Corso di Laurea in Informatica
Facoltà di Scienze Matematiche, Fisiche e Naturali

PRIVSeq: una Libreria Python per la Valutazione del Rischio di Privacy su Dati Sequenziali.

Relatori:
Prof. Anna Monreale
Prof. Roberto Pellungrini

Candidato:
Francesco Gemignani

24 Luglio 2020

A mia moglie Raffaella e mio figlio Tiziano.
Senza la vostra spinta, non sarei mai riuscito a tuffarmi.

Ringraziamenti

Sarebbe impensabile non essere riconoscente nei confronti di tutte quelle persone che, durante questo percorso, mi hanno sopportato e supportato in questa scelta, accompagnandomi per mano fino a tagliare questo mio personale traguardo.

Innanzi tutto, vorrei ringraziare i miei relatori: Anna Monreale e Roberto Pellungrini per la chiarezza e l'ampia disponibilità con cui hanno saputo appoggiarmi durante lo svolgimento di questo progetto, sia dal punto di vista professionale che da quello emotivo. Un ringraziamento va anche a tutti i docenti del dipartimento di informatica che hanno investito parte del proprio tempo per darmi consigli, allineandomi nello scegliere la direzione più adatta alla mia attuale e "difficile" situazione. In particolare, ringrazio il prof. Francesco Romani che ha saputo sostenermi e incoraggiarmi, dandomi molteplici consigli, sia umani che professionali.

Un grande ringraziamento anche ai miei amici, ai miei ex-colleghi di lavoro e a tutti coloro che hanno incrociato la loro vita con la mia lasciandomi qualcosa di buono. Grazie per essere stati miei complici, ognuno a suo modo, in questo percorso così intricato quanto entusiasmante, nel bene e nel male.

Un immenso grazie alla mia famiglia. A mia madre, mio padre e al mio caro fratello. Senza di voi tutto questo non potrebbe esistere, questo è soltanto uno degli infiniti momenti in cui non posso altro che esservi grato.

Un grande riconoscimento non posso non spenderlo nei confronti della mia seconda famiglia: Elena e Benedetto. In questo angosciante periodo mi avete dato la possibilità fisica e morale di continuare questo percorso. Avete creduto in me e ad ogni piccolo traguardo avete esternato più gioia di quanta potessi averne io. Mi avete trattato come un terzo figlio riempendo il mio cuore di grande affetto. Grazie.

Un immenso grazie agli amori della mia vita che portano lo stesso nome: mio figlio e mio fratello Tiziano. Siete la prima cosa che penso al mattino e l'ultima prima di addormentarmi. Il vostro calore ed il vostro affetto mi riempie di forza e motivazione. Una grande fetta di questo traguardo è merito vostro.

Infine, il ringraziamento più importante va alla persona che più mi è stata vicina: mia moglie Raffaella. La vera forza motrice, in grado di spingermi e darmi fiducia, non solo in questi mesi, ma ormai da molti anni. Se alla fine sono riuscito a prendere una decisione è soltanto merito tuo, la mia sola sicurezza e determinazione non sarebbero state sufficienti.

Grazie a tutti. Mi avete aiutato a voltare pagina, facendomi capire che nella vita non bisogna mai accontentarsi e che, soprattutto, non dobbiamo mai arrenderci. Finché abbiamo ossigeno nei polmoni possiamo trovare la forza e la motivazione di fare qualsiasi cosa. Non esiste niente di impossibile, basta iniziare per accorgersi di essere già oltre metà strada. Dipende soltanto da noi e da quello in cui crediamo realmente. Siamo noi il mezzo con cui raggiungere i nostri obiettivi, non esistono ulteriori ostacoli se non il timore di non farcela.

Grazie

Sommario

La privacy nell'analisi dei dati rappresenta uno dei problemi di maggior rilievo che ogni azienda deve saper affrontare e gestire nel migliore dei modi. La tutela e la gestione dei dati riservati implica pertanto un'attenta analisi. Il nostro obiettivo in questa relazione è quello di misurare il grado con il quale un certo individuo si espone, fornendo una libreria in grado di quantificare in modo empirico il rischio di privacy in funzione delle potenziali conoscenze che un malintenzionato potrebbe utilizzare per identificare un certo individuo. Ogni tipologia di attacco, quindi, rappresenta una conoscenza e allo stesso tempo una probabile minaccia che un malintenzionato potrebbe utilizzare per i propri scopi e che, invece, un'azienda potrebbe sfruttare per tutelare i propri clienti. PRIVSeq si basa su questo approccio, mettendo a disposizione dell'utilizzatore un ventaglio di attacchi con il quale calcolare il rischio di privacy a partire da dati sequenziali generici di tipo arbitrario. La relazione è composta da quattro capitoli, nel primo trattiamo il concetto di privacy e protezione dei dati, sottolineando quelli che sono gli attori di un processo di analisi fino a introdurre i principali frameworks di valutazione del rischio. Il capitolo successivo definisce i formalismi matematici con i quali modellare i diversi tipi di dato utilizzati nella struttura generale di moderazione del rischio. Nel quarto capitolo descriviamo nel dettaglio la libreria, evidenziando ed analizzando le principali procedure implementate.Nell'ultimo capitolo testiamo la libreria su un dataset di retail reale, valutiamo il rischio su ampia scala e analizziamo l'andamento dei vettori di rischio plottandone la distribuzione.

Indice

1 Introduzione **9**

2 Letteratura **11**
2.1 Normative 11
2.2 Elementi di un Processo di Analisi 12
2.3 Dati Sequenziali 12
2.4 Framework per il Calcolo del Rischio di Privacy 14
2.4.1 PRUDEnce 14
2.4.2 Altri Framework in Letteratura 15

3 Definizioni **17**
3.1 Modellazione dei Dati 17
3.1.1 Strutture Ausiliarie 18
3.2 Struttura di Moderazione di Calcolo del Rischio 21
3.2.1 PRUDEnce ed il Rischio di Privacy 21
3.2.2 Valutazione Generale del Rischio 22
3.2.3 Elements Based Knowledge: Valutazione del Rischio 23
3.2.4 Sequence Based Knowledge: Valutazione del Rischio 24
3.2.5 Full Sequence Knowledge: Valutazione del Rischio 25
3.3 Modellazione degli Attacchi 26
3.3.1 Elements Attack 26
3.3.2 Elements Sequence Attack 28
3.3.3 Elements Time Attack 30
3.3.4 Frequency Attack 33
3.3.5 Probability Attack 35
3.3.6 Proportion Attack 37

4 Libreria **40**
4.1 Analisi e Struttura del Problema 40

4.2 PrivacyDataFrame: Analisi e Struttura di un Dataframe Generico 41
4.3 Generalità di un Attacco 43
4.3.1 Il file Attacks.py 43
4.3.2 Creazione ed Esecuzione di un Attacco 43
4.3.3 Metodi di Calcolo del Rischio 44
4.4 Struttura di un Attacco 44
4.4.1 Costruzione di un Generico Attacco 46
4.4.2 Generazione del Background Knowledge 46
4.4.3 Valutazione del Rischio di Privacy Individuale 48
4.4.4 Valutazione del Rischio di Privacy Totale 51
4.4.5 Tipi di Attacchi: Analisi e Struttura 52

5 Testing sperimentale **58**
5.1 Tipo di Dato 58
5.2 Analisi dei Dati e Scelte Progettuali 60
5.3 Risultati e Distribuzione dei Rischi 63

6 Conclusioni **68**

Capitolo 1

Introduzione

Ogni giorno milioni di utenti accedono alla rete e forniscono con leggerezza i propri dati per ricevere servizi, senza riflettere sul fatto che stanno implicitamente acconsentendo ad essere monitorati in ogni attività. Ciò può avvenire sia in modo volontario che in modo involontario. Si pensi a quando andiamo ad acquistare una nuova auto. Tra i mille comport e servizi potrebbe esserci installato un sensore di rilevazione della posizione che invia periodicamente le nostre coordinate ad un provider, il quale ci offre una splendida applicazione che tiene traccia delle traiettorie percorse, la media giornaliera, i chilometri ed il periodo di tempo trascorso in auto, il tutto corredato da una meravigliosa interfaccia grafica. Nessuno ha detto che la volessimo, ma l'acconsentimento alla trasmissione dei dati personali l'abbiamo accettato, senza saperlo, nel contratto d'acquisto. Per quanto possano essere teoricamente al sicuro i nostri dati, per quanto possa essere affidabile il provider del servizio, niente e nessuno esclude che un malintenzionato possa accedere al database delle informazioni e "in qualche modo" identificarci, venendo a conoscenza non solo di tutti i nostri viaggi, ma anche di altre informazioni sensibili associate a noi (i.e., carte di credito, numero di telefono, residenza). Il malintenzionato potrebbe essere una persona che conosciamo e che un giorno potrebbe decidere di pedinarci, arricchendo il bagaglio di conoscenza con il quale aumentare la probabilità di identificarci all'interno del database.

Poter quantificare il rischio di re-identificazione dei propri clienti, fornisce ad un'azienda un arma molto potente. Equivale a garantire l'anonimato e tutelare la privacy di ogni individuo iscritto. I concetti di privacy e protezione dei dati sono spesso confusi, come se non esistesse una semantica che ne evidenziasse le differenze. In realtà, sono tanto diversi quanto interconnessi. La privacy, fa riferimento al diritto alla riservatezza delle informazioni personali e della propria vita privata. Si tratta di un principio che usiamo come strumento per tutelare la sfera intima del singolo individuo volto ad impedire che le informazioni siano divulgate in assenza di specifica autorizzazione. Nella letteratura,

usiamo il termine privacy quando vogliamo rappresentare uno spazio personale che gli sconosciuti non possono oltrepassare. La protezione dei dati, invece, è un sistema di trattamento degli stessi che identifica direttamente o indirettamente una persona.
Riallacciandoci al precedente esempio, possiamo identificare con il termine privacy tutti i nostri viaggi, ma anche tutte le informazioni che derivano da essi e le informazioni molto sensibili associate al nostro profilo utente. La protezione o tutela dei dati personali è il servizio erogato dal provider con cui abbiamo (intenzionalmente o meno) sottoscritto l'abbonamento.

Nell'ambito della calcolo del rischio di privacy esistono differenti tipologie di framework in grado di valutare il rischio utilizzando diversi approcci (i.e., rischio empirico o probabilistico), ma a livello pratico le soluzioni non sono molte, soprattutto per quanto riguarda dati di tipo sequenziale. I dati sequenziali hanno caratteristiche molto particolari perché, a differenza dei dati tabulari, non è immediato definire quali siano gli attributi sensibili o quali siano gli attributi che identificano un soggetto. Un chiarissimo esempio di dati sequenziali è appunto rappresentato dai dati di mobilità, ma anche da dati di retail. La valutazione del rischio di privacy per questi tipo di dato non è banale.

