identifica un utente che sta richiedendo accesso ad un sistema, ad una rete oppure ad un dispositivo. Spesso l'identificazione di un utente viene fatta attraverso l'utilizzo di credenziali (username e password), oppure altre tecnologie di autenticazione, che possono essere anche di tipo biometrico.
Un'autenticazione è perciò un metodo che non permette ad utenti non autorizzati l'accesso a informazioni sensibili. Per esempio, solo l'utente A ha accesso a informazioni rilevanti, e non può vedere le informazioni che possono essere sensibili dell'utente B.
Visti i molti tipi di autenticazioni presenti è importante definire delle caratteristiche che ogni autenticazione ha, per poter riuscire a confrontarle.
Quindi, classificare un'autenticazione definendo dei parametri per i quali ogni autenticazione andrà poi confrontata è fondamentale per riuscire a valutare quale autenticazione faccia al caso di ciascun sistema in particolare.
Definire degli standard e dei criteri che aiutino a fare questo è perciò necessario.

2.1.Classificare un’Autenticazione

Il modo in cui qualcuno può essere autenticato ricade in tre categorie, chiamate fattori di autenticazione. Ogni fattore di autenticazione racchiude un insieme di elementi che possono essere usati per autenticare/verificare l’identità di una persona, fornendo quindi l’accesso ad eseguire una qualsiasi operazione.
La prima viene detta "knowledge-based" quindi basata sulla conoscenza del soggetto che deve essere autenticato, è perciò qualcosa che sai come una password o un codice PIN che solo tu, utente identificato, conosci.
La seconda è detta "property-based" quindi basata sul fatto che utente possiede qualcosa come un badge, una chiave o un dispositivo autorizzato che solo tu dovresti avere.
La terza è detta "biologically-based" o biometrica quindi basata su qualche parte del tuo corpo (come potrebbe essere la tua impronta digitale) o l'analisi di un comportamento che è unico per ogni utente (come potrebbe essere il modo in cui una persona cammina).
Tutte le autenticazioni che sono basate su una parte del corpo richiedono di esporre quella parte ad una misurazione e possono essere più o meno invasive salvo avere sensori indossabili. Quelle basate su un comportamento sono più trasparenti e meno invasive in quanto possono avvenire senza che all'utente sia richiesto di fare altro rispetto a quello che deve già fare.
Esiste quindi la differenza tra autenticazione passiva ed attiva. Un'autenticazione si dice passiva se all'utente non è richiesto di fornire alcune credenziali, che siano knowledge-based, property-based o biologically-based, altrimenti si dice attiva.
Ogni autenticazione può essere poi essere classificata anche rispetto alla sua disponibilità. Si può infatti definire tre tipo di disponibilità per ogni autenticazione: bassa, media, alta.
La bassa disponibilità si ha quando è necessario l'utilizzo di un hardware esterno per fare in modo di autenticarsi, oppure se ci sono grandi preoccupazioni riguardo la sicurezza che essa offre, come può essere come la digitazione vocale di una password.
Si dice media disponibilità invece quando ha bisogno di specifici settaggi per funzionare correttamente, senza però extra hardware, come ad esempio nel riconoscimento dell'andatura o di azioni specifiche.
Alta disponibilità invece è definita quando l'autenticazione funziona senza alcun tipo di aiuto esterno, come può essere un PIN.
Un'altra caratteristica fondamentale quando si cerca di classificare un’autenticazione è sicuramente la sua invasività. L'invasività di un sistema può essere definita come quanta

interazione l'utente deve fare e fin quando questa iterazione risulta tollerabile. Anche questa può essere classificata in bassa, media ed alta.

Bassa invasività significa che il sistema non ha bisogno di alcuna iterazione con l'utente per funzionare, un esempio può essere le autenticazioni basate sul movimento.

Un sistema a media invasività invece ha bisogno di una piccola e contenuta interazione come l'inserimento di un pin o di una gesture.

Un'alta invasività infine ha bisogno di operazioni non comuni solitamente scomode da usare in pubblico e velocemente, un esempio può essere descritto in [17] dove si richiede all'utente di disegnare un cerchio in aria con il polso.

Per ogni autenticazione è comunque importate definire e valutare tutti le tipologie di errore che può avere.

Si può quindi definire il risultato di ciascuno dei tentativi di autenticazione:

$$TP = True\ Positive$$

Un vero positivo (TP) è un tentativo di autenticazione avvenuto correttamente, e che autentica un utente con la sua vera identità.

$$FN = False\ Negative$$

Un falso negativo (FN) è un tentativo di autenticazione dove l'autenticazione non avviene anche se avrebbe dovuto autenticare correttamente l'utente, anche detto "Miss".

$$FP = False\ Positive$$

Un falso positivo (FP) è un tentativo di autenticazione che avviene correttamente anche se non dovrebbe (es. sbaglio password ed accedo comunque), è anche detto "False".

$$TN = True\ Negative$$

Un vero negativo (TN) è un tentativo di autenticazione che non avviene proprio come è giusto che succeda. È poi importante poi definire le varie frequenze d'errore:

$$False\ positive\ rate = \frac{FP}{N_n}$$

N_n= totale autenticazioni che dovrebbero fallire

$$False\ negative\ rate = \frac{FN}{N_p}$$

N_p= totale autenticazioni che dovrebbero avvenire correttamente

Come le frequenze d'errore anche a precision ed il recall sono metriche importati legate alle prestazioni rispetto ai dati recuperati ed analizzati.

La precision (precisione) è la frazione di istanze rilevanti tra le istanze recuperate, mentre il recall (noto anche come sensibilità) è la frazione di istanze rilevanti che sono state recuperate. Sia la precisione che la sensibilità si basano sulla pertinenza dei

dati raccolti e quindi la loro importanza può variare a seconda del sistema che si deve analizzare. E vengono definite così:

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

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

2.2. Valutazione comparativa di un'autenticazione

Per agevolare la comparazione delle autenticazioni, in [4] è stato studiato un framework che ci permette vi valutare ogni autenticazione anche in confronto a quelle già conosciute e studiate. La tabella studiata infatti ci permette di trarre delle conclusioni con un minimo sforzo, vedendo i pregi ed i difetti di ogni tipo di autenticazione.
La seguente tabella comparativa, infatti, permette di confrontare direttamente il sistema proposto con i sistemi più usati. Il framework offre una valutazione "UDS" (usability-deployability-security) proprio perché vengono introdotte 3 macrocategorie per le quali ogni autenticazione verrà classificata: Usability, Deployability, Security.
Per ogni voce poi vengo usati dei cerchi pieni, vuoti o nessun cerchio per mostrare intuitivamente i benefici offerti da ogni tipo di autenticazione. Cerchio pieno vuol dire che quell'autenticazione offre quel beneficio, quello vuoto che lo offre parzialmente, mentre se non c'è il cerchio non offre il beneficio.
In figura 1 si può vedere un esempio che confronta i principali metodi usati.

	Usability								Deployability						Security												
	Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-Learn	Efficient-to-Use	Infrequent-Errors	Easy-Recovery-from-Loss	Accessible	Negligible-Cost-per-User	Server-Compatible	Browser-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Throttled-Guessing	Resilient-to-Unthrottled-Guessing	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-Verifiers	Resilient-to-Phishing	Resilient-to-Theft	No-Trusted-Third-Party	Requiring-Explicit-Consent	Unlinkable
Password			●		●	●	◎	●	●	●	●	●	●	●		◎						●	●	●	●
Fingerprint	●	●	●	◎	●	◎			◎			◎			●		●							●	
OTP over SMS	●	●	◎		●		◎	◎	◎			●	●	●	●	●	●	●	◎	●	●	◎		●	●
RSA SecurID					●	◎	◎					●	●		●	●	●	●	●	●	●	●		●	●
Eye-movement	●		●	◎					◎						●	●			●	●	●	●	n/a	●	●
Keystroke	●		●	◎			◎		●	●					◎				●		●			●	

●= offers the benefit; ◎ = almost offers the benefit; no circle = does not offer the benefit.

Figura 1 : Tabella di confronto dei metodi di autenticazione

La tabella presenta 3 macro-parametri: usabilità, implementabilità e sicurezza.

Il primo contiene tutti quei parametri che valutano l'usabilità di un'autenticazione come può essere la facilità d'uso, la necessità o meno di un dispositivo esterno o lo sforzo di doversi ricordare qualcosa.
L'implementabilità invece contiene tutti quei parametri che valutano la possibilità di utilizzo della data autenticazione sul mondo reale determinando quindi la sua fattibilità.
Infine, la sicurezza contiene tutti quei parametri necessari a valutare la sicurezza di ogni autenticazione.
Dalla tabella si può per esempio vedere che autenticazioni biometriche come può essere quella dell'impronta digitale, hanno sicurezza aggiuntiva rispetto alle password, sono ad esempio, resilienti all'osservazione fisica.

3. Captcha

Con il termine captcha si intende un test informatico che ha lo scopo di determinare se un utente sia o meno umano. I primi captcha sono nati nel 1997 quando il motore di ricerca altervista, soffriva di attacchi di bot che inserivano url spesso malevoli al loro motore di ricerca. Da allora i captcha si sono evoluti in molti modi; tuttavia, la loro finalità è rimasta sempre la stessa, impedire a bot di utilizzare determinati servizi, registrazione a siti ecc., in particolare comunque tutte quelle attività che potrebbero costituire spam.
I vari sistemi di captcha non sono quindi nati per assistere ed aiutare un'autenticazione, ma sostanzialmente per dissuadere hacker dall'abusare dei servizi online perché gli impediscono di inviare richieste online false o nefaste.
Tuttavia, i captcha sono anche usati durante un'autenticazione fornendo un controllo in più e rendendola più sicura ed affidabile, specialmente per quei metodi di autenticazione biometrici, un captcha potrebbe essere estremamente utile, visto che la stessa autenticazione si basa su caratteristiche del solo essere umano.