Il mio contributo, durante lo svolgimento di questo progetto, è stato quello di sviluppare una libreria che permetta di valutare in modo empirico il rischio di privacy su dati sequenziali di tipo generico e definire un insieme di attacchi sulla base delle conoscenze che un malintenzionato potrebbe utilizzare per re-identificare un individuo. In questa maniera possiamo affrontare questa problematica, utilizzando gli attacchi sviluppati su tutti i tipi di dato che rientrano nella famiglia dei dati sequenziali. Ad esempio, riallacciandoci ai dati di mobilità, un malintenzionato potrebbe pedinarci e conoscere il numero di volte che siamo andati in un certo luogo e successivamente re-identificarci in base alla frequenza. In sintesi, utilizziamo il set di attacchi per simulare la maggior parte delle aggressioni con cui un malintenzionato potrebbe individuarci all'interno del database e quantifichiamo tale valore al caso pessimo: definendolo come massima probabilità di re-identificazione o, rischio di privacy.

La tesi è suddivisa in quattro parti. Nella prima definiamo la letteratura descrivendo le normative e gli attori che ruotano intorno alla protezione e tutela della privacy. Accenniamo i dati sequenziali ed i frameworks maggiormente utilizzati in ambito privacy, evidenziandone le principali differenze. La seconda parte è interamente dedicata alla modellazione e formalizzazione sia dei dati sequenziali di tipo generico che delle strutture che li ospitano. Definiamo i metodi di calcolo del rischio e tutti i tipi di attacchi sviluppati. La terza parte analizza la struttura della libreria, evidenziandone le principali scelte implementative. Infine dedichiamo un capitolo al testing sperimentale prendendo in esame un dataset di retail, con cui quantifichiamo e visualizziamo differenti tipologie di rischio.

Capitolo 2

Letteratura

In questo capitolo trattiamo alcuni concetti della letteratura, spaziando dalle normative che tutelano la protezione dei dati personali, per poi passare a definire letteralmente i ruoli degli attori che partecipano ad un processo di protezione della privacy. Successivamente descriviamo quali sono i tipi di dato maggiormente sensibili, accennando ad alcuni frameworks tra cui PRUDEnce [12] il quale permette di calcolare il rischio di privacy a partire da un dataset di dati di mobilità relativo ad un insieme di individui.

2.1 Normative

L'Unione europea ha redatto, nella direttiva sulla protezione dei dati del 1995 [5], una serie di norme per il trattamento dei dati personali che includono una serie di diritti a tutti gli interessati. Inoltre, recentemente, il Parlamento Europeo ha proposto una riforma, pubblicando il regolamento relativo alla protezione delle persone fisiche con riguardo al trattamento dei dati personali [13], principali concetti della direttiva. I principali concetti sono:

- **Avviso**: ogni cittadino europeo deve essere avvisato sulla raccolta dei suoi dati personali, dando la propria esplicita autorizzazione.
- **Scopi**: i dati devono essere utilizzati solo per lo scopo dichiarato.
- **Sicurezza**: i dati devono essere protetti da qualsiasi tipo di abuso.
- **Divulgazione**: i cittadini interessati devono essere informati su chi raccoglie i loro dati.
- **Accesso**: gli interessati devono avere il libero accesso ai propri dati, modificarli e correggerli da eventuali errori.

- **Responsabilità**: i cittadini europei devono avere la possibilità di ritenere responsabile l'ente che gestisce i loro dati.

2.2 Elementi di un Processo di Analisi

Innanzi tutto, con processo di analisi delle minacce o "threat modeling" ci riferiamo alla procedura con la quale vengono identificate, classificate ed analizzate potenziali minacce, valutandone il rischio e fornendo le necessarie contromisure. Tale processo fornisce un'analisi sistematica di quali controlli o difese debbano essere incluse con l'obiettivo di proteggere le informazioni personali dei clienti.

É importante conoscere chi sono gli attori principali di tale processo. Questo problema è stato discusso per la prima volta in [6], dove vengono identificati tre principali attori, ognuno dei quali con differenti problematiche relative alla privacy.

- *Rispondente*: è la persona che genera i dati il cui principale obiettivo è quello di evitare la divulgazione dei dati sensibili. E' considerato un soggetto passivo all'interno del processo di privacy: nel senso che, una volta generati i dati, non si preoccupa di garantirne la riservatezza.

- *Titolare*: è il soggetto, l'organizzazione o la persona che raccoglie e conserva i dati. Titolare o "proprietario" spesso considerati la medesima persona, spesso viene usato quest'ultimo in letteratura [6]. Secondo il GDPR [13], i titolari sono i diretti responsabili nell'attuazione di tali misure al fine di garantire la tutela della privacy delle persone coinvolte: assicurandosi, quindi, che le informazioni non vengano divulgate.

- *Utente*: è l'individuo che genera dati attraverso l'utilizzo di uno specifico servizio e che ha una partecipazione diretta alla protezione della propria privacy. Il principale obiettivo è quello di assicurare la privacy durante l'accesso e l'utilizzo di uno specifico servizio o sistema.

C'è anche un quarto soggetto: l'*avversario*, il cui obiettivo è attaccare uno dei suddetti attori col fine di divulgare o conoscere parte delle informazioni. In [1] il termine avversario è usato in modo equivalente al termine "attaccante", il cui attacco condotto da quest'ultimo viene spesso definito come attacco alla privacy.

2.3 Dati Sequenziali

Prima di trattare i dati sequenziali è necessario introdurre i dati tabulari, evidenziandone le differenze sia in termini di rappresentazione logica che in un contesto di valutazione del

rischio. Il dato tabulare è un dato relazionale. Ogni record contiene i valori di tutti gli attributi che definiscono un determinato individuo, pertanto ogni record rappresenta un unico utente. In Python lo rappresento con una matrice bidimensionale in cui ogni riga è un individuo o un esempio del mio fenomeno naturale e ed ogni colonna è un attributo o un a caratteristica del mio fenomeno naturale. Ad esempio si consideri il database di uno studio medico: ogni record contiene tutte le informazioni mediche relative ad un individuo (i.e., altezza, peso, pressione, etc).

In un dato di tipo sequenziale, invece, un individuo viene rappresentato da un insieme di record, o meglio da un record composto da attributi aventi una dimensione superiore ad uno. La differenza cruciale tra i due tipi di dato è che in questi ultimi viene introdotto il concetto di dimensione. Per chiarirci meglio di cosa trattiamo, si pensi ad un record di un individuo avente un attributo che descrive la traiettoria. La traiettoria è composta, salvo casi eccezionali, da un insieme di punti superiore (banalmente maggiori di uno). Di conseguenza il record del nostro individuo si espande trasversalmente in un ulteriore insieme di record.

Alla luce di quanto appena detto possiamo immaginare la netta differenza tra una struttura di dati tabulari e sequenziali. I dati sequenziali generici di tipo arbitrario trattati in questo progetto sono rappresentati in Python da matrici bidimensionali, nel quale ogni individuo viene definito da un determinato numero di record: una sequenza di record.

La valutazione del rischio su un dato tabulare è banale. E' sufficiente scegliere un determinato numero di attributi sensibili su cui fare la valutazione del rischio senza definire nessuna dimensione. Indipendentemente se conosco o meno gli attributi con cui l'attaccante intende re-identificare la vittima, il calcolo del rischio lo ottengo calcolando le probabilità rispetto a tutta la popolazione: se conosco i tipi degli attributi calcolo direttamente le probabilità, altrimenti genero le combinazioni e per ognuna di esse calcolo le probabilità, prendendo alla fine quella peggiore.

Il calcolo del rischio in un dato di tipo sequenziale deve tenere conto anche della dimensione del tipo dell'informazione conosciuta. Ad esempio, l'attaccante conosce un insieme di punti appartenenti all'attributo traiettoria: il numero dei punti rappresenta la dimensione del attributo sequenziale. Il proprietario, non conoscendo quali sono i due punti, procede combinatoricamente generando tutti i sottoinsiemi di quella data dimensione.

Il dato sequenziale generico contenente elementi di tipo arbitrario che trattiamo in questo progetto è la sequenza (da non confondere con la sequenza di record che definisce un individuo). Essa può rappresentare una traiettoria composta da elementi che definiscono coppie di valori (latitudine e longitudine) oppure una spesa i cui elementi contengono il nome del prodotto acquistato.

I dati sequenziali possono contenere informazioni molto sensibili. Ad esempio i dati

di mobilità umana contengono informazioni personali sensibili e possono rivelare molti aspetti della vita privata di una persona, rendendo non trascurabile la possibilità di violarne seriamente la privacy da parte di un malintenzionato.
É stato dimostrato che quattro punti spazio temporali sono sufficienti per identificare in modo univoco il 95% delle persone in un dataset di mobilità [10]. Una panoramica dei problemi, metodologie e tecniche dei dati di mobilità urbana e della loro analisi può essere trovata in [19].

Tuttavia, negli ultimi anni, sono state proposte in letteratura diverse tecniche per preservare la privacy (i.e., randomizzazione[2], privacy differenziale[3]) dei dati di mobilità umana [7], dimostrando che è possibile progettare servizi di mobilità nel quale far coesistere sia la qualità dei risultati che la protezione dei dati personali.

In questo contesto, uno dei framework di rilievo per la valutazione del rischio è PRUDEnce [12] il quale permette di preservare la qualità dei dati e calcolare il rischio di privacy empirico a partire da un dataset di dati di mobilità relativo ad un insieme di individui.

2.4 Framework per il Calcolo del Rischio di Privacy

In questa sezione descriviamo alcuni tipi di framework utilizzati per il calcolo del rischio di privacy, valutandone i vantaggi e gli svantaggi in termini di efficienza e complessità.

2.4.1 PRUDEnce

I dati che descrivono le attività umane sono una risorsa fondamentale sia per studiare i comportamenti di un insieme di individui che per sviluppare un ampia gamma di servizi. Sfortunatamente, questo tipo di informazioni è molto sensibile, perché la posizione delle persone può potenzialmente consentire la loro re-identificazione all'interno di un database. Tuttavia, i providers di dati, prima di condividere queste informazioni, devono applicare una sorta di anonimizzazione per ridurre il rischio di privacy, ma allo stesso tempo devono poter gestire e controllare anche la qualità dei dati. Non sempre è facile trovare il miglior compromesso tra questi due fattori. In questa sezione descriviamo PRUDEnce, un ecosistema di condivisione e protezione dei dati. Il framework permette di calcolare sia il rischio di privacy empirico relativo ad un individuo del dataset che di preservarne la qualità dei dati. L'idea è quella di supportare il provider dei dati, fornendogli un insieme di strumenti che gli permettano di manipolare e trasformare le informazioni garantendo il miglior compromesso tra qualità e protezione dei dati. Gli strumenti messi a disposizione da PRUDEnce permettono al provider di garantire la protezione delle informazioni e la qualità dei dati degli individui di un dataset. La protezione della privacy è quantificata

in base alla probabilità che un certo individuo possa essere re-identificato all'interno del dataset, mentre la qualità dei dati è rappresentata dalla quantità di informazioni preservate considerando soltanto individui a basso rischio ed entro una certa soglia. PRUDEnce è stato progettato per assistere le organizzazioni nel condividere le informazioni personali dei propri clienti, prevenendo violazioni della privacy e aiutando loro nella scelta del miglior metodo che ne garantisca l'anonimato. In conclusione, PRUDEnce fornisce un approccio nel quale prima ancora di applicare qualsiasi tipo di trasformazione che garantisca la privacy, permette di valutare il rischio effettivo all'interno dei dati anziché fare affidamento sulle teorie di preservazione della privacy.
Spesso non è necessario effettuare trasformazioni sull'intero insieme grezzo di dati, infatti sono rarissimi i casi in cui abbiamo bisogno direttamente di essi per sviluppare un servizio. In generale, ogni servizio ha bisogno di un insieme di dati già processati. Per esempio, le rilevazioni GPS registrate dalle automobili degli individui, non sono tutte necessarie e quindi non vengono utilizzate direttamente per sviluppare servizi. Le applicazioni moderne richiedono aggregazioni di dati da posizioni spazio temporali differenti, potrebbe essere necessario conoscere la presenza di un individuo in una certa posizione o se la stessa posizione fa parte di una certa traiettoria. Lo stesso ragionamento potrebbe essere esteso in più intervalli di tempo di diversa lunghezza, ad esempio ponendo dei vincoli in funzione della frequenza delle attività degli utenti. Questi differenti passaggi di elaborazione dei dati, che chiamiamo dataviews, possono far emergere proprietà molto interessanti e differenti tra loro. Ogni volta che elaboriamo i dati dobbiamo sempre tenere conto del valore del rischio precedente. Se non lo facessimo, potremmo rischiare di distruggere senza motivo la qualità dei dati, come spesso accade nella privacy differenziale [5,20,16]. In realtà, come è stato sottolineato in [32], la privacy differenziale può seriamente e ingiustificatamente danneggiare le informazioni a causa del calcolo della sensibilità globale, che non prende in considerazione le informazioni da proteggere. In questo complesso contesto, PRUDEnce utilizza un sistema di condivisione e protezione dei dati il quale permette ad un Data Provider(DP) di condividere le informazioni dei propri clienti con un Service Developer (SD) dopo aver valutato i rischi e la qualità di tutte le dataviews e selezionando quella con le sue aspettative di privacy. Il sistema proposto è generale e richiede semplicemente l'implementazione della valutazione del rischio e la mitigazione del rischio.

2.4.2 Altri Framework in Letteratura

Uno dei più importanti progetti per il calcolo del rischio di privacy è Linddun [4], un framework che utilizza la metodologia Stride di Microsoft [18] per modellare le minacce di privacy su sistemi software. Tuttavia, Linddun non utilizza un approccio quantitativo per la valutazione della privacy.

Negli ultimi anni, sono state proposte ulteriori soluzioni per la gestione dei rischi. Un ulteriore esempio è la metodologia di valutazione del rischio OWASP [11], OCTAVE [8] di SEI e DREAD [9] di Microsoft. Sfortunatamente, però, la maggior parte di questi approcci non tiene conto della valutazione rischio di privacy ma include soltanto considerazioni sulla privacy quando si valuta l'impatto delle minacce. Esiste, inoltre, la libreria skmob [16] specializzata nel calcolo empirico del rischio di privacy su dati di mobilità. Tale libreria implementa un insieme di attacchi sulla base del tipo di conoscenza spazio-temporale che possiede l'aggressore, a partire da un dataset di mobilità di più individui. Tale libreria utilizza un approccio quantitativo per il calcolo del rischio però soltanto su dati di mobilità.

Il mio contributo, in questo progetto, è stato quello di sviluppare un modulo nel quale valutare il rischio di privacy a partire da un insieme generico di dati sequenziali, quindi non necessariamente dati di mobilità. Nel prossimo capitolo vengono formalizzate le definizioni con cui abbiamo strutturato il modulo e gli attacchi con i quali abbiamo simulato una potenziale aggressione, col fine di calcolare la peggiore probabilità di re-identificazione di un individuo all'interno di un dataset composto da dati sequenziali generici.

Capitolo 3

Definizioni

In questo capitolo definiamo i formalismi matematici con i quali modellare i diversi tipi di dato utilizzati in questo progetto. Gestiamo dati sequenziali di tipo arbitrario, quindi dati non necessariamente di mobilità o di retail. Vediamo come contestualizzare questa generalizzazione dei dati in una struttura univoca di moderazione di calcolo del rischio. Infine definiamo ogni attacco, quantificandone il rischio empirico in funzione del metodo di calcolo con cui intendo valutarlo.

3.1 Modellazione dei Dati

Una sequenza è una successione di informazioni associata ad un individuo, la cui funzione è quella di raggruppare ed associare una serie di eventi a quella determinata sequenza. Ad esempio, in un dataset che tratta informazioni di mobilità di un certo individuo, una certa sequenza potrebbe definire una traiettoria o un percorso effettuato dal nostro utente, a sua volta composta da una progressione di informazioni che delineano il tipo della sequenza. In modo analogo, se trattassimo informazioni di retail, potremmo supporre che la sequenza rappresenti il basket di una certa spesa effettuata dall'utente. La sequenza può quindi generalizzare, indipendentemente dal contesto, un insieme di informazioni dello stesso tipo.

Definizione 1 ***Sequenza.*** *La sequenza d_u^i di indice i di un individuo u è una successione temporale ordinata di tuple $d_u^i = \langle (e_1, t_1, o_1), ..., (e_m, t_m, o_m) \rangle$ dove e_p è un generico elemento, t_p è il timestamp corrispondente mentre o_p è l'ordine sequenziale dei record letti della sequenza, con $t_p < t_q$, $o_p < o_q$ se $p < q \forall p$, $q \leq m$, con $m = |d_u^i|$.*

Si noti dalla definizione di sequenza come la notazione sia stata volutamente semplificata, infatti ogni tripla all'interno di d_u^i non tiene traccia ne dell'utente, ne dell'identificatore

della sequenza.
Ogni sequenza contiene quindi una successione di triple o record, ognuna delle quali, oltre al timestamp e all'ordine, definisce una variabile generica: l'elemento.
Gli elementi che compongono la sequenza sono di tipo variabile e possono assumere un tipo arbitrario, a seconda di che cosa devono generalizzare. Ritornando all'esempio precedente, potremmo immaginare un generico elemento che costituisce una traiettoria come una coppia di coordinate, le quali associate ad un certo timestamp e ad un ordine, compongono la tripla. La tripla realizzata può rappresentare uno dei tanti punti che compongono quella particolare traiettoria di un certo individuo. Analogamente, l'elemento di un basket potrebbe essere il nome del prodotto acquistato oppure una coppia, aggiungendo prezzo al nome. In tal caso la tripla descrive l'acquisto di un prodotto e la successione di esse definisce una certa spesa.
Lavorando con elementi generici la sequenza può ricoprire diversi ruoli. Ad esempio, si consideri il caso in cui un elemento rappresenti un'informazione di mobilità: la sequenza in tal caso identifica la traiettoria.

Definizione 2 ***Individuo.*** *Un individuo u è un insieme di sequenze* $D_u = \{d_u^1, d_u^2, \ldots, d_u^n\}$ *dove* $d_u^i (1 <= i <= n)$ *è la sequenza i-esima dell'individuo u.*

In conclusione, definiamo la macrostruttura che contiene tutte le informazioni precedentemente definite.

Definizione 3 ***Privacy Dataset.*** *Un Privacy Dataset è un insieme di individui* $D = \{D_1, D_2, \ldots, D_l\}$ *dove* $D_u (1 \leq u \leq l)$ *è l'insieme delle sequenze che definiscono l'individuo u, oppure potremmo anche dire che* D_u *è il privacy dataset di u.*

3.1.1 Strutture Ausiliarie

Le definizioni descritte nella sezione precedente sono quelle che definiscono un Privacy Dataset ma, a seconda della specifica applicazione che intendiamo utilizzare, posso avere bisogno di strutture di aggregazione differenti.

Esistono attacchi che necessitano di contare il numero di volte in cui un elemento occorre in una certa sequenza. Di conseguenza tali attacchi devono lavorare su strutture di aggregazione di dati differenti. A tale proposito è stato implementato il vettore di frequenza, che definiamo come segue:

Definizione 4 ***Vettore di Frequenza.*** *Il vettore di frequenza* W_u *appartenente all'individuo u è una successione di tuple* $W_u = \langle (e_1, w_1), \ldots, (e_n, w_n) \rangle$ *dove* e_i *è un elemento,*

w_i è la rispettiva frequenza, ad esempio il numero di volte che e_p appare in D_u, e $w_i > w_j$, se $i < j$.
In modo analogo, senza appesantire la notazione, definiamo con W_u^i il vettore di frequenza dell'i-esima sequenza di u. Un vettore di frequenza è quindi un'aggregazione dell'insieme di sequenze D_u.

In modo analogo definiamo il vettore di probabilità, con il quale viene calcolata la probabilità con cui un elemento occorre, ad esempio, in una certa sequenza. Tale struttura utilizza a sua volta il vettore di frequenza.

Definizione 5 ***Vettore di Probabilità.*** *Il vettore di probabilità P_u appartenente all'individuo u è una successione di tuple $P_u = \langle (e_1, p_1), \ldots, (e_n, p_n) \rangle$ dove e_i è un elemento, p_i è la probabilità che e_i appaia in W_u, ad esempio $p_i = w_i / \sum_{e_i \in W_u} w_i$, e $p_i > p_j$ se $i < j$.*
In modo analogo, senza appesantire la notazione, definiamo con P_u^i il vettore di probabilità dell'i-esima sequenza di u. Un vettore di probabilità è quindi un'aggregazione di un vettore di frequenza W_u.

La successiva struttura tiene traccia della proporzione di ogni elemento, ovvero del rapporto tra la frequenza dell'elemento corrente e di quello con frequenza maggiore.