3.1. Cos'è un captcha

Captcha, abbreviazione di "*Completely Automated Public Turing test to tell Computers and Humans Apart*", è un programma pubblico ed automatico che ha il compito di distinguere un utente a seconda che sia un computer oppure una persona.
Solitamente i captcha sono utilizzati per prevenire attacchi hacker su siti web, il test captcha protegge infatti gli utenti dallo spam, chiedendo di superare un semplice test che ha lo scopo di provare che l'utente è una persona.
Generalmente il test proposto è difficile per i computer ma facile per gli esseri umani.

La maggior parte dei captcha sono fatti di numeri, caratteri, che vengono spesso allungati e ruotati, il test consiste nel riconoscimento degli stessi, azione che risulta molto complessa per dei bot.

Tuttavia, visto il recente progresso dell'Intelligenza artificiale in generale, molti programmi automatizzati sono migliorati significativamente nel risolvere questi tipi di test. Come risultato di ciò molti dei captcha tradizionali sono stati dimostrati vulnerabili come in [15] e [16].

Tuttavia, col passare del tempo progettare un captcha efficace ed user-friendly sta diventando sempre più difficile, e questo ha portato all'introduzione di anche nuovi tipi di captcha come quelli basati su l'analisi dei comportamenti e sul rilevamento di alcuni sensori. [14]

3.2.Esempi più comuni

I diversi captcha possono essere divisi in sei principali categorie, a seconda di su cosa sono basati, queste categorie sono:

- Testi
- immagini
- audio
- video
- esercizi matematici
- piccoli giochi

Tuttavia, questa classificazione può considerarsi incompleta visto che non considera molti nuovi captcha come, ad esempio, quello annunciato da Google nel 2014 che con una precisione del 99.8% riesce a distinguere un umano da un bot grazie ad un'analisi comportamentale. Questo è uno dei captcha più usati al momento. [14]

Ora descriveremo tre delle principali categorie di captcha, quelli basati su testo, immagini ed audio.

I captcha testuali possono utilizzare parole o frasi o combinazioni casuali di cifre e lettere, i caratteri sono spesso modificati in modo tale che richiedano un po' di interpretazione. Questa modifica può comportare il ridimensionamento, rotazione o la distorsione.

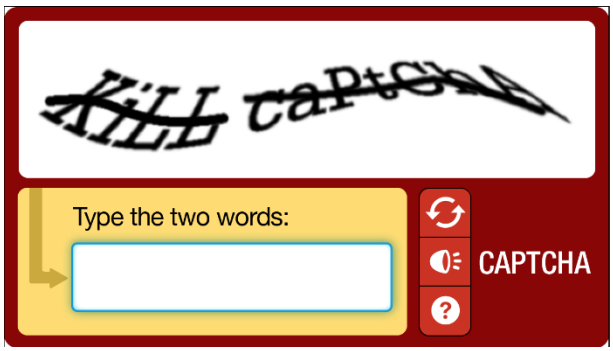

Figura 2: esempio di captcha testuale

Un'alternativa ai captcha basati su testo possono essere quelli basati su immagini. Questi captcha utilizzano elementi grafici riconoscibili come foto di animali, forme o scene. In genere i captcha basati su immagini richiedono agli utenti di selezionare immagini corrispondenti a un tema o di identificare le immagini che non si adattano.

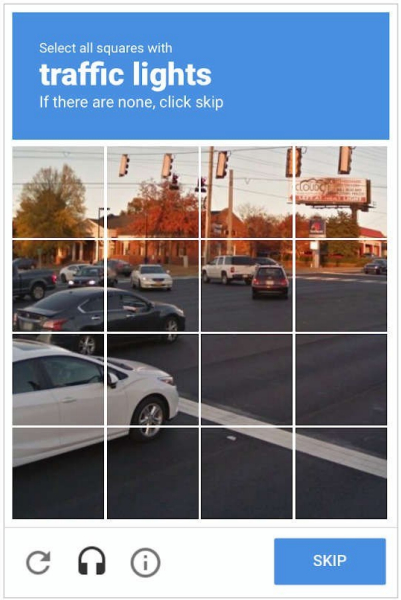

Figura 3: esempio di captcha basato su immagini

I captcha audio sono stati sviluppati come alternativa che garantisce l'accessibilità agli utenti ipovedenti. Questi captcha vengono spesso utilizzati in combinazione con captcha basati su testo o immagini. I captcha audio presentano una registrazione audio di una serie di lettere o numeri che un utente inserisce poi.

Figura 4: esempio di captcha audio

3.3.Invisible biometric captcha

Attraverso la biometria è possibile usare e ricavare un pattern unico caratterizzato da caratteristiche fisiologiche o comportamentali che riescono a identificare degli individui. Solitamente l'approccio biometrico ad un tipo di autenticazione viene considerato più affidabile rispetto a quello tradizionale. [4]

Ci sono molte caratteristiche biometriche che possono identificare un individuo, come l'impronta digitale, il riconoscimento del viso, riconoscimento dell'iride, ecc.[3]

Così come esistono autenticazioni basate sulla biometria di un essere umano, esistono anche captcha basati su di essa.
In particolare, nel nostro caso sarà interessante studiare se il movimento di apertura di una porta può essere usato come un captcha invisibile che conferma o meno l'apertura di una porta da parte di un soggetto.
Il captcha in questo caso avrebbe semplicemente il ruolo di confermare che il soggetto sta aprendo la porta, individuando quindi casi di attacchi hacker che potrebbero aprire la porta da remoto.
Uno dei casi più famosi di hackeraggi di porte [7] ha interessato nientemeno che Google, nel luglio del 2018, un hacker è infatti riuscito ad aprire tutte le porte degli uffici a Sunnyvale senza utilizzare le richieste chiavi magnetiche RFID. Dopo aver mandato un codice malevole nella rete di Google, tutte le porte dell'edificio sono state sbloccate.
Come vedremo nel paragrafo successivo lo stesso smartwatch può essere portatore del RFID che permette di sbloccare una porta, e l'inserire un captcha nell'azione di apertura della porta potrebbe incrementare considerevolmente la sicurezza della autenticazione che avviene in questo caso, distinguendo infatti un bot da un essere umano.

4. Utilizzo dello smartwatch come identity proxy

L'identità digitale non è molto diversa dall'identità che ogni persona ha nel mondo reale; infatti, si tratta pur sempre di rispondere a due domande, "chi sei?" e "cosa fai?".
Ogni sito web, infatti, ha una diversa immagine sulla stessa persona a seconda di "chi è" e da "cosa fa".
Nella maggior parte delle piattaforme è però necessaria un'autenticazione, un modo per riconoscere chi è la persona che sta effettuando ciascuna operazione ed essere sicuri che sia autorizzato ad eseguirla.
Il modo più comune per identificarsi attualmente è sicuramente attraverso l'uso di password. Tuttavia, da tempo si sta cercando di sostituire l'utilizzo di password che sono molto vulnerabili a favore di un approccio un po' più dinamico che riesca a migliorare la user-experience. [4]
Il problema principale delle password è che gli utenti non riescono a ricordare con facilità stringhe di caratteri alfanumerici privi di senso, e sono quindi portati a scegliere parole e numeri prevedibili, oppure vengono scritte per paura di essere dimenticate. A ciò va sommato il fatto che sono vulnerabili da attacchi di varia natura.
Spesso viene infatti utilizzato un identity-proxy che ha lo scopo di identificare una persona.

Un identity-proxy non è altro che un'applicazione o un servizio che ha un ruolo da intermediario, tra l'utente e il fornitore del servizio di identità.
Un identity-proxy fornisce i dati necessari a fornire un'autenticazione quanto più precisa possibile ogni talvolta che un utente deve dimostrare la sua identità rappresentandolo quindi nel mondo digitale. Un identity-proxy, infatti, fornisce le informazioni personali necessarie di cui il servizio ha bisogno.
In questa tesi verrà studiato l'utilizzo dello smartwatch come identity-proxy durante l'atto di apertura di una porta valutando quindi se il movimento può essere usato come autenticazione. Verrà quindi valutata la possibilità di usare questo movimento come possibile autenticazione.
I vari valori biometrici che lo smartwatch può registrare sono di grande interesse visto il relativo basso costo di uno smartwatch e il fatto che sia un dispositivo che può essere indossato e portato con te, oltre al fatto che è in grado di trasmettere dati via bluetooth oppure attraverso una rete wi-fi.
Uno dei principali vantaggi che lo smartwatch offre rispetto ad altri dispositivi è sicuramente il fatto che è sempre indossato nella stessa posizione e orientato nello stesso modo (a differenza dello smartphone). Anche la posizione dello smartwatch è favorevole, visto che nel polso avvengono molto più movimenti, e movimenti più significativi rispetto ad altre posizioni come quella dello smartwatch che è molto spesso all'interno della tasca dei pantaloni.
Le informazioni biometriche di uno smartwatch possono infatti essere alla base di un sistema di autenticazione. Per esempio, lo smartwatch di un soggetto che si avvicina alla sua smart-house potrebbe trasmettere le letture dell'accelerometro e del giroscopio, e confrontarle con delle vecchie letture per infine aprire la porta della casa se queste letture combaciano.
Le informazioni biometriche basate su dei comportamenti potrebbero essere utilizzate in un sistema di autenticazione a più fattori, andando quindi ad aumentare la sicurezza o sostituire i metodi più tradizionali.
Alcuni tra i comportamenti più analizzati e studiati sono: [5]

- andatura
- scrittura a tastiera
- scrittura a mano

Lo smartwatch viene comunque definito come un dispositivo personale, e già ora identifica una persona permettendo di effettuare operazioni come il pagamento contactless tramite NFC.

Lo smartwatch è quindi già molto spesso portatore di un RFID[1], ed i possibili utilizzi di tale tecnologia possono essere molteplici, da pagare al supermercato, a sostituire un badge per accedere a locali e stanze che richiedono un'autenticazione per entrare.