Definizione 6 ***Vettore di Proporzione.*** *Il vettore di proporzione PP_u appartenente all'individuo u è una successione di tuple $PP_u = \langle (e_1, pp_1), ..., (e_n, pp_n) \rangle$ dove e_i è un elemento, pp_i è la proporzione tra e_i e l'elemento con frequenza maggiore in W_u, ad esempio, sia $(e_{max}, w_{max}) \in W_u$ l'elemento con frequenza massima, allora $pp_i = w_i / w_{max}$, con $e_i \neq e_{max}$ e $pp_i > pp_j$ se $i < j$.*
In modo analogo, senza appesantire la notazione, definiamo con PP_u^i il vettore di proporzione dell'i-esima sequenza di u. Un vettore di proporzione è quindi un'aggregazione di un vettore di frequenza W_u.

Successivamente definiamo alcuni insiemi, anzi, più precisamente multinsiemi. Questa sottile differenza garantisce che gli elementi contenuti in tali strutture possano essere ripetuti.
Le strutture Sequence Elements Multiset e Individual Elements Multiset permettono di filtrare tutti gli elementi rispettivamente di una sequenza e di un individuo.

Definizione 7 ***Sequence Elements Multiset.*** *É il multinsieme di elementi dell'i-esima sequenza di u, $E_{set}(d_u^i) = \{e_1, \ldots, e_m\}$,dove e_p è un elemento. $E_{set}(d_u^i)$ è un aggregazione dell'i-esima sequenza di u.*

Definizione 8 ***Individual Elements Multiset.*** *É il multinsieme di elementi dell'individuo u,* $E_{set}(D_u) = \{E_{set}(d_u^1) \cup, ..., \cup E_{set}(d_u^n)\}$*, dove* $E_{set}(d_u^i)$ *è il multinsieme di elementi dell'i-esima sequenza di u.* $E_{set}(D_u)$ *è un'aggregazione dell'insieme di sequenze di u.*

Successivamente consideriamo strutture che non solo prendono in considerazione elementi ripetuti ma garantiscono anche la sequenzialità di tali dati. Ad esempio, continuando l'esempio su dati di mobilità, esistono differenti attacchi in cui un potenziale aggressore non solo conosce alcuni punti di una traiettoria di un certo individuo, ma è al corrente anche dell'ordine con i quali sono stati visitati. Possiamo riportare lo stesso esempio anche sui prodotti di un basket.

Le strutture Sequence Elements Progression e Individual Elements Progression hanno come funzione quella di aggregare, rispettivamente, tutti gli elementi di una sequenza e di un individuo, mantenendo la sequenzialità.

Definizione 9 ***Sequence Elements Progression.*** *É la successione di tuple dell'i-esima sequenza di u,* $E_{seq}(d_u^i) = \langle (e_1, o_1), ..., (e_m, o_m) \rangle$ *,dove* e_p *è un elemento e* o_p *è la sequenzialità dell'elemento, con* $o_p < o_q$ *se* $p < q \forall p$*,* $q \leq m$*, con* $m = |d_u^i|$*.* $E_{seq}(d_u^i)$ *è un aggregazione di* d_u^i*.*

Definizione 10 ***Individual Elements Progression.*** *É una successione di tuple dell'individuo u,* $E_{seq}(D_u) = \langle E_{seq}(d_u^1) \cup, ..., \cup E_{seq}(d_u^n) \rangle$ *,dove* $E_{seq}(d_u^i)$ *è la successione di tuple dell'i-esima sequenza di u.* $E_{seq}(D_u)$ *è un aggregazione di* D_u*.*

Analogamente alle strutture precedenti, in questo caso le strutture filtrano sia l'elemento che il timestamp associato ad esso.

Definizione 11 ***Sequence Elements Time Progression.*** *É una successione di tuple dell'i-esima sequenza di u,* $ET_{set}(d_u^i) = \langle (e_1, t_1), \ldots, (e_m, t_m) \rangle$ *,dove* (e_p, t_p) *dove* e_p *è l'elemento e* t_p *il timestamp di quando è stato appreso, con* $t_p < t_q$ *se* $p < q \forall p$*,* $q \leq m$*, con* $m = |d_u^i|$*.* $ET_{set}(d_u^i)$ *è un aggregazione di* d_u^i*.*

Definizione 12 ***Individual Elements Time Progression.*** *É una successione di tuple dell'individuo u,* $ET_{set}(D_u) = \langle ET_{set}(d_u^1) \cup, ..., \cup ET_{set}(d_u^n) \rangle$*, dove* $ET_{set}(d_u^i)$ *è il multinsieme di coppie* (e_p, t_p) *dell'i-esima sequenza di u.* $ET_{set}(D_u)$ *è un'aggregazione di* D_u

3.2 Struttura di Moderazione di Calcolo del Rischio

In questa sezione descriviamo in generale il rischio di privacy o rischio di re-identificazione all'interno dello scenario presentato da PRUDEnce. Successivamente descriviamo i tipi di approcci con il quale calcolare il rischio, dando un formalismo al concetto di probabilità di re-identificazione e rischio di privacy.

3.2.1 PRUDEnce ed il Rischio di Privacy

In questa sezione presentiamo PRUDEnce [12]: il framework utilizzato in questa tesi per la valutazione sistematica del rischio di privacy.

Il framework considera uno scenario dove un data analyst (DA) richiede ad un data provider (DP) alcuni dati, necessari per offrire un certo servizio. Da parte sua, il provider deve garantire la privacy di ogni cliente registrato. Come primo passo, l'analista comunica al provider i requisiti a lui necessari per offrire il servizio. Supponiamo che il provider gestisca i propri clienti all'interno di un database D, il suo compito è quello di produrre un insieme di dataset $D_1, D_2, ..., D_n$ con l'obiettivo di andare a soddisfare le richieste dell'analista. I dataset prodotti rappresentano trasformazioni ottenute tramite aggregazione, selezione e filtraggio di D, ognuna di esse rappresenta una struttura dati differente.

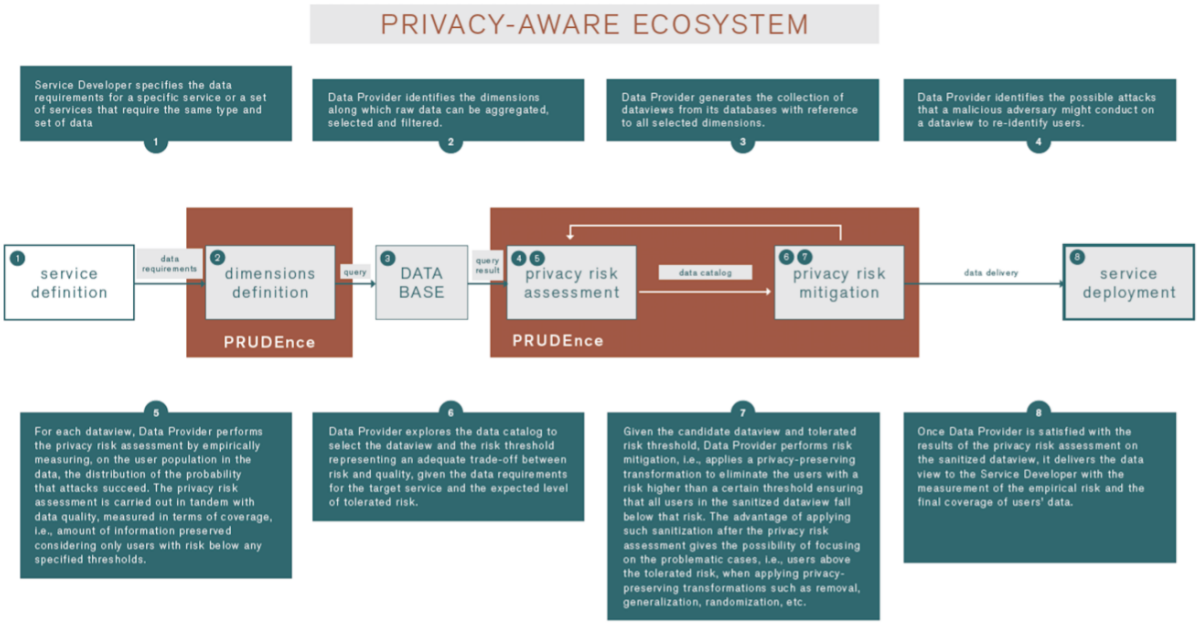


Figura 3.1: PRUDEnce ecosistema di condivisione e protezione dei dati

Il data provider a questo punto utilizza una procedura iterativa fino a quando i dati non si considerano sicuri. Solo a quel punto potranno essere consegnati all'analista. Un'iterazione della procedura è composta da quattro passi:

- *Identificazione dell'attacco*: identifico il potenziale insieme di attacchi che il malintenzionato può condurre con l'obiettivo di reidentificare gli individui nei dataset $D_1, D_2, ..., D_n$
- *Calcolo del rischio*: simulo gli attacchi e calcolo un insieme di rischi per ogni individuo dei dataset $D_1, ..., D_n$
- *Selezione del dataset*: seleziono un dataset $D \in D_1, D_2, ..., D_n$ con il miglior compromesso tra rischio di privacy degli individui e qualità dei dati, a partire da un certo livello di tolleranza del rischio e sulla base dei requisiti forniti dall'analista dei dati.
- *Mitigazione del rischio e consegna dei dati*: una volta selezionato il dataset, elimino gli individui che superano il range di tolleranza definito al principio, ottenendo un dataset filtrato D_{filt}. Il dataset sarà consegnato una volta considerato adeguatamente sanificato.

3.2.2 Valutazione Generale del Rischio

Il rischio di privacy di un individuo è strettamente legato alla sua probabilità di reidentificazione, ottenuta simulando un attacco. Infatti, l'aggressore, sulla base di una certa conoscenza, prova a re-identificare l'individuo sotto attacco all'interno dell'intero dataset. In particolare, utilizziamo il framework PRUDEnce adattandolo in maniera specifica sul nostro tipo di dato generico. In questo progetto utilizziamo la definizione di rischio di privacy (o rischio di re-identificazione) introdotta in [15][14][17] ed ampiamente utilizzata in letteratura.

La conoscenza dell'avversario, anche detta background knowledge, può essere associata a più categorie, ognuna delle quali può avere più configurazioni ed ogni configurazione può avere a sua volta più istanze.
La categoria del background knowledge rappresenta il tipo delle informazioni conosciute dall'avversario. Tecnicamente, indica le dimensioni del dato note all'avversario. Ad esempio l'aggressore può conoscere l'elemento, l'elemento e il tempo in cui è stato registrato, l'elemento e la frequenza con cui si ripete. L'elemento, come è stato descritto precedentemente, può rappresentare una qualsiasi informazione. Nel caso si considerassero dati di

retail, l'elemento potrebbe essere l'identificatore del prodotto o il suo nome ed il prezzo. Le configurazioni del background knowledge indicano il numero k di informazioni, appartenenti ad una certa categoria, conosciute dall'avversario. Ad esempio il malintenzionato potrebbe conoscere $k = 3$ informazioni relative ad u , dove potremmo immaginare che l'informazione conosciuta sia l'elemento e l'istante in cui è vi ha acceduto. Essenzialmente, le configurazioni di una base di conoscenza identificano la massima "quantità" di conoscenza che un avversario può possedere nei confronti di un individuo.
Infine, un'istanza di una certa conoscenza rappresenta la specifica informazione conosciuta dall'avversario, quindi il valore che associo ad un certo elemento. In estrema sintesi: categorie, configurazioni e istanze possono rispettivamente rispondere alle domande: "che tipo","quante" e "quali" sono le informazioni in possesso dell'aggressore? Successivamente diamo una formalizzazione ed alcuni esempi di questi concetti.

Definizione 13 ***Background Knowledge: Categoria, Configurazione e Istanza.*** *Data una categoria della conoscenza B, definiamo con $B_k = \{B_1, B_2, ..., B_n\} \in B$ una specifica configurazione del background knowledge, dove k rappresenta il numero degli elementi in B conosciuti dall'avversario e ogni elemento $b \in B_k$ definisce una specifica istanza della stessa configurazione.*

Esempio 1 *Supponiamo che $ET_{set}(d_u^i) = \langle (e_1, t_1), (e_2, t_2), (e_3, t_3), (e_4, t_4) \rangle$ sia l'i-esima sequenza dell'individuo u, dove e_p è un generico elemento e t_p il timestamp dell'elemento, con $p = 1, ..., 4$ e $t_p < t_q$ se $p < q$. Supponiamo che l'aggressore conosca due elementi che appartengono all'i-esima sequenza di u. Il provider non potendo conoscere quali sono questi due elementi, genererà tutte le combinazioni di sottosequenze con $k = 2$, ottenendo l'insieme $B_2 = \{((e_1, t_1), (e_2, t_2)), ((e_1, t_1), (e_3, t_3)), ((e_1, t_1), (e_4, t_4)), ((e_2, t_2), (e_3, t_3)), ((e_2, t_2), (e_4, t_4)), ((e_3, t_3), (e_4, t_4))\}$ contenente tutte le combinazioni di sottosequenze con $k = 2$. Ad esempio l'attaccante conosce $b = ((e_1, t_1), (e_4, t_4)) \in B_2$ e vuole re-identificare u tra tutte le sequenze di tutti gli individui del dataset.*

3.2.3 Elements Based Knowledge: Valutazione del Rischio

: Nel contesto in cui la conoscenza è basata solo sugli elementi, i record che appartengono ad un singolo individuo vengono considerati come un'unica grande struttura dati e genero le mie conoscenze come combinazione di elementi di questa grande macrostruttura.

Valutando il rischio con questo approccio, l'attaccante conosce le k informazioni relative ad un individuo u. Ad esempio, nel caso in cui si gestiscano dati di retail, l'attaccante può conoscere $k = 2$ prodotti che l'individuo ha acquistato. A differenza dell'approccio basato sulle sequenze, in questo caso i prodotti acquistati non devono necessariamente essere acquistati all'interno del solito basket.

Definizione 14 ***Probabilità di re-identificazione.*** *Dato un attacco, una funzione di matching(r,b) che indica se l'istanza $b \in B_k$ matcha positivamente nella struttura $r \in D$, e una funzione $M(D, b) = \{r \in D |$ matching(r,b) = True $\}$, definiamo la probabilità di re-identificazione di un individuo u in un dataset D come:*

$$PR_D(r = u|b) = \frac{1}{|M(D, b)|} \tag{3.1}$$

che è la probabilità di associare la struttura $r \in D$ all'individuo u, data l'istanza $b \in B_k$.

A questo punto, possiamo definire il rischio di privacy come il rischio di re-identificazione peggiore. Questo perchè il responsabile del dataset, non conoscendo quale sia l'istanza posseduta dall'aggressore, calcola la probabilità di re-identificazione di tutte le istanze $b \in B_k$. Possiamo formalizzarlo come segue:

Definizione 15 ***Rischio di privacy o Rischio di re-identificazione.*** *Sia B_k l'insieme di tutte le potenziali conoscenze che l'attaccante potrebbe sapere, allora il rischio di privacy di un individuo u è dato da $Risk(u, D) = maxPR_D(r = u|b)$, dove $Risk(u, D)$ è la massima probabilità di re-identificazione, per ogni $b \in B_k$. Ovviamente $Risk(u, D) = 0$ se $u \notin D$.*

3.2.4 Sequence Based Knowledge: Valutazione del Rischio

Nel contesto in cui la conoscenza è basata solo sulle sequenze, i record che appartengono ad una singola sequenza di un certo individuo vengono considerati come un'unica grande struttura dati e genero le mie conoscenze come combinazione di elementi di tutte le macrostrutture dell'individuo.
In particolare, si considera l'unione delle combinazioni di elementi ottenute disgiuntamente su ogni struttura che rappresenta una sequenza dello stesso individuo.

L'attaccante, quindi, conosce k informazioni relative ad una sequenza di u. Ad esempio, nel caso si gestiscano dati di mobilità, l'attaccante può conoscere $k = 3$ punti visitati da un individuo in una determinata traiettoria.

Definizione 16 ***Probabilità di re-identificazione.*** *Dato un attacco, una funzione di matching(r,b) che indica se l'istanza $b \in B_k$ matcha positivamente nella struttura $r \in \cup_{i=1}^{n} D_i$, una funzione $M_N(D_u, b) = \{r \in D_u |$ matching(r,b) = True $\}$, una funzione $M_D(\cup_{i=1}^{n} D_i, b) = \{r \in \cup_{i=1}^{n} D_i |$ matching(r,b)=True$\}$ definiamo la probabilità di re-identificazione di un individuo u in un dataset D come:*

$$PR_D(r = u|b) = \frac{|M_N(D_u, b)|}{|M_D(\cup_{i=1}^{n} D_i, b)|} \tag{3.2}$$

che è la probabilità di associare la struttura $r \in \cup_{i=1}^{n} D_i$ *all'individuo u, data l'istanza* $b \in B_k$.

Il rischio di privacy, rispetto al precedente metodo non varia. La differenza sta nella generazione del background knowlege. Infatti, come vediamo nella prossima sezione, l'insieme delle potenziali conoscenze che un attaccante potrebbe avere nei confronti di un individuo viene definito come l'unione di ogni background knowledge applicato ad ogni sequenza di *u*.

Definizione 17 ***Rischio di privacy o Rischio di re-identificazione.*** *Sia* B_k *l'insieme di tutte le potenziali conoscenze che l'attaccante potrebbe sapere, allora il rischio di privacy di un individuo u è dato da* $Risk(u, D) = maxPR_D(r = u|b)$*, dove* $Risk(u, D)$ *è la massima probabilità di re-identificazione, per ogni* $b \in B_k$*. Ovviamente* $Risk(u, D) = 0$ *se* $u \notin D$.

3.2.5 Full Sequence Knowledge: Valutazione del Rischio

Quando la conoscenza è relativa al contenuto dell'intera sequenza, i record che appartengono ad una singola sequenza di un certo individuo vengono considerati come un'unica grande struttura dati e genero le mie conoscenze come combinazione di elementi di tutte le macrostrutture dell'individuo.

Questa conoscenza presuppone che l'attaccante conosca l'intero contenuto di k sequenze di un certo individuo *u*. É una conoscenza molto forte. Ad esempio un'attaccante potrebbe conoscere tutti i prodotti acquistati da un individuo durante una spesa.

In questo caso sia la probabilità di re-identificazione che il rischio si definiscono esattamente come nel primo metodo. La differenza sta nella funzione di matching, il cui compito è quello di confrontare il contenuto delle sequenze.

Definizione 18 ***Probabilità di re-identificazione.*** *Dato un attacco, una funzione di matching(r,b) che indica se l'istanza* $b \in B_k$ *matcha positivamente nella struttura* $r \in D$*, e una funzione* $M(D, b) = \{r \in D|$ *matching(r,b) = True* $\}$*, definiamo la probabilità di re-identificazione di un individuo u in un dataset D come:*

$$PR_D(r = u|b) = \frac{1}{|M(D, b)|} \tag{3.3}$$

che è la probabilità di associare la struttura $r \in D$ *all'individuo u, data l'istanza* $b \in B_k$.

Definizione 19 ***Rischio di privacy o Rischio di re-identificazione.*** *Sia B_k l'insieme di tutte le potenziali conoscenze che l'attaccante potrebbe sapere, allora il rischio di privacy di un individuo u è dato da $Risk(u, D) = maxPR_D(r = u|b)$, dove $Risk(u, D)$ è la massima probabilità di re-identificazione, per ogni $b \in B_k$. Ovviamente $Risk(u, D) = 0$ se $u \notin D$.*

3.3 Modellazione degli Attacchi

In questa sezione, formalizziamo tutti gli attacchi realizzati. Ognuno di essi rappresenta una potenziale conoscenza dell'attaccante. Il calcolo del rischio posso quantificarlo sulla base di tre approcci, ognuno dei quali ha un raggio d'azione differente.

3.3.1 Elements Attack

Un Elements Attack è una tipologia di attacco nel quale il malintenzionato prova a re-identificare un determinato individuo in base al numero di elementi visitati da quest'ultimo, ma non è al corrente dell'ordine temporale nel quale sono stati appresi.

A seconda del metodo con il quale calcolo il rischio, l'elements background knowledge sarà composto a partire da differenti insiemi di elementi e conseguentemente da differenti configurazioni di quest'ultimi. Lo stesso vale per la funzione matching. Definiamo adesso tale attacco, in funzione di come il rischio viene valutato.

Elements Based Knowledge Valutando il rischio con una elements based knowledge, il sottoinsieme di k elementi conosciuti dall'avversario appartiene ad un certo individuo u. Ad esempio, nel caso in cui il nostro dataset sia composto da dati di mobilità, possiamo immaginare che l'attaccante sia a conoscenza dei k luoghi visitati dall'individuo che intendo reidentificare.

Sia k il numero di elementi e_p di un individuo u conosciuti dal malintenzionato. Allora l'elements background knowledge, in un contesto elements based, è rappresentato dall'insieme di tutte le possibili configurazioni di k elementi, definito come segue:

$$B_k = E_{set}(D_u)^{[k]} \tag{3.4}$$

dove $E_{set}(D_u)^{[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi e_p ottenute a partire dall'insieme di tutti gli elementi di u: $E_{set}(D_u)$.

Poiché ogni istanza $b \in B_k$ è un sottoinsieme di elementi $X_u \subseteq E_{set}(D_u)$ di lunghezza k, allora sia $r = E_{set}(D_u) \in D$ la struttura contenente tutti gli elementi di un generico individuo u, la funzione di matching è definita come segue:

$$matching(r, b) = \begin{cases} true & b \subseteq r \\ false & altrimenti \end{cases} \tag{3.5}$$

Sequence Based Knowledge Calcolando il rischio con una sequence based knowledge, l'attaccante è a conoscenza di k elementi appartenenti ad una certa sequenza dell'individuo sotto attacco.