4.1. Importanza del Machine Learning e dell'intelligenza Artificiale

Il problema di ricercare dei modelli (pattern) all'interno dei dati, è un problema fondamentale da molto tempo.

Il machine learning è un metodo di analisi dati che automatizza la costruzione di modelli analitici. È una branca dell'Intelligenza Artificiale e si basa sull'idea che i sistemi possono imparare dai dati, identificare modelli autonomamente e prendere decisioni con un intervento umano ridotto al minimo.

Per intelligenza artificiale si intende la riproduzione parziale dell'attività intellettuale propria dell'uomo (con particolare riguardo ai processi di apprendimento, di riconoscimento, di scelta) realizzata o attraverso l'elaborazione di modelli ideali, o, concretamente, con la messa a punto di macchine che utilizzano per lo più elaboratori elettronici.

L'utilizzo del machine learning, in particolare del pattern recnonition è fondamentale nel caso di un'autenticazione smartwatch-based, o nel gestire un test captcha basato su valori biometrici come il movimento che una persona fa quando apre una porta.

Con pattern recognition si intende l'uso di aloritmi per riconoscere delle regolarità e dei modelli nei dati. Questo tipo di analisi può essere fatta nei più svariati tipi di input come dati biometrici, colori, immagini ecc., grazie a questa flessibilità il pattern recognition è stato applicato in diversi campi.

In figura 5 si può vedere il processo utilizzato per il riconoscimento dei pattern.

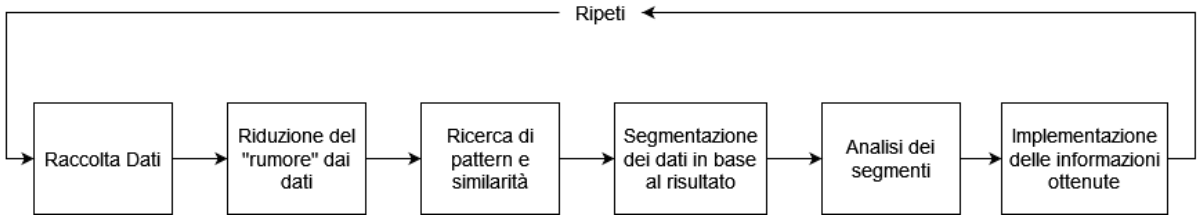

Figura 5: Processo di pattern recnonition

Il pattern recognition può essere di due tipi:

- Supervisionato
- Non supervisionato

[1] RFID è una tecnologia di riconoscimento e validazione e/o memorizzazione automatica di informazioni a distanza. Una delle più recenti implementazioni di RFID è lo standard Near Field Communication (NFC).

La principale differenza tra i due tipi è che nel pattern recognition supervisionato il computer deve riconoscere e mappare un input in un output dati degli esempi di input-output già classificati.
In quello non supervisionato invece non viene fornito alcun esempio di input-output e l'algoritmo dovrà scoprire da solo e naturalmente come classificare i pattern nel data set.
Uno dei processi più utilizzati che risulta fondamentale nel pattern recognition e che verrà usato molto in questa tesi è sicuramente il curve fitting.
Il curve fitting è il processo di costruzione di una curva o di una funzione matematica, che abbia la migliore corrispondenza ad una serie di punti assegnati, possibilmente soggetti a limitazioni.
Tuttavia, effettuando un curve fitting che usare gradi di polinomi anche molto elevati si può ricadere in due tipi di fenomeni che non sono positivi e che possono compromettere la curva che corrisponde alla serie di punti che rappresentano i dati.
Questi fenomeni sono detti: [8]

- Overfitting
- Underfitting

Nell'overfitting ci sono troppi parametri nel modello e un'elevata variabilità della classificazione. Il modello è troppo complesso e sensibile ai dati sulla quale è stato allenato, la figura 6 è un esempio di overfitting.

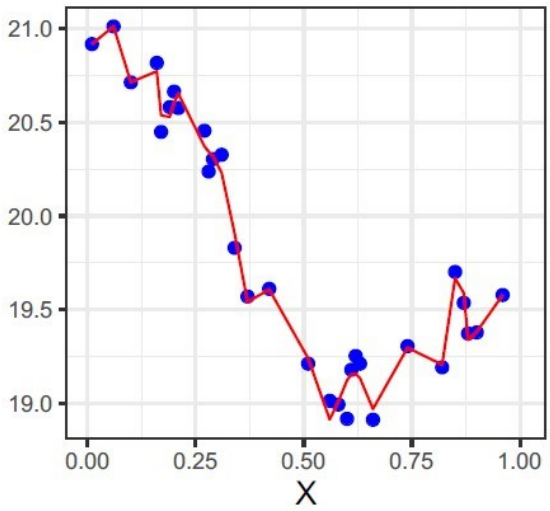

Figura 6: esempio di overfitting

Nell'underfitting ci sono pochi parametri nel modello e un'elevata discrepanza nella classificazione. Il processo di apprendimento è troppo semplice, come su può vedere dalla figura 7.

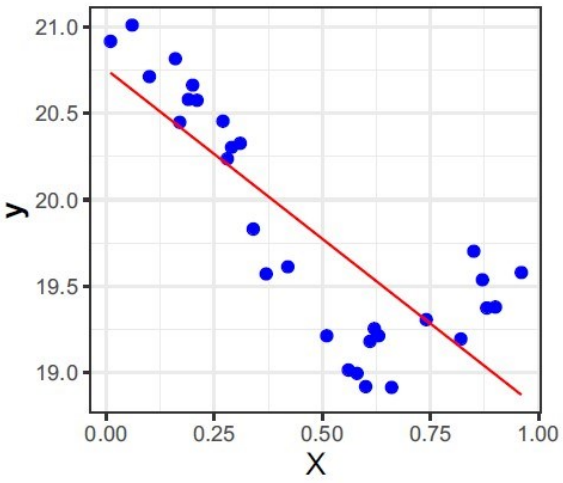

Figura 7: esempio di underfitting

5. Il dispositivo utilizzato

Per studiare e raccogliere i dati necessari a valutare ed analizzare l'autenticazione di una persona dal modo in cui apre la porta oppure l'identificazione del movimento stesso al fine di fornire un test captcha, è stato necessario un dispositivo con vari sensori di medio/alta precisione in modo tale che attraverso l'accelerometro ed il giroscopio si riesca a raccogliere e analizzare i dati rilevanti a raggiungere lo scopo.

5.1.Scelta dello smartwatch

Per svolgere il compito appena descritto si è scelto uno smartwatch, il Fossil Smartwatch GEN 5.

C'è infatti bisogno di uno smartwatch che possa permettere l'analisi accurata, tramite i sensori su esso presenti, dei vari movimenti fatti da un soggetto e valutare la metodologia di autenticazione.

Le principali caratteristiche che, oltre al prezzo, hanno influito sulla scelta dello smartwatch sono le seguenti:

- Quantità e qualità dei sensori
- Ambiente di sviluppo
- Affidabilità

I sensori che saranno principalmente utilizzati e quindi indispensabili sono il giroscopio e l'accelerometro.

Per valutare il prezzo degli smartwatch si è fatto riferimento al sito Amazon.it ed al sito dei produttori dei vari smartwatch, in modo tale da tenere sotto controllo sia le caratteristiche ma anche il prezzo dei vari dispositivi.

Dopo un'accurata ricerca la scelta è stata ristretta ai seguenti tre possibili candidati, con i rispettivi prezzi:

- Fossil Smartwatch GEN 5 (159,20 €)
- Oppo Watch (229,99 €)
- Samsung Galaxy Watch Active 2 (226,98 €)

La figura 8 riassume i sensori e sistemi operativi presenti su ciascun smartwatch:

	Fossil Gen 5	Oppo Watch	Samsung Galaxy Watch Active 2
Sistema Operativo	Wear OS by Google	Wear OS by Google	Tizen OS
Sensori principali	Accelerometro, Giroscopio, Battito Cardiaco, Bussola, Barometro	Accelerometro, Giroscopio, Battito Cardiaco, Bussola, Barometro	Accelerometro, Giroscopio, Battito Cardiaco, Barometro
Comunicazione	Wi-Fi, Bluetooth 4.2, GPS, NFC	Wi-Fi, Bluetooth 4.2, GPS, NFC	Wi-Fi, Bluetooth 5.0, GPS, NFC

Figura 8: Tabella comparativa di diversi smartwatch

I sensori principali sono condivisi da tutti e tre gli smartwatch, sotto questo punto di vista la scelta di uno rispetto all'altro perciò è pressoché indifferente.

Per quanto riguarda l'analisi e la scelta del sistema operativo sulla quale lavorare si ha optato per uno smartwatch con Wear OS by Google.

La principale differenza tra Wear OS e Tizen OS è il fatto che il primo ha un ambiente di sviluppo Java, mentre il secondo C++.

Tuttavia, la scelta di Wear OS deriva dal fatto che vanta una community più ampia ed affermata rispetto a quella di Tizen OS.

Una volta scartato quindi il Samsung Galaxy Watch Active 2 per Tizen OS, la scelta ricadeva tra il Fossil Smartwatch GEN 5 e l'Oppo Watch.

Visto il prezzo inferiore si è deciso di acquistare il Fossil Smartwatch GEN 5.

Sia il Fossil Smartwatch GEN 5 che l'Oppo Watch vantavano buone recensioni, entrambi i dispositivi sono infatti valutati con un 4/5 dagli acquirenti.

5.2.Sensori presenti sul Fossil Gen 5

Fossil Smartwatch GEN 5 è dotato di un accelerometro digitale a 3 assi ad alte prestazioni ed un giroscopio digitale a tre assi.

Monta infatti un LSM6DSO della STMicroelectronics.