Sia k il numero di elementi e_p dell'*i-esima* sequenza di un individuo u conosciuti dall'avversario. L'elements background knowledge, in un contesto sequence based,è rappresentato dall'unione, per ogni sequenza di u, di tutti gli insiemi di configurazioni di k elementi, definito come segue:

$$B_k = \bigcup_{i=1}^{n} E_{set}(d_u^i)^{[k]} \tag{3.6}$$

dove $E_{set}(d_u^i)^{[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi e_p ottenute a partire da tutti gli elementi dell'*i-esima* sequenza di u: $E_{set}(d_u^i)$.

Poiché ogni istanza $b \in B_k$ è un sottoinsieme di elementi $X_u \subseteq E_{set}(d_u^i)$ di lunghezza k: allora sia $r = E_{set}(D_u) \in D$ la struttura contenente tutti gli elementi di un generico individuo u, la funzione di matching è definita come segue:

$$matching(r, b) = \begin{cases} true & \exists i : s = E_{set}(d_u^i) \in r \mid b \subseteq s \\ false & altrimenti \end{cases} \tag{3.7}$$

Full Sequence Knowledge In questo caso, l'avversario è a conoscenza dell'intero contenuto di k sequenze, o meglio di tutti gli elementi che le compongono.

Definiamo con H_u l'insieme di tutti gli indici delle sequenze di u, dove $\forall i \in H_u$, allora i è l'indice della *i-esima* sequenza di u, con $i > 0$.
Sia k il numero di sequenze di un individuo u conosciute interamente da un avversario. Allora l'elements background knowledge, in un contesto full sequence,è rappresentato dall'insieme di tutte le possibili configurazioni di k sequenze, definito come segue:

$$B_k = {H_u}^{[k]} \tag{3.8}$$

dove ${H_u}^{[k]}$ rappresenta l'insieme di tutte le possibili combinazioni degli indici di k sequenze ottenute a partire da tutti gli indici delle sequenze di u: H_u.

Poiché ogni istanza $b \in B_k$ è un sottoinsieme di indici di sequenze $X_u \subseteq H_u$ di lunghezza k: allora sia $r = H_u \in D$ la struttura di un ad un generico individuo u, la funzione di matching è definita come segue:

$$matching(r, b) = \begin{cases} true & \forall i \in b,\ \exists j \in r \mid E_{set}(d_u^i) = E_{set}(d_u^j) \\ false & altrimenti \end{cases} \tag{3.9}$$

Per non complicare eccessivamente la notazione, con $E_{set}(d_u^i) = E_{set}(d_u^j)$ si presuppone che l'operatore di uguaglianza confronti l'intero contenuto di entrambe le sequenze: ovvero che svolga il seguente confronto: $\forall e_p \in E_{set}(d_u^i) \wedge \forall e_q \in E_{set}(d_u^j) \mid e_p = e_q$.

3.3.2 Elements Sequence Attack

Un Elements Sequence Attack è un attacco nel quale l'avversario conosce un sottoinsieme di elementi appartenenti ad determinato individuo e l'ordine temporale nel quale quest'ultimo vi ha acceduto.

In base al metodo con il quale calcolo il rischio, definisco un elements sequence background knowledge ed una funzione di matching differenti, tenendo conto in questo caso dell'ordine non trascurabile con cui l'individuo accede agli elementi.

Elements Based Knowledge Calcolando il rischio in questo modo, l'avversario conosce un sottoinsieme ordinato di k elementi che appartengono ad un certo individuo u. Ad esempio, nel caso in cui il nostro dataset sia composto da dati di retail, possiamo supporre che l'attaccante sappia che l'individuo ha acquistato k prodotti in un certo ordine.

Sia k il numero ordinato di elementi (e_p, o_p) di un individuo u conosciuti dall'attaccante. L'elements sequence background knowledge, in un contesto elements based, è l'insieme di tutte le possibili configurazioni di k elementi, definito come segue:

$$B_k = E_{seq}(D_u)^{[k]} \tag{3.10}$$

dove $E_{seq}(D_u)^{[k]}$ rappresenta l'insieme di tutte le possibili k-sottosequenze di elementi e_p ottenute a partire da tutti gli elementi di u: $E_{seq}(D_u)$

Indichiamo con $a \preceq b$ il concetto che a è una sottosequenza di b. Poichè ogni istanza $b \in B_k$ è una sottosequenza di elementi $X_u \subseteq E_{seq}(D_u)$ di lunghezza k, allora sia $r = E_{seq}(D_u) \in D$ la struttura appartenente ad un generico individuo u, la funzione di matching è così definita:

$$matching(r, b) = \begin{cases} true & b \preceq r \\ false & altrimenti \end{cases} \tag{3.11}$$

Sequence Based Knowledge Valutando il rischio con una sequence based knowledge, l'aggressore è a conoscenza di un sottoinsieme ordinato di k elementi appartenenti ad una sequenza di un certo individuo u. Ad esempio, in un dataset di dati di retail, possiamo supporre che l'attaccante sappia che l'individuo ha acquistato k prodotti in un certo ordine in una unica spesa.

Sia k il numero di elementi (e_p, o_p) dell'*i-esima* sequenza di un individuo u conosciuti dall'avversario. L'elements sequence background knowledge,in un contesto sequence based, è rappresentato dall'unione, per ogni sequenza di u, di tutti gli insiemi di configurazioni di k elementi, definito come segue:

$$B_k = \bigcup_{i=1}^{n} E_{seq}(d_u^i)^{[k]} \tag{3.12}$$

dove $E_{seq}(d_u^i)^{[k]}$ rappresenta l'insieme di tutte le possibili k-sottosequenze di elementi (e_p, o_p) ottenute a partire da tutti gli elementi dell'*i-esima* sequenza di u: $E_{seq}(d_u^i)$

Poiché ogni istanza $b \in B_k$ è una sottosequenza di elementi $X_u \subseteq E_{seq}(d_u^i)$ di lunghezza k: allora sia $r = E_{seq}(D_u) \in D$ la struttura dati appartenente ad un generico individuo u, la funzione di matching è definita come segue:

$$matching(r, b) = \begin{cases} true & \exists i : s = E_{seq}(d_u^i) \in r \mid b \preceq r \\ false & altrimenti \end{cases} \tag{3.13}$$

Full Sequence Knowledge L'attaccante conosce l'intero contenuto di k sequenze, ovvero di tutti gli elementi, in ordine temporale, che la compongono.

Definiamo con H_u l'insieme di tutti gli indici delle sequenze di u, dove $\forall i \in H_u$, allora i è l'indice della *i-esima* sequenza di u, con $i > 0$.
Sia k il numero di sequenze di un individuo u conosciute per intero dall'aggressore. Allora

l'elements sequence background knowledge, in un contesto full sequence, è rappresentato dall'insieme di tutte le possibili configurazioni di indici di k sequenze, definito come segue:

$$B_k = H_u^{[k]} \tag{3.14}$$

dove ${H_u}^{[k]}$ rappresenta l'insieme di tutte le possibili combinazioni degli indici di k sequenze ottenute a partire da tutti gli indici delle sequenze di u: H_u.

Poiché ogni istanza $b \in B_k$ è un sottoinsieme di indici di sequenze $X_u \subseteq H_u$ di lunghezza k, allora sia $r = H_u \in D$ la struttura appartenente ad un generico individuo u, la funzione di matching è definita come segue:

$$matching(r, b) = \begin{cases} true & \forall i \in b,\ \exists j \in r \mid E_{seq}(d_u^i) = E_{seq}(d_u^j) \\ false & altrimenti \end{cases} \tag{3.15}$$

Per non complicare eccessivamente la notazione, si presuppone con $E_{seq}(d_u^i) = E_{seq}(d_u^j)$ che l'operatore di uguaglianza confronti l'intero contenuto di entrambe le sequenze: ovvero che svolga il seguente confronto: $\forall(e_p, o_p) \in E_{seq}(d_u^i) \wedge \forall(e_q, o_q) \in E_{seq}(d_u^j) | e_p = e_q \wedge o_p = o_q$.

3.3.3 Elements Time Attack

Un Elements Time Attack è un tipo di attacco nel quale il malintenzionato conosce un sottoinsieme di elementi appartenenti ad un individuo e l'istante nel quale sono stati appresi.

La conoscenza del tempo può essere definita in più termini di raggruppamento: ad esempio l'aggressore può sapere il giorno in cui ha visitato una certa città e non il secondo o il minuto preciso nel quale è accaduto. In base al metodo con il quale calcolo il rischio, definisco un elements time background knowledge ed una funzione di matching differenti.

Elements Based Knowledge In un approccio elements based, l'attaccante conosce sia il sottoinsieme di k elementi appartenenti ad un individuo che i k istanti nel quale sono stati appresi. Ad esempio, nel caso in cui il nostro dataset sia composto da dati di mobilità, possiamo supporre che l'attaccante conosca gli istanti con cui il l'individuo ha visitato k città.

Sia k il numero di elementi (e_p, t_p) di un individuo u conosciuti dal malintenzionato. Allora l'elements time background knowledge, in un contesto elements based, è rap-

presentato dall'insieme di tutte le possibili configurazioni di k elementi, definito come segue:

$$B_k = ET_{seq}(D_u)^{[k]} \tag{3.16}$$

dove $ET_{seq}(D_u)^{[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi (e_p, t_p) ottenute a partire da tutti gli elementi di u: $ET_{seq}(D_u)$

Poichè ogni istanza $b \in B_k$ è una sottosequenza di elementi X_u di lunghezza k, allora sia $r = ET_{seq}(D_u) \in D$ la struttura appartentente ad un generico individuo u, la funzione di matching è così definita:

$$matching(r, b) = \begin{cases} true & \forall (e_p, t_p) \in b,\ \exists (e_q, t_q) \in r \mid e_p = e_q \wedge t_p = t_q \\ false & altrimenti \end{cases} \tag{3.17}$$

Sequence Based Knowledge Valutando il rischio in un contesto sequence based, l'avversario conosce sia i k elementi che i k tempi appartenenti ad una certa sequenza dell'individuo.
In pratica, supponendo di avere dati di mobilità, l'attaccante conosce i momenti in cui l'individuo ha visitato k luoghi durante un suo spostamento.