Il sensore lsm6dso è tra i migliori sensori per quanto riguarda il rilevamento di movimento; infatti, è in grado di rilevare l'orientamento e le gesture per fornire agli sviluppatori di applicazioni e ai consumatori una funzionalità e una capacità più sofisticata rispetto al semplice orientamento vin modalità verticale e orizzontale del proprio dispositivo.
Gli interrupt di rilevamento degli eventi consentono un monitoraggio del movimento efficiente e affidabile, implementando a livello hardware sistemi di riconoscimento da eventi come:

- Caduta
- Orientamento 6D
- Rilevamento di clic e doppio clic
- Attività o Inattività
- Rilevamento di staticità o di movimento
- Eventi di wake-up.

Il sensore supporta i principali requisiti del sistema operativo, offrendo sensori in modalità reale, virtuale e batch.
Inoltre, il sensore lsm6dso può eseguire in modo efficiente tutte le funzioni basate sui sensori che Android richieda, risparmiando energia ed avendo un tempo di reazione più veloce.
In particolare, il sensore è stato progettato per implementare funzionalità hardware come: rilevamento del movimento, inclinazione, funzioni pedometro, timestamp e supporto nell'acquisizione di dati di un magnetometro esterno.

6. Ambiente di sviluppo e raccolta dati dei sensori

In questo paragrafo verrà spiegato come è avvenuta la raccolta dei dati sensoristici, in particolare come è stata sviluppata l'app che ha permesso la loro registrazione.
Verrà infatti illustrato l'ambiante di sviluppo utilizzato e le varie registrazione effettuate, infine, verrà esposto il dataset usato.

6.1. Ambiente di sviluppo

Allo scopo di raccogliere i dati è stata sviluppata un'app per Wear OS che raccoglie i dati dei sensori e li salva nel dispositivo per poi essere analizzati in un secondo momento.
Per sviluppare l'app è stato usato Android Studio (versione Artic Fox 2020.3.1).
Android Studio è uno dei migliori strumenti per sviluppare applicazioni per Android, ed in particolare per Wear OS.
Android Studio ha una semplice interfaccia che rende lo sviluppo dell'app molto intuitivo, oltre ad offrire anche un emulatore del dispositivo se ce ne fosse la necessità.

Alcune tra le funzioni più utilizzate durante lo sviluppo dell'app che ha reso possibile la registrazione e lo studio dei dati sono sicuramente il sistema di debug che Android Studio offre e il Device File Manager (di cui parleremo successivamente).
Si è dovuto poi scegliere il linguaggio di programmazione, si disponevano infatti di due alternative: Java o Kotlin.
Per una questione di familiarità è stato scelto di sviluppare l'applicazione in Java, anche se Kotlin è un linguaggio di programmazione molto più giovane di Java e che offre un mix di programmazione funzionale ed orientata agli oggetti.
Per quanto riguarda lo smartwatch, il Fossil Smartwatch GEN 5 ha Wear OS 2.33 basato Android 9 e il livello 28 della versione delle API Android.
Per eseguire il debug dell'applicazione sullo smartwatch c'erano due opzioni:

- Debug ADB tramite Bluetooth
- Debug ADB tramite Wi-Fi

Per una questione di facilità d'uso è stato scelto il debug tramite Wi-Fi; infatti, rendere il dispositivo online e pronto al debug dell'app bastava un'istruzione sul prompt dei comandi di Windows, indicando l'indirizzo IP locale che lo smartwatch possiede.
L'istruzione era la seguente:

```
adb connect 192.168.1.241:5555
```

E dopo pochi secondi veniva stabilita la connessione

```
connected to 192.168.1.241:5555
```

Per quanto riguarda il look dell'app, ne è stata sviluppata una con un look molto minimalista come mostrato in Figura 9 e 10, con due semplici bottoni start e stop, che rappresentano l'inizio e la fine del raccoglimento di dati, assieme ad un textView [2]che mostra lo stato nella quale l'app si trova (se sta registrando o meno i dati dai sensori).

[2] Un elemento dell'interfaccia utente che mostra il testo all'utente.

Figura 9: screenshot app mentre sta campionando i dati dei sensori

Figura 10: screenshot app mentre non sta campionando i dati dei sensori

Una volta che l'app ha raccolto i dati e salvati in un csv nel file manager dello smartwatch, questi verranno visualizzati e spostati in un computer tramite il tool di Android Studio "Device File Manager" per poi venire analizzati in un secondo momento.
La figura 11 mostra l'interfaccia del Device File Manager mentre si accedeva ai file presenti sul dispositivo.

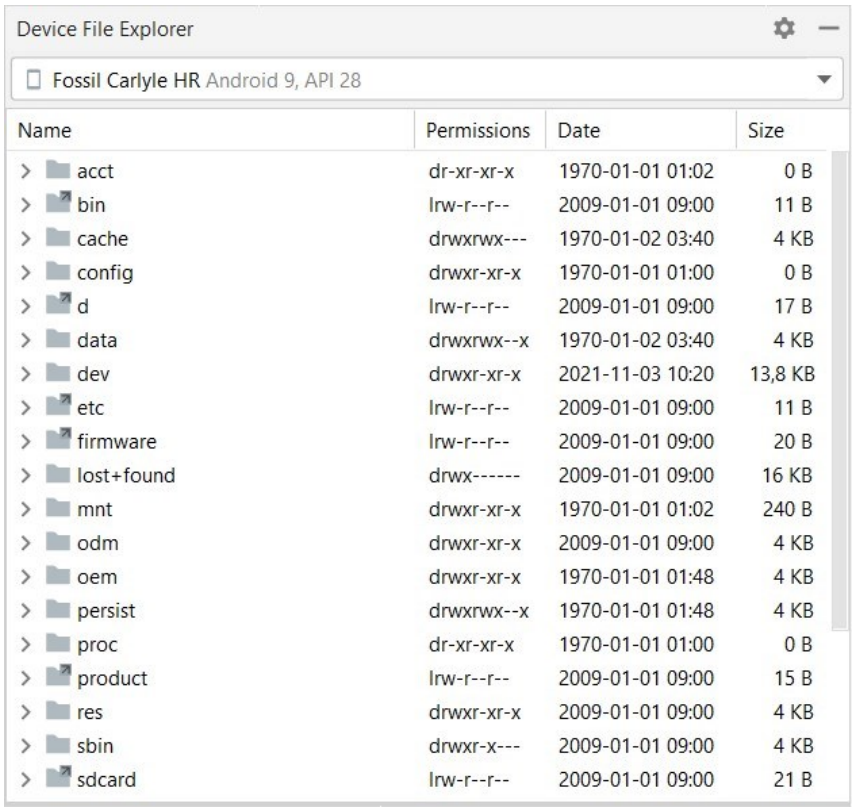

Figura 11: screenshot del tool "Device File Manager"

6.2. Raccolta dati

L'app sviluppata per Wear OS riesce a raccogliere i dati dell'accelerometro e del giroscopio, in modo campionario.

Ogni qual volta i sensori rilevano un cambio di valore in una delle loro coordinate, questo viene salvato.

I dati non vengono registrati direttamente durante l'esecuzione dell'app bensì solo una volta premuto il bottone start, dopo averlo premuto, cominceranno ad essere immagazzinati in un'arraylist e poi una volta premuto il bottone save and stop, questi verranno salvati nella memoria dello smartwatch in un formato di file csv.

Il file csv conterrà tutte le variazioni di valore in tutti i sensori che vengono considerati, e per ogni variazione, il nome del sensore, il momento in qui avviene ed i nuovi valore per ogni coordinata, come si può vedere in figura 12.

Nome	Tempo	X	Y	Z
lsm6dso Gyroscope Non-wakeup	34968850974985	-0.03087921	-0.0075061...	0.00185572...
lsm6dso Accelerometer Non-wakeup	34968869276131	1.4571445	-0.1354583	10.175519
lsm6dso Accelerometer Non-wakeup	34968888776860	1.1076062	-0.49936134	9.98878
lsm6dso Accelerometer Non-wakeup	34968908277589	1.3278632	-0.25516325	9.787676
lsm6dso Accelerometer Non-wakeup	34968927778318	1.5050265	-0.21685766	9.653606
lsm6dso Accelerometer Non-wakeup	34968947279047	1.4762974	-0.17376387	9.624878
lsm6dso Accelerometer Non-wakeup	34968966779829	1.4523563	-0.11151731	9.462078
lsm6dso Accelerometer Non-wakeup	34968986280558	1.4092625	-0.1354583	9.423773
lsm6dso Accelerometer Non-wakeup	34969005781287	1.4236271	-0.10672912	9.375891
lsm6dso Accelerometer Non-wakeup	34969025282016	1.418839	-0.24079865	9.505173
lsm6dso Accelerometer Non-wakeup	34969044782745	1.3805333	-0.26952782	9.543477
lsm6dso Gyroscope Non-wakeup	34969045860245	-0.07192932	0.021937527	0.004910046

Figura 12: esempio di dati dei sensori catturati dall'app

L'applicazione, perciò, non funziona in background perché il flusso di salvataggio dei dati viene gestito da due bottoni start e stop, questo permette di registrare solo i dati con una rilevata importanza e strettamente rilevanti allo studio che deve essere fatto.
Per fare in modo che l'applicazione abbia le autorizzazioni per scrivere nella memoria del dispositivo, è stato necessario aggiungere al manifesto dell'app le seguenti righe:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

6.2.1. Campionamento dei dati dai sensori

Per raccogliere i dati in Wear OS sono stati usati principalmente quattro classi: SensorEventListener, SensorManager, SensorEvent, Sensor.
La classe Sensor rappresenta il sensore, infatti può avere tanti valori quanti sensori ha il dispositivo, alcuni tra i quali sono: [6]

```
int TYPE_GRAVITY //A constant describing a gravity sensor type.
int TYPE_GYROSCOPE //A constant describing a gyroscope sensor type.
int TYPE_ACCELEROMETER //A constant describing an accelerometer
sensor type.
```

La classe SensorEvent ha tutte le informazioni che riguardano ogni rilevazione del sensore come:

- Tipo (nome sensore)
- time-stamp[3] in nanosecondi
- accuracy (precisione)
- Dati/valori che il sensore rileva

[3] il timestamp è un valore numerico intero che esprime il numero di secondi trascorsi da una data arbitraria, cioè la mezzanotte (UTC) del 1 gennaio 1970, momento che prende il nome di epoch.