Sia k il numero di elementi (e_p, t_p) dell'*i-esima* di un individuo u conosciuti dall'avversario. L'elements time background knowledge, in un contesto sequence based, è rappresentato dall'unione, per ogni sequenza di u, di tutti gli insiemi di configurazioni di k elementi, definito come segue:

$$B_k = \bigcup_{i=1}^{n} ET_{seq}(d_u^i)^{[k]} \tag{3.18}$$

dove $ET_{seq}(d_u^i)^{[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi (e_p, t_p) ottenute a partire da tutti gli elementi dell'*i-esima* sequenza di u: $ET_{seq}(d_u^i)$

Poiché ogni istanza $b \in B_k$ è un sottoinsieme di elementi $X_u ET_{seq}(d_u^i)$ di lunghezza k: allora sia $r = ET_{seq}(D_u) \in D$ la struttura dati appartenente ad un generico individuo u, la funzione di matching è definita come segue:

$$matching(r,b) = \begin{cases} true & \exists s = ET_{seq}(d_u^i),\ \forall (e_p,t_p) \in b,\ \exists (e_q,t_q) \in s \mid \\ & e_p = e_q \wedge t_p = t_q \\ \\ false & altrimenti \end{cases} \tag{3.19}$$

Full Sequence Knowledge L'attaccante conosce l'intero contenuto di k sequenze, ovvero di tutti gli elementi con i rispettivi tempi che le compongono.
Ad esempio, in caso di dati di mobilità, l'attaccante potrebbe aver pedinato il nostro individuo, conoscendo tutte le tappe con i rispettivi orari di una o più traiettorie.

Definiamo con H_u l'insieme di tutti gli indici delle sequenze di u, dove $\forall i \in H_u$, allora i è l'indice della *i-esima* sequenza di u, con $i > 0$.
Sia k il numero di sequenze di un individuo u conosciute interamente da un avversario. Allora l'elements time background knowledge, in un contesto full sequence,è rappresentato dall'insieme di tutte le possibili configurazioni di indici di k sequenze, definito come segue:

$$B_k = H_u^{[k]} \tag{3.20}$$

dove ${H_u}^{[k]}$ rappresenta l'insieme di tutte le possibili combinazioni degli indici di k sequenze ottenute a partire da tutti gli indici delle sequenze di u: H_u.

Poiché ogni istanza $b \in B_k$ è un sottoinsieme di indici di sequenze $X_u \subseteq H_u$ di lunghezza k, allora sia $r = H_u \in D$ la struttura appartenente ad un generico individuo u, la funzione di matching è definita come segue:

$$matching(r,b) = \begin{cases} true & \forall i \in b,\ \exists j \in r \mid ET_{seq}(d_u^i) = ET_{seq}(d_u^j) \\ false & altrimenti \end{cases} \tag{3.21}$$

Per non complicare eccessivamente la notazione, si presuppone con $ET_{seq}(d_u^i) = ET_{seq}(d_u^j)$ che l'operatore di uguaglianza confronti l'intero contenuto di entrambe le sequenze: ovvero che svolga il seguente confronto: $\forall (e_p,t_p) \in ET_{seq}(d_u^i) \wedge \forall (e_q,t_q) \in ET_{seq}(d_u^j) | e_p = e_q \wedge t_p = t_q$.

3.3.4 Frequency Attack

In un Frequency Attack, l'avversario conosce un sottoinsieme di elementi appartenenti ad un certo individuo e la frequenza con la quale questi occorrono.

Per realizzare ciò devo necessariamente convertire il dataset D, in un vettore di frequenza W, valutando il rischio su quest'ultimo. Ricordiamo che ogni elemento appartenente a W è definito da una coppia $(e_p, w_p) \in W$ che rappresenta l'elemento e rispettiva frequenza. Inoltre, ricordiamo che con W_u e W_u^i indichiamo rispettivamente i vettori di frequenza di u e dell'*i-esima* sequenza di u.

Elements Based Knowledge Con un approccio elements based, l'attaccante conosce sia il sottoinsieme di k elementi appartenenti ad un individuo che le k frequenze con i quali occorrono. Ad esempio, nel caso in cui il nostro dataset sia composto da dati di retail, possiamo supporre che l'attaccante conosca quante volte l'individuo u ha acquistato un certo prodotto.

Sia k il numero di elementi (e_p, w_p) del vettore di frequenza di un individuo u conosciuti dall'aggressore. Il frequency background knowledge, in un contesto elements based, è rappresentato dall'insieme di tutte le possibili configurazioni di k elementi, definito come segue:

$$B_k = W_u^{[k]} \tag{3.22}$$

dove $W_u^{[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi (e_p, w_p) ottenute a partire da tutti gli elementi del vettore di frequenza di u: W_u.

Poichè ogni istanza $b \in B_k$ è un vettore di frequenza, allora sia $r = W_u \in W$, la funzione di matching è così definita:

$$matching(r, b) = \begin{cases} true & \forall (e_p, w_p) \in b,\ \exists (e_q, w_q) \in r \ \mid \\ & e_p = e_q \wedge w_p \in [w_q - \delta, w_q + \delta] \\ \\ false & altrimenti \end{cases} \tag{3.23}$$

Sequence Based Knowledge In questo caso, l'attaccante conosce sia il sottoinsieme di k elementi appartenenti alla sequenza di un individuo che le k frequenze con i quali occorrono. Ad esempio, con un dataset di dati di retail, possiamo immaginare che l'attaccante conosca quante volte l'individuo u ha acquistato un certo prodotto durante una

determinata spesa.

Sia k il numero di elementi (e_p, w_p) dell'*i-esima* sequenza di un individuo u conosciuti dal avversario. Il frequency background knowledge, in un contesto sequence based, è rappresentato dall'unione, per ogni sequenza di u, di tutti gli insiemi di configurazioni di k elementi, definito come segue:

$$B_k = \bigcup_{i=1}^{n} W_u^{i\,[k]} \tag{3.24}$$

dove $W_u^{i\,[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi (e_p, w_p) ottenute a partire da tutti gli elementi dell'*i-esima* sequenza del vettore di frequenza di u: W_u^i

Poichè ogni istanza $b \in B_k$ è un vettore di frequenza, allora sia $r = W_u \in W$, la funzione di matching è così definita:

$$matching(r, b) = \begin{cases} true & \exists i : s = w_u^i \in r,\ \forall (e_p, w_p) \in b,\ \exists (e_q, w_q) \in s \ \mid \\ & e_p = e_q \wedge w_p \in [w_q - \delta, w_q + \delta] \\ \\ false & altrimenti \end{cases} \tag{3.25}$$

Full Sequence Knowledge L'attaccante conosce l'intero contenuto di k sequenze, ovvero tutti gli elementi con le rispettive frequenze di ognuna.

Definiamo con H_u l'insieme di tutti gli indici delle sequenze di u, dove $\forall i \in H_u$, allora i è l'indice di dell' *i-esima* sequenza del vettore di frequenza, W_u , tale che $i > 0$.
Sia k il numero di sequenze di un individuo u conosciute interamente da un avversario. Allora il frequency background knowledge, in un contesto full sequence,è rappresentato dall'insieme di tutte le possibili configurazioni di indici di k sequenze, definito come segue:

$$B_k = H_u^{[k]} \tag{3.26}$$

dove ${H_u}^{[k]}$ rappresenta l'insieme di tutte le possibili combinazioni degli indici di k sequenze ottenute a partire da tutti gli indici delle sequenze di u: H_u.

Poiché ogni istanza $b \in B_k$ è un sottoinsieme di indici di sequenze di W_u di lunghezza k, allora sia $r = H_u \in W_u$ la struttura appartenente ad un generico individuo u, la funzione di matching è definita come segue:

$$matching(r, b) = \begin{cases} true & \forall i \in b,\ \exists j \in r \mid W_u^i = W_u^j \\ \\ false & altrimenti \end{cases} \tag{3.27}$$

Per non complicare eccessivamente la notazione, si presuppone con $W_u^i = W_u^j$ che l'operatore di uguaglianza confronti l'intero contenuto di entrambe le sequenze: ovvero che svolga il seguente confronto: $\forall(e_p, w_p) \in W_u^i \wedge \forall(e_q, w_q) \in w_u^j \mid e_p = e_q \wedge w_p \in [w_q - \delta, w_q + \delta]$

3.3.5 Probability Attack

In un Probability Attack, l'attaccante è a conoscenza di un sottoinsieme di elementi appartenenti ad un certo individuo e alla probabilità con la quale questi occorrono.

A differenza del frequency attack, in questo caso il rischio viene valutato su un vettore di probabilità P. Ricordiamo che ogni elemento appartenente a P è rappresentato da una coppia $(e_p, p_p) \in P$ contenente l'elemento e la rispettiva probabilità. Inoltre, ricordiamo che con P_u e P_u^i si definiscono rispettivamente i vettori di probabilità di u e dell'*i-esima* sequenza di u.

Elements Based Knowledge L'attaccante, valutando il rischio con questo approccio, conosce sia il sottoinsieme di k elementi appartenente ad un individuo u che le k probabilità con le quali tali elementi occorrono. Ad esempio, con dati di tipo retail, il malintenzionato potrebbe sapere che l'individuo ha acquistato un prodotto con una certa probabilità e tentare di reidentificarlo.

Sia k il numero di elementi (e_p, p_p) del vettore di probabilità di un individuo u conosciuti dall'attaccante. Il probability background knowledge, in un contesto elements based, è rappresentato dall'insieme di tutte le possibili configurazioni di k elementi, definito come segue:

$$B_k = P_u^{[k]} \tag{3.28}$$

dove $P_u^{[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi (e_p, p_p) ottenute a partire da tutti gli elementi del vettore di probabilità di u: P_u.

Poichè ogni istanza $b \in B_k$ è un vettore di probabilità di lunghezza k, allora sia $r = W_u \in W$ la struttura appartenente ad un generico individuo u, la funzione di matching è così definita:

$$matching(r, b) = \begin{cases} true & \forall (e_p, p_p) \in b,\ \exists (e_q, p_q) \in r \ \mid \\ & e_p = e_q \wedge p_p \in [p_q - \delta, p_q + \delta] \\ \\ false & altrimenti \end{cases} \tag{3.29}$$

Sequence Based Knowledge Valutando il rischio con una sequence based knowledge, la conoscenza dell'avversario è relativa ad una sequenza dell'individuo: conoscerà sia i k elementi che le probabilità rispettive.

Sia k il numero di elementi (e_p, p_p) dell'*i-esima* sequenza di un individuo u conosciuti dall'avversario. Il probability background knowledge, in un contesto sequence based, è rappresentato dall'unione, per ogni sequenza di u, di tutti gli insiemi di configurazioni di k elementi, definito come segue:

$$B_k = \bigcup_{i=1}^{n} P_u^{i\,[k]} \tag{3.30}$$

dove $P_u^{i\,[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi (e_p, p_p) ottenute a partire da tutti gli elementi dell'*i-esima* sequenza del vettore di probabilità di u: P_u^i.