Molto spesso i valori hanno delle coordinate come x, y e z. Queste coordinate sono relative allo schermo del dispositivo nella sua orientazione di default, gli assi infatti non vengono cambiati quando l'orientamento dello schermo cambia.
In sostanza l'asse delle X è orizzontale e punta a destra, l'asse delle Y è verticale e punta in alto, mentre l'asse delle Z punta direttamente fuori dallo schermo del dispositivo.
SensorEventListener è la classe responsabile a gestire l'arrivo di notifiche dal SensorManager non appena si rilevano novi dati relativi ai sensori.
SensorEventListener ha i seguenti metodi pubblici:

```
abstract void onAccuracyChanged(Sensor sensor, int accuracy)
abstract void onSensorChanged(SensorEvent event)
```

onAccuracyChanged viene chiamato quando la precisione del sensore è stata cambiata, siccome il metodo non è così rilevante allo studio non verrà pressoché utilizzato.
onSensorChanged invece è a dir poco fondamentale perché viene chiamato ogni volta che c'è un nuovo SensorEvent, il quale notifica anche un cambio di valori del sensore che stiamo prendendo in considerazione. Tuttavia "on changed" potrebbe essere d'inganno perché questo metodo verrà chiamato anche se si ha solo una nuova lettura con lo stesso valore dallo stesso sensore, solo che avverrà con un nuovo timestamp.
Infine, SensorManager è la classe che permette di avere accesso ai sensori del dispositivo.
Una volta raccolti i dati, essi venivano scritti e salvati su un file csv all'interno del dispositivo. I dati venivano scritti seguendo lo standard csv (comma separated values) facilitando poi l'analisi e lo studio degli stessi.

6.2.2. Problema dell'Ambient Mode

Wear OS gestisce automaticamente il passaggio alla modalità di risparmio energetico per una app quando un utente non utilizza più l'orologio. Questa è chiamata Ambient-Mode. Se l'utente interagisce ancora con l'orologio Wear OS ripristina l'app esattamente dove si era lasciata.
Wear OS interpreta l'abbassamento del braccio come il momento in cui non si utilizza l'orologio.
È comunque possibile controllare cosa è mostrato nel display anche quando è in ambient mode, per esempio potrebbe essere necessario mostrare le informazioni del battito cardiaco durante una corsa. Questo tipo di app che funzionano anche in ambient mode sono chiamate always-on apps.
Sviluppare una app always-one impatta particolarmente la batteria del dispositivo.

Nel nostro caso usa volta cominciata la sessione di registrazione ed abbassato il braccio l'app smetteva di registrare i dati dei sensori in quanto l'orologio entrava in ambient mode.
Questa cosa si può ben vedere in figura 13, dove ogni puntino rappresenta un cambio del valore di ciascun sensore sulla coordinata x, e per la porzione di tempo centrale dove viene effettuata l'apertura quella porta non avviene nessuna registrazione.

Figura 13: registrazione dati sensori con il dispositivo che va in ambient mode

Per aggirare questo problema è stato aggiunto il seguente attributo xml nel layout dell'app da noi sviluppata:

```
android:keepScreenOn="true"
```

Quest'attributo infatti tiene lo schermo sempre attivo se si è all'interno dell'app così come la CPU, permettendo quindi il campionamento di tutti i dati.

6.2.3. Il Dataset

Per svolgere i vari esperimenti è stato necessario che diversi soggetti effettuino lo stesso movimento.
Nella tesi è stato preso in considerazione un dataset composto da 5 persone di età e sesso diverse.
Ognuna della quali ha aperto la stessa porta più volte registrando il cambiamento dei sensori, rendendo poi quindi possibile l'analisi delle varie registrazioni.

6.2.4. Movimenti registrati

Per rendere possibile l'analisi sui dati rilevanti che permettono di valutare se il movimento studiato può essere utilizzato come autenticazione o captcha si è cominciato a raccoglierli con il soggetto in posizione verticale con le braccia parallele al corpo.
Una volta cominciata la registrazione in questa posizione che chiameremo posizione base, il soggetto effettuerà il movimento di aprire una porta e tornerà in posizione base.

Una volta tornato in posizione base si fermerà la registrazione ed il campionamento dei dati che lo smartwatch effettua.
Per ogni soggetto vengono effettuate almeno due registrazioni dell'apertura della porta, in modo tale da riuscire a confrontare le varie aperture.
Assieme alla registrazione delle aperture delle porte vengono registrate anche altre azioni comuni, come camminare, scrivere al computer, scrivere a mano, grattarsi la testa, indicare qualcosa o qualcuno e bere. Queste registrazioni saranno necessarie per capire quanto diverse sono rispetto all'apertura di una porta, il che è fondamentale da capire se si pensa di utilizzare il movimento di apertura di una porta come un invisible captcha.
A questo scopo è infatti fondamentale capire se l'apertura di una porta è un movimento sostanzialmente diverso da un altro movimento che un soggetto può fare.

7. Analisi movimento come possibile autenticazione

Una volta registrati i dati diverse aperture di porte effettuate da diverse persone, è necessario un metodo per confrontarli e vedere se i vari movimenti hanno sostanziali differenze a seconda del soggetto che le effettua.
Al fine di riconoscere pattern che permettano o meno l'autenticazione tramite le letture dell'accelerometro e del giroscopio, e di calcolare la correlazione tra le varie curve è stato usato Matlab (versione R2021a).
Lo scopo di questo capitolo è infatti quello di studiare e capire se il movimento di apertura di una porta con uno smartwatch al polso può essere usato come un'autenticazione.
Verrà infatti studiata la correlazione tra le varie curve, ed a secondo del risultato di questa operazione verranno tratte delle conclusioni.
Una correlazione molto altra tra due aperture dello stesso soggetto dare una risposta positiva alla possibilità dell'utilizzo di questo metodo come autenticazione. Invece se questo non dovesse accadere e se le registrazioni del movimento di soggetti diversi potessero risultare simili, l'uso del movimento studiato non potrebbe essere usato come autenticazione.

7.1. Spiegazione dell'esperimento

Per capire se l'apertura di una porta può essere usata come metodo di autenticazione sono stati registrati i dati che l'accelerometro e il giroscopio hanno rilevato e campionato come spiegato nel capitolo precedente.
Come abbiamo visto precedentemente la procedura di raccolta dati non separa i cambiamenti di valore a seconda del sensore; quindi, questa operazione viene effettuata in un secondo momento quando si comincia ad analizzare i dati usando Matlab.

Risulta quindi necessario separare i dati dell'accelerometro da quelli del giroscopio, dopo questa operazione si ottengono delle tabelle separate con esclusivamente i valori dell'accelerometro e del giroscopio.
Al fine di usare la funzione che calcola la correlazione tra due curve è necessario che la lunghezza sia la stessa, ma purtroppo non tutte le registrazioni hanno la stessa lunghezza e gli stessi movimenti possono impiegarci più o meno tempo.
Al fine di adattare la lunghezza, e fare in modo che sia uguale, è stata tagliata la parte finale delle curve più lunghe (che impiegano quindi più tempo).
È possibile fare ciò, visto che le lunghezze sono sempre più o meno comparabili e la parte tagliata non sarebbe nient'altro che la parte di assestamento dove si stoppa la registrazione dei dati.
Il tagliare parte della registrazione potrebbe compromettere tutta l'analisi successiva; tuttavia, dai risultati analizzati si può notare che questa rimozione di dati non è rilavante per lo studio in sé dell'apertura della porta.
Una volta fatto questo è stato necessario normalizzare i dati per riuscire a renderli comparabili tra gli stessi ed avere valori significativi.
L'obiettivo della normalizzazione è infatti di modificare i dati usando una scala comune senza distorcere le differenze nell'intervallo o perdere informazioni, questo processo aiuta anche la modellizzazione dei dati stessi.
Per fare ciò è stata usata la funzione Matlab normalize(A), che restituisce i dati con 0 come centro ed una deviazione standard di 1.
Una volta che i dati sono stati normalizzati, si può procedere con il calcolo della correlazione tra ciascuna curva che i sensori hanno catturato.
Per fare ciò è stato effettuato un fit polinomiale, in modo di generalizzare in un modello il movimento effettuato ed attenuare i piccoli possibili errori di misurazione che possono essere rilevati.
Per ogni sensore, e per ogni dimensione che esso fornisce viene quindi effettuato un fit polinomiale come su può vedere in figura 14

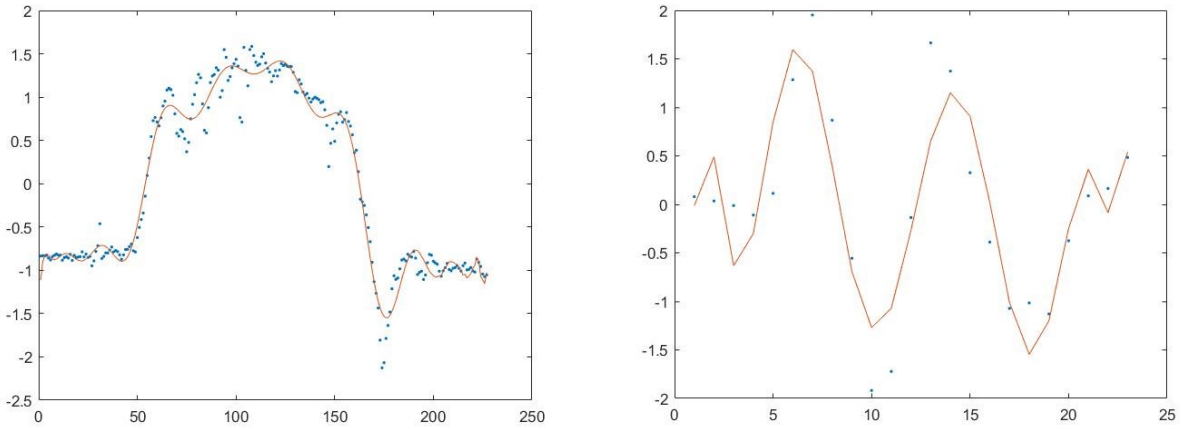

Figura 14: esempio di un fit polinomiale applicato sull'asse x di accelerometro e giroscopio di una registrazione, rispettivamente figura di sinistra e di destra

Siccome i sensori raccolgono diverse quantità di dati non è stato possibile usare un polinomio con lo stesso grado.

Infatti, usare un polinomio di grado ≥ 10 nei dati raccolti dal giroscopio può portare ad un caso di overfitting, in questo caso infatti il fit in sè perderebbe di significato.

Il grado del polinomio viene quindi deciso in base alla dimensione dei dati raccolti, dopo vari tentativi e studio dei risultati è stato scelto rapporto tra dati e grado del polinomio che rappresenta al meglio i dati, questo rapporto è 10:1.

Tuttavia, se i dati sono pochi come nel caso del giroscopio, un grado troppo basso non rappresenta i dati in modo veritiero, perciò, è stata impostata una soglia a 10.

Al fine di confrontare le curve di ciascun movimento, è stato necessario sportarle, allineando l'atto di aprire una porta, eliminando quindi i fattori temporali che caratterizzano ciascun movimento, fattori temporali come aspettare un secondo di più prima di aprire la porta.

Per spostare la curva ed individuare di quanto spostamento ha bisogno è stata utilizzata la funzione xcorr.

Xcorr misura la similarità tra due vettori, uno dei quali viene ritardato di vari valori, è possibile quindi individuare il valore per il quale la correlazione/similarità è maggiore, ed il ritardo applicato al seguente confronto.

In figura 15 si può vedere visualmente come questo ritardo viene applicato a due aperture di porte che risultano molto simili. Prima di applicare il ritardo le curve risultano disallineate, a differenza di come si mostrano con un ritardo applicato.

Per spostare le curve è stato usato la funzione circshift.

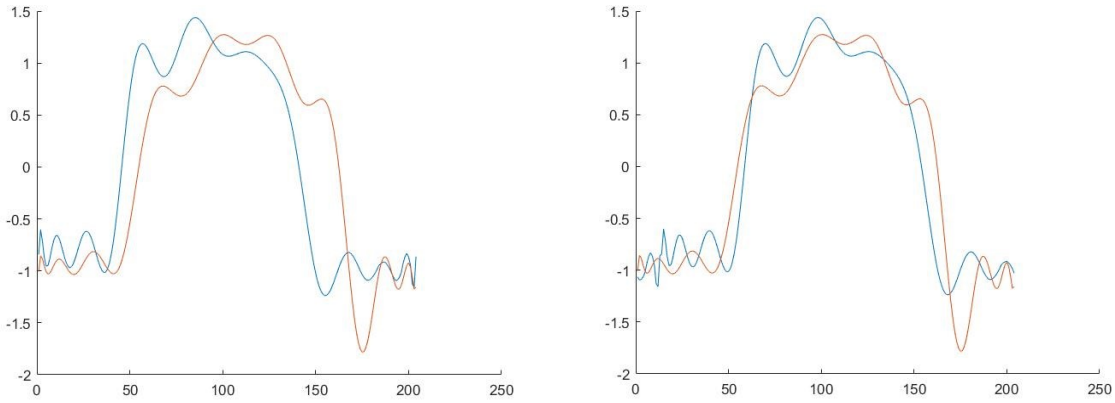

Figura 15: a sinistra: due movimenti di apertura di porta senza applicare alcun ritardo e spostamento, a destra le stesse curve con il ritardo applicato

Una volta spostate le curve, si può applicare la funzione corr che ci restituisce il coefficiente di correlazione tra le dure curve. Questo coefficiente sarà il modo che utilizzeremo per capire la similarità tra le varie curve, permettendoci di fare delle valutazioni sull'autenticazione.
Una volta calcolata la correlazione per ogni dimensione viene calcolata una media dei tre risultati, ottenendo quindi un risultato quanto più veritiero possibile che si riferisce a tutte e tre le dimensioni offerte dai sensori.
La funzione corr restituisce un coefficiente tra -1 ed 1, con 1 che rappresenta una correlazione al 100% quindi la stessa identica curva.

7.2. Risultati

Una volta spiegato come vengono elaborate e confrontate ogni apertura di porta si può cominciare a valutare i risultati di tali confronti.
Per interpretare i risultati e distinguere le aperture di tutti i soggetti, ogni apertura verrà rappresentata da un prefisso di lettere che identifica ogni soggetto ed un suffisso corrisposto da numeri che rappresenta il singolo tentativo di ogni soggetto.
Come possibile criterio per la determinazione per la procedura di autenticazione è ragionevole pensare ad un algoritmo di tipo k-NN[4], associando quindi il pattern più simile come autenticazione, nel nostro caso sarà una k-NN con $k = 1$.[10]
Tuttavia, molto spesso il pattern più simile non coincide con lo stesso soggetto, e porterebbe ad un'autenticazione sbagliata, dando infatti accesso ad un soggetto che non è lo stesso.

[4] Il k-nearest neighbors(k-NN), è un algoritmo utilizzato nel riconoscimento di pattern per la classificazione di oggetti basandosi sulle caratteristiche degli oggetti vicini a quello considerato. In entrambi i casi, l'input è costituito dai k esempi di addestramento più vicini nello spazio delle funzionalità. Un oggetto è classificato da un voto di pluralità dei suoi vicini, con l'oggetto assegnato alla classe più comune tra i suoi k vicini più vicini (k è un numero intero positivo, tipicamente piccolo). Se k = 1, l'oggetto viene semplicemente assegnato alla classe di quel singolo vicino più prossimo.

Questo non succede sempre, infatti come si può vedere nel caso F1 (figura 16), il pattern più vicino è F2 che corrisponderebbe ad un'autenticazione corretta.

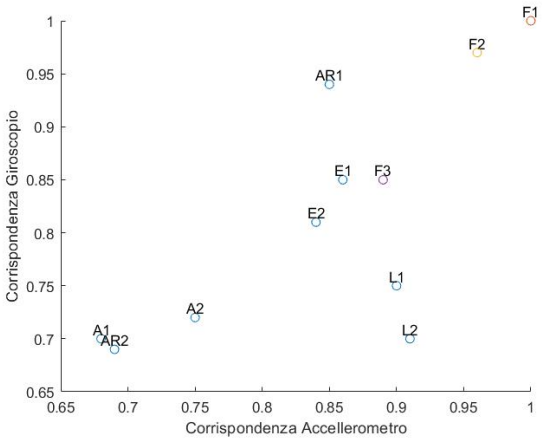

Figura 16: corrispondenza di tutti i movimenti registrati rispetto ad F1

Come metodo di autenticazione, non è abbastanza affidabile, perché analizzando i dati delle varie aperture solo nel caso del soggetto F, l'apertura con la corrispondenza assoluta (tenendo in considerazione sia giroscopio che accelerometro) più vicina è la seconda apertura del soggetto F, negli altri casi infatti questo non succede.

Confrontando infatti tutte le aperture solo nel caso di F ottengo la giusta corrispondenza, avendo quindi una precision del 20% nei casi analizzati.

Le seguenti figure mostrano i casi per i soggetti A, E, L, AR. In tutti questi casi l'apertura più vicina alla prima apertura di ogni soggetto non corrisponde alla seconda dello stesso soggetto.

Anche nella figura 17 il caso di E1, L1 e leggermente più vicina di E2, le distanze tra i due punti sono rispettivamente: 0.15 e 0.152.

Il metodo k-NN quindi darebbe l'autenticazione al soggetto sbagliato.

In altri casi come AR e L le aperture dello stesso soggetto sono sostanzialmente diverse da come si può vedere nella figura 17,

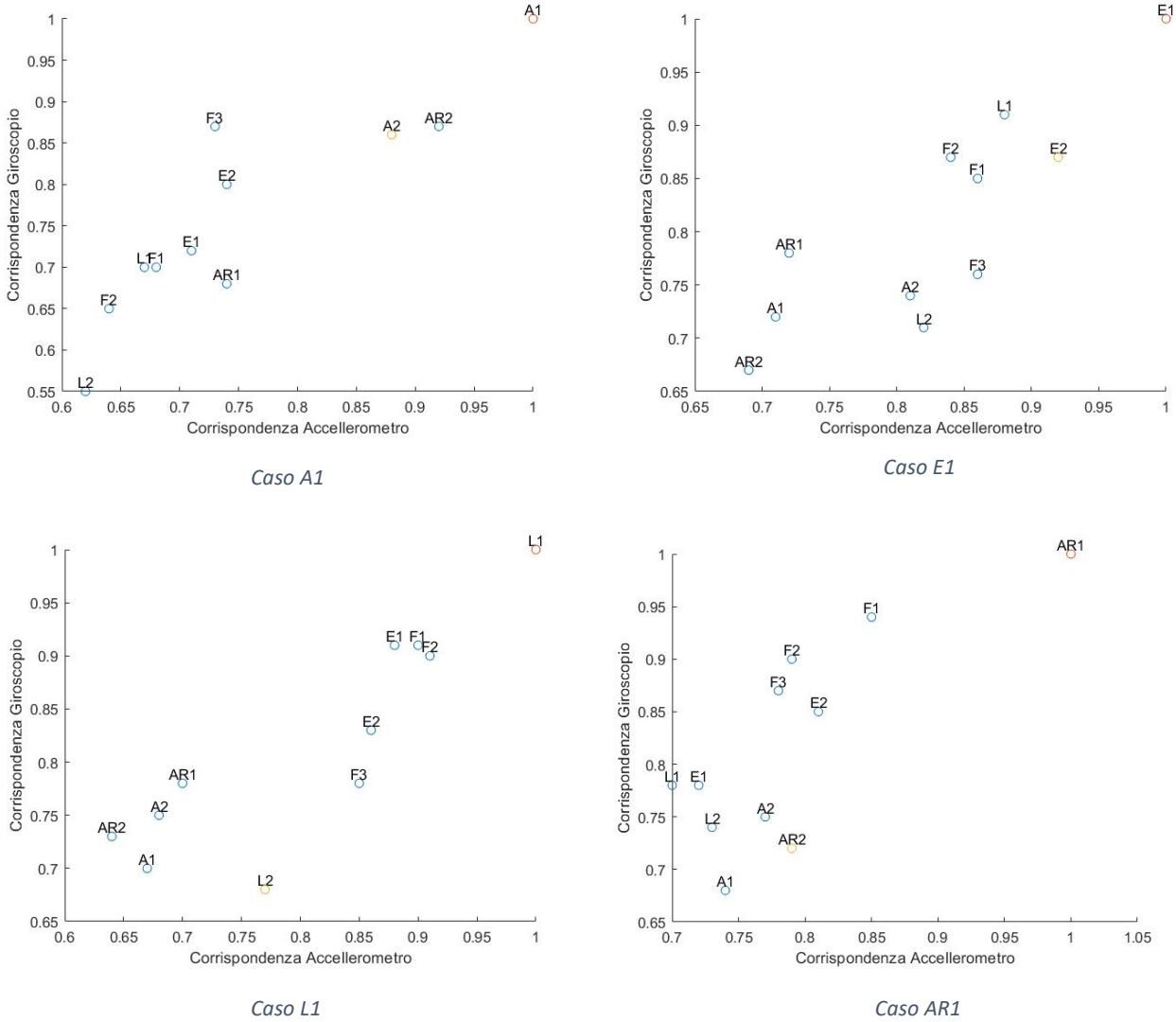

Figura 17: corrispondenza di tutti i movimenti registrati rispetto ad A1, E1, L1, AR1

7.3. Conclusioni e studio sulla fattibilità

Visti i risultati sopra esposti è ragionevole pensare che un'autenticazione solamente basata sul movimento di apertura di una porta non sia possibile, o almeno non sia ancora possibile. Utilizzando infatti un dispositivo destinato al pubblico come quello utilizzato nell'esperimento (Fossil Gen 5), il rilevamento dei dati dei suoi sensori durante il movimento registrato non riesce a permettere un'autenticazione di un soggetto.

Questo potrebbe accadere per due motivi, è infatti possibile che i sensori non siano abbastanza precisi oppure che il movimento in sé non riesce a distinguere un soggetto. Si tratta infatti di una possibilità plausibile il fatto che il movimento di apertura di una porta non riesca a identificare un soggetto in modo talmente preciso da essere un metodo per autenticarlo.

8. Analisi del movimento come possibile captcha

Visto l'impossibilità al momento di utilizzare uno smartwatch come strumento per autenticare un soggetto durante l'apertura di una porta, risulta comunque interessante lo studio e l'analisi della possibilità di usarlo come captcha.
Un invisible captcha che consiste nel verificare che un soggetto stia effettivamente aprendo o meno una porta potrebbe essere estremamente utile.
Come abbiamo infatti visto gli smartwatch moderni dispongono una grande connettività (NFC, Bluetooth, Wi-fi) e spesso identificano un soggetto solo data la loro presenza al polso, preceduta da un'autenticazione; e se uno smartwatch identifica una persona è presumibile che possa essere usato come autenticazione in molteplici azioni come può essere quella di entrare ed accedere ad una stanza, è infatti già usato per operazioni come pagare in un POS[5].
Un captcha che verifichi l'azione di apertura della porta stia avvenendo potrebbe ridurre considerevolmente il rischio di attacchi hacker che provano ad aprire le porte da remoto, distinguendo infatti l'azione umana dall'azione effettuata da un bot.

8.1.Spiegazione dell'esperimento

Similmente a quanto fatto per valutare il movimento di apertura di una porta come possibile autenticazione, è stato necessario effettuare delle operazioni sui dati prima di riuscire ad interpretare e valutare i risultati.
Per fare ciò, i dati sono stati normalizzati, è stata effettuato un fit polinomiale, e ogni curva è poi stata spostata in modo da avere tutte le curve sincronizzate, con il movimento di apertura della porta che avviene nello stesso momento. Queste operazioni sono state eseguite ugualmente a quanto spiegato nel paragrafo precedente, quando si valutava il movimento come possibile autenticazione.
Tuttavia, per riuscire ad impostare un captcha si necessita di dati che rappresentino al meglio l'apertura di una porta in modo generico. È infatti necessaria una curva che rappresenti l'apertura di una porta, non in modo specifico e diverso da persona a persona, bensì in modo generico rappresentando solamente il gesto.
Lo scopo è infatti quello di distinguere il movimento di apertura di una porta da un altro movimento che può essere effettuato.
Per individuare questa curva si è ricorso alla media delle curve del movimento di apertura di una porta per ogni dimensione. Avendo infatti registrato diversi movimenti, è bastato fare una media di tutti i movimenti per avere una curva che rappresenti al meglio il movimento stesso. Questa operazione è stata effettuata per tutte le dimensioni che ogni sensore forniva.

[5] Un POS è un dispositivo elettronico che consente di effettuare pagamenti mediante moneta elettronica, ovvero tramite carte di credito, di debito o prepagate.

Dalla figura 18 si può vedere graficamente il calcolo della media ed il risultato per la coordinata x dell'accelerometro, la linea più grossa in verde, infatti, rappresenta la media delle curve.

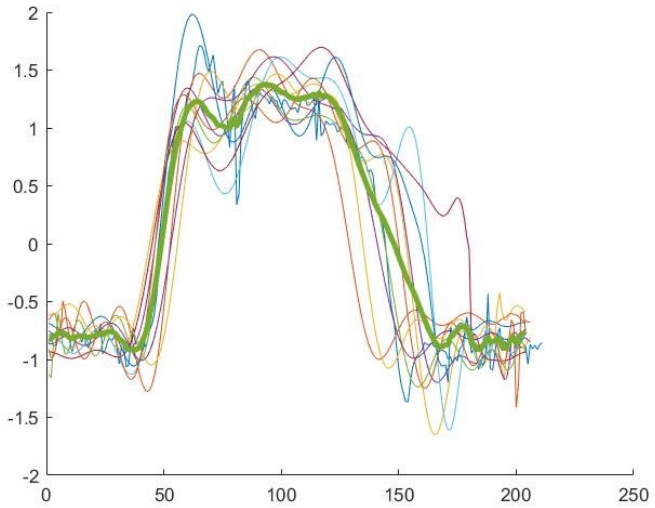

Figura 18: media dei dati catturati dalle aperture della coordinata x dell'accelerometro

Una volta fatto ciò, avendo ottenuto quindi una curva che rappresenta al meglio l'apertura di una porta in ogni sua dimensione, tramite la funzione corr si calcola la somiglianza rispetto ai movimenti di apertura di una porta specifici di ogni soggetto ed altri movimenti che potrebbero essere effettuati con uno smartwatch in mano.

8.2. Risultati

I risultati hanno una tassonomia come quella introdotta nel paragrafo 7.2.
Dall'analisi dei risultati che la funzione corr restituisce si può notare che il movimento di apertura di una porta è facilmente individuabile, tutte le aperture registrate hanno una correlazione ≥ 0.79 per quanto riguarda l'accelerometro e ≥ 71 per quanto riguarda il giroscopio. A differenza di altre azioni comuni che hanno corrispondenze decisamente inferiori.
L'istogramma in figura 19 mostra la correlazione tra le varie aperture delle porte rispetto a quella generica. La media delle correlazioni delle aperture rispetto alla generica è del 0.86 e 0.85 rispettivamente per accelerometro e giroscopio. Ricordando che la correlazione ha un intervallo di valori compreso tra -1 ed 1, una correlazione media di 0.86 è molto alta.

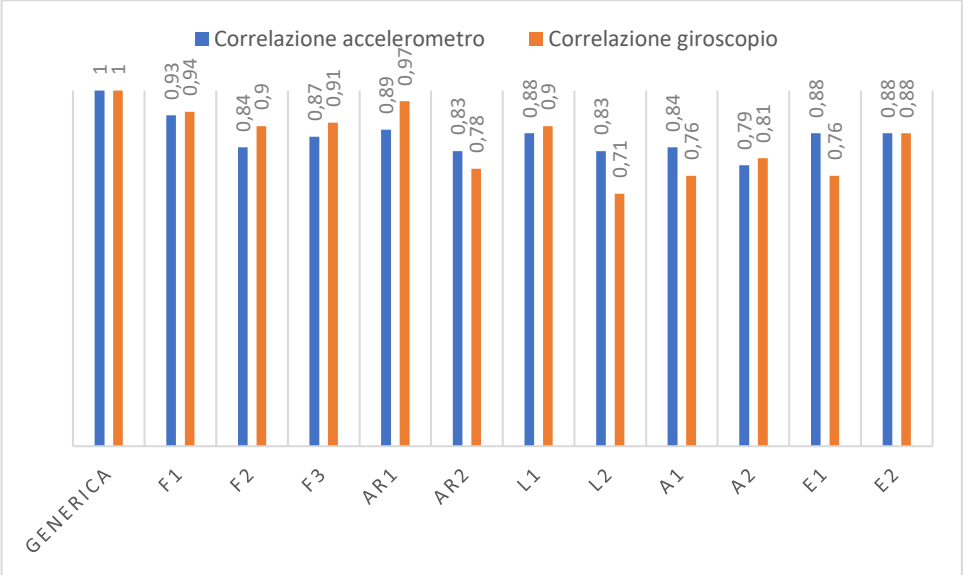

Figura 19: corrispondenza tra l'apertura generica e le altre aperture registrate

Gli altri movimenti, come bere, camminare, grattarsi la testa ed indicare oggetti sono molto distanti con una correlazione rispetto all'apertura generica considerevolmente inferiore, come si può ben vedere in figura 20.

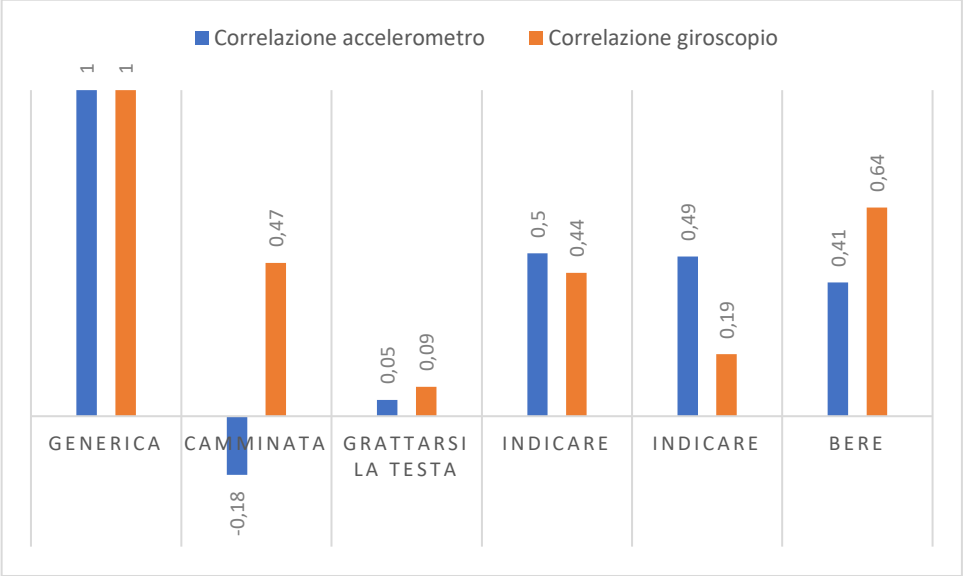

Figura 20: corrispondenza tra l'apertura generica e altri movimenti registrati

I movimenti diversi dall'aprire una porta hanno delle medie di correlazioni pari a 0,25 e 0,37 per le correlazioni rispetto all'accelerometro ed al giroscopio.

8.3. Conclusioni e studio sulla fattibilità

I risultati mostrano chiaramente che è possibile riconoscere il movimento di apertura di una porta. Tutti i movimenti registrati che corrispondono all'apertura di una porta infatti sono simili tra loro, soprattutto in confronto ad altri movimenti.

Una volta calcolata l'apertura generica e la correlazione tra essa e gli altri movimenti registrati (sia di aperture di porte che non) è facile vedere che le correlazioni tra i movimenti

di apertura della porta sono molto simili tra loro. Questo risultato dimostra che l'analisi di questo movimento potrebbe essere usato come possibile captcha, andando a distinguere un movimento fatto da un uomo che sta aprendo la porta ed uno che non lo sta facendo con valori diversi rilevati dallo smartwatch. Nonostante sia un dispositivo destinato al pubblico la distinzione tra i diversi tipi di movimenti è molto chiara, il che potrebbe permettere l'utilizzo di un catpcha del genere anche in grande scala.
Grazie, infatti, anche alla sempre più grande diffusione di smartlock e soluzioni interconnesse che comunicano con smartwatch e smartphone, è già possibile ora aprire una porta con uno smartwatch.
Uno di questi esempi lo porta l'azienda ButterflyMX, che attraverso il loro citofono consente di aprire una porta senza utilizzare chiavi o telecomandi, basta infatti possedere uno smartphone e consentire l'accesso tramite la loro app. App che è presente anche su dispositivi wearable come smartwatch. [13]
L'introduzione di un captcha durante il movimento descritto può quindi aumentare considerevolmente la sicurezza del sistema ed andare a ridurre il rischio di attacchi hacker che cercassero di operare da remoto.

9. Conclusioni

Questa tesi presenta l'analisi di uno nuovo schema di autenticazione che si basa sul movimento che fanno gli utenti quando aprono una porta, attraverso la sensoristica presente su uno smartwatch. Purtroppo, però, come dimostrato nella tesi, questo movimento non può rappresentare un nuovo metodo di autenticazione visto che i risultati che sono stati ottenuti non sono in grado di identificare un utente.
I motivi per i quali un'autenticazione basata su questo movimento non risulta possibile dall'analisi fatta potrebbero essere molteplici, ma tra i più probabili c'è sicuramente la possibile poca accuratezza del dispositivo, è possibile infatti che negli anni e con dispositivi più precisi un'autenticazione del genere sia possibile; un altro motivo probabile potrebbe essere il fatto che il movimento stesso di apertura di una porta non riesca a distinguere un utente, perché non è un movimento significativamente diverso a seconda di ogni utente, per verificare questa ipotesi però sarebbe necessario acquisire dati con sensori di precisone maggiore rispetto a quella disponibile su dispositivi consumer del giorno d'oggi.
Tuttavia, nella tesi è stata anche esplorata l'opzione di usare lo stesso movimento come possibile captcha. Un invisible captcha è infatti un risultato possibile monitorando i dati sul comportamento dell'utente mentre effettua il movimento di apertura di una porta. Tali dati mostrano che il movimento studiato è particolarmente unico indipendentemente dal soggetto

che lo effettua, rendendo possibile la distinzione tra un soggetto che sta facendo il movimento o meno.

In questo momento l'utilità e gli utilizzi di tale captcha potrebbero essere molteplici però necessita ancora di tanta ricerca. Due sono i punti principali che devono essere migliorati ed analizzati.

Il primo dei quali è sicuramente l'individuare un movimento di wake-up che permetta di capire che sta per avvenire il movimento di apertura di una porta.

Il secondo comprende tutta la parte di comunicazione e gestione dei dati, compresa la loro analisi. È importante capire come la comunicazione tra lo smartwatch è il sistema che apre le porte avviene, e dove verrà confrontato il movimento di apertura di una porta per approvarlo o meno.

10.Bibliografia

[1] Pooja Kohli. "Why Continuous Authentication has Replaced MFA" in Youtube
[2] Silvano Marioni, 2019, Password e metodi di autenticazione: caratteristiche tecniche e nuove soluzioni, cybersecurity360.it, visitato il 13 dicembre 2021, <https://www.cybersecurity360.it/soluzioni-aziendali/password-e-metodi-di-autenticazione-caratteristiche-tecniche-e-nuove-soluzioni/>
[3] A. Siripitakchai, S. Phimoltares and A. Mahaweerawat, "EYE-CAPTCHA: An enhanced CAPTCHA using eye movement," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), 2017, pp. 2120-2126, doi: 10.1109/CompComm.2017.8322911.
[4] J. Bonneau, C. Herley, P. C. v. Oorschot and F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," 2012 IEEE Symposium on Security and Privacy, 2012, pp. 553-567, doi: 10.1109/SP.2012.44.
[5] G. M. Weiss, K. Yoneda and T. Hayajneh, "Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living," in IEEE Access, vol. 7, pp. 133190-133202, 2019, doi: 10.1109/ACCESS.2019.2940729.
[6] Android Development Documentation
[7] Thomas Brewster, 2018, Google's Doors Hacked Wide Open By Own Employee, Forbes, visitato il 13 dicembre 202, <https://www.forbes.com/sites/thomasbrewster/2018/09/03/googles-doors-hacked-wide-open-by-own-employee/>
[8] Badillo, Solveig & Banfai, Balazs & Birzele, Fabian & Davydov, Iakov & Hutchinson, Lucy & Kam-Thong, Tony & Siebourg-Polster, Juliane & Steiert, Bernhard & Zhang, Jitao David. (2020). An Introduction to Machine Learning. Clinical Pharmacology & Therapeutics. 107. 10.1002/cpt.1796.
[9] Fred B. Schneider, Something You Know, Have, or Are, su cs.cornell.edu
[10] K-nearest neighbors. (13 ottobre 2021). Wikipedia, L'enciclopedia libera. Tratto il 13 dicembre 2021, 15:01 da //it.wikipedia.org/w/index.php?title=K-nearest_neighbors&oldid=123437007.
[11] Zhang M, Guo C. A Kind of Mobile Phone Users'Authentication System Based on the Characteristics of Keystrokes [J]. Computer Programming Skills & Maintenance, 2009

[12] M. O. Derawi, C. Nickel, P. Bours and C. Busch, "Unobtrusive User-Authentication on Mobile Phones Using Biometric Gait Recognition," 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2010, pp. 306-311, doi: 10.1109/IIHMSP.2010.83.

[13] Meredith Murray, 2020, Open The Door With An Apple Watch, butterflymx.com, visitato il 13 dicembre 2021, < https://butterflymx.com/blog/open-the-door-with-an-apple-watch/>

[14] Meriem Guerar, Luca Verderame, Mauro Migliardi, Francesco Palmieri, and Alessio Merlo. 2021. Gotta CAPTCHA 'Em All: A Survey of 20 Years of the Human-or-computer Dilemma. ACM Comput. Surv. 54, 9, Article 192 (December 2022), 33 pages. DOI:https://doi.org/10.1145/3477142

[15] Kumar Chellapilla and Patrice Y. Simard. 2004. Using machine learning to break visual human interaction proofs
(HIPs). In 17th International Conference on Neural Information Processing Systems (NIPS'04). The MIT Press, Cambridge,
MA, 265–272. DOI:https://doi.org/10.5555/2976040.2976074

[16] H. Gao, J. Yan, Fang Cao, Zhengya Zhang, Lei Lei, Mengyun Tang, P. Zhang, X. Zhou, Xuqin Wang, and J. Li. 2016.
A simple generic attack on text captchas. In Network and Distributed System Security Symposium.

[17] J. Yang, Y. Li, and M. Xie. "MotionAuth: Motion-based authentication for wirst worn smart devices". In: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). 2015.