Poichè ogni istanza $b \in B_k$ è un vettore di probabilità, allora sia $r = W_u \in W$, la funzione di matching è così definita:

$$matching(r, b) = \begin{cases} true & \exists i : s = P_u^i \in r,\ \forall (e_p, p_p) \in b,\ \exists (e_q, p_q) \in s \ \mid \\ & e_p = e_q \wedge p_p \in [p_q - \delta, p_q + \delta] \\ \\ false & altrimenti \end{cases} \tag{3.31}$$

Full Sequence Knowledge In questo caso, l'avversario conosce l'intero contenuto di k sequenze, nonchè tutti gli elementi con le rispettive probabilità.

Definiamo con H_u l'insieme di tutti gli indici delle sequenze di u, dove $\forall i \in H_u$, allora i è l'indice di dell' *i-esima* sequenza del vettore di probabilità, P_u , tale che $i > 0$.

Sia k il numero di sequenze di un individuo u conosciute interamente da un avversario. Allora il probability background knowledge, in un contesto full sequence, 'è rappresentato dall'insieme di tutte le possibili configurazioni di indici di k sequenze, definito come segue

$$B_k = H_u^{[k]} \tag{3.32}$$

dove ${H_u}^{[k]}$ rappresenta l'insieme di tutte le possibili combinazioni degli indici di k sequenze ottenute a partire da tutti gli indici delle sequenze di u: H_u.

Poiché ogni istanza $b \in B_k$ è un sottoinsieme di indici di sequenze di P_u di lunghezza k, allora sia $r = H_u \in P_u$ la struttura appartenente ad un generico individuo u, la funzione di matching è definita come segue:

$$matching(r, b) = \begin{cases} true \quad \forall i \in b,\ \exists j \in r \ \mid \ P_u^i = P_u^j \\ \\ false \quad altrimenti \end{cases} \tag{3.33}$$

Per non complicare eccessivamente la notazione, si presuppone con $P_u^i = P_u^j$ che l'operatore di uguaglianza confronti l'intero contenuto di entrambe le sequenze: ovvero che svolga il seguente confronto: $\forall (e_p, p_p) \in P_u^i \wedge \forall (e_q, p_q) \in P_u^j \mid e_p = e_q \wedge p_p \in [p_q - \delta, p_q + \delta]$.

3.3.6 Proportion Attack

In un Proportion Attack assumiamo che l'avversario conosca un sottoinsieme di elementi appartenenti ad un individuo u e le relative proporzioni.
In particolare, l'avversario conosce la proporzione tra la frequenza di ciascun elemento con quello avente frequenza maggiore.

Per realizzare ciò devo necessariamente convertire il privacy dataset D, in un vettore di proporzione PP, calcolando il rischio su quest'ultimo. Ricordiamo che ogni elemento appartenente a PP è definito da una coppia $(e_p, pp_p) \in PP$ il quale rappresenta l'elemento e la rispettiva proporzione. Inoltre, ricordiamo che con PP_u e PP_u^i indichiamo rispettivamente i vettori di proporzione di u e dell'*i-esima* sequenza di u.

Elements Based Knowledge Con un approccio elements based, l'attaccante conosce sia il sottoinsieme di k elementi appartenenti ad un individuo che le k proporzioni con le quali occorrono.

Sia k il numero di elementi $(e_p, pp_p) \in PP_u$ di un individuo u conosciuti dal malintenzionato. Il proportion background knowledge, in un contesto elements based, è rappresentato dall'insieme di tutte le possibili configurazioni di k elementi, definito come segue:

$$B_k = PP_u^{[k]} \tag{3.34}$$

dove $PP_u^{[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi (e_p, pp_p) ottenute a partire da tutti gli elementi del vettore di proporzione di u: PP_u.

Poichè ogni istanza $b \in B_k$ è un vettore di proporzione di lunghezza k, allora sia $r = PP_u \in PP$ la struttura contenente tutti gli elementi con le rispettive proporzioni di un individuo u, definiamo la funzione di matching come segue:

$$matching(r, b) = \begin{cases} true & \forall (e_p, pp_p) \in b,\ \exists (e_q, pp_q) \in r \ \mid \\ & e_p = e_q \wedge pp_p \in [pp_q - \delta, pp_q + \delta] \\ false & altrimenti \end{cases} \tag{3.35}$$

Sequence Based Knowledge In questo caso, l'attaccante conosce sia il sottoinsieme di k elementi appartenenti ad una sequenza dell'individuo che le k proporzioni con le quali occorrono.

Sia k il numero di elementi $(e_p, pp_p) \in PP_u^i$ dell'*i-esima* sequenza di un individuo u conosciuti dal malintenzionato. Il proportion background knowledge, in un contesto sequence based, è rappresentato dall'unione, per ogni sequenza di u, di tutti gli insiemi di configurazioni di k elementi, definito come segue:

$$B_k = \bigcup_{i=1}^{n} PP_u^{i\,[k]} \tag{3.36}$$

dove $PP_u^{i\,[k]}$ rappresenta l'insieme di tutte le possibili k-combinazioni di elementi (e_p, pp_p) ottenute a partire da tutti gli elementi dell'*i-esima* sequenza del vettore di u: PP_u^i.

Poichè $b \in B_k$ è il vettore di proporzione di lunghezza k, allora sia $r = PP_u \in PP$ la struttura contenente tutti gli elementi con le rispettive proporzioni di un individuo u,

definiamo la funzione di matching come segue:

$$matching(r, b) = \begin{cases} true & \exists i : s = PP_u^i \in PP,\ \forall (e_p, pp_p) \in b,\ \exists (e_q, pp_q) \in s \ \mid \\ & e_p = e_q \wedge pp_p \in [pp_q - \delta, pp_q + \delta] \\ \\ false & altrimenti \end{cases} \tag{3.37}$$

Capitolo 4

Libreria

Abbiamo utilizzato il framework PRUDEnce adattandolo in maniera specifica a dati sequenziali generici in modo tale da sviluppare una libreria che non trattasse soltanto dati di mobilità ma dati sequenziali di tipo arbitrario. La libreria PRIVSeq concretizza questa idea. Scritta interamente in Python l'intero insieme di dati viene gestito e manipolato in dataframes della libreria Pandas, la quale mi garantisce di utilizzare procedure performanti senza andare ad intaccare le performances di calcolo.In questo capitolo descriviamo la struttura e la realizzazione della libreria: prestando attenzione ai moduli e alle scelte implementative adottate durante lo sviluppo.

4.1 Analisi e Struttura del Problema

L'analisi del problema rispecchia lo scenario introdotto nel capitolo precedente, nel quale il responsabile dei dati di un provider deve fornire ad un analista un dataset, garantendo la privacy dei clienti. In particolare, la libreria implementa il secondo step della procedura definendo come istanza del problema, la valutazione del rischio di privacy associata ad ogni individuo del dataset in questione, calcolata mediante la simulazione di un attacco.

La procedura di valutazione del rischio prende in ingresso il PrivacyDataFrame e l'attacco che intendo simulare: in particolare l'identificatore del tipo di attacco, la dimensione del background knowledge e il metodo di calcolo del rischio. Un'astrazione generale dell'esecuzione di un generico calcolo può essere schematizzata come segue:

- *Costruzione del PrivacyDataFrame:* a partire da un dataset generico costruisco un data frame contenente tutti i dati sequenziali degli individui. E' la macrostruttura principale di aggregazione sul quale è possibile applicare ogni procedura di calcolo del rischio, indipendentemente dal tipo di attacco e dal metodo di calcolo.

- *Costruzione dell'attacco:* a partire dall'identificatore dell'attacco e dalla dimensione della background knowledge costruisco l'istanza dell'attacco che intendo simulare.

- *Valutazione del Rischio:* la procedura esterna principale che prende in ingresso il PrivacyDataFrame, l'istanza dell'attacco e il metodo di calcolo del rischio e restituisce un vettore di rischio per ogni individuo. In funzione del tipo di attacco può utilizzare ulteriori parametri necessari al calcolo. Tale procedura utilizza a sua volta tre metodi a cascata, necessari per la modularizzazione del calcolo per ogni utente, la generazione delle configurazioni del background knowledge ed il matching delle istanze con tutti gli individui del dataframe.

4.2 PrivacyDataFrame: Analisi e Struttura di un Dataframe Generico

Un PrivacyDataFrame è un Pandas.DataFrame composto da cinque attributi, i quali possono generalizzare un dataset arbitrario.

L'istanza costruita è sempre composta da una quintupla di attributi, i quali generalizzano il dataset ricevuto in ingresso. Ogni record rappresenterà quindi un generico elemento ed i suoi attributi. Essi sono

- *user_id:* attributo che rappresenta l'identificatore di un certo individuo all'interno del data frame.

- *datetime:* contiene il timestamp dell'istante di rilevazione dell'elemento, con un livello di misurazione al decimo di secondo. Tale attributo definisce la sequenzialità dei dati ricevuti nel tempo. Il dataframe verrà ordinato in funzione di questo attributo. Qualora non fosse presente, la sequenzialità delle informazioni viene garantita dall'attributo order.

- *sequence_id*: attributo che rappresenta l'identificatore della sequenza. E' un attributo con un grado di generalizzazione, infatti a seconda del tipo di elemento che intendo generalizzare assumerà una semantica differente. Ad esempio, se ho un dataset di mobilità la sequenza potrà identificarmi una traiettoria mentre se il dataset contenesse informazioni di retail potremmo interpretarlo come se rappresentasse l'identificatore della ricevuta (o del basket).

- *order:* attributo con il quale si preserva la sequenzialità dei dati all'interno di una sequenza. Infatti, qualora venisse omesso il datetime o aggregassimo un cluster di

elementi aventi un timestamp meno preciso (i.e., una ricerca con timestamp su base oraria o giornaliera) senza questo attributo perderemmo l'ordine delle informazioni.

- *elements:* un elemento di tipo arbitrario, implementato tramite una tupla la cui dimensione e il cui tipo dipendono dalle informazione del dataset che intendo caratterizzare. Ad esempio, se il dataset di retail contiene oltre al nome del prodotto, anche il prezzo ed una breve descrizione allora un generico elements sarà una tupla $e_p =< product\_id, price, overview >$.

Un'istanza di PrivacyDataFrame è un'estensione della classe Pandas.DataFrame, che oltre a ridefinire il costruttore, implementa *properties* e metodi di istanza, sia pubblici che privati. Posso costruire un PrivacyDataFrame invocando il costruttore:

```
def __init__(self, data, user_id = k.USER_ID, datetime = k.DATETIME,
             order = k.ORDER_ID, sequence_id = k.SEQUENCE_ID,
             elements = None, timestamp = False):
```

Descriviamo brevemente i parametri richiesti:

- *data:* il dataset da importare. Può essere a sua volta un pandas.DataFrame, una lista, un dizionario o un file csv.
- *user_id:* nome dell'attributo che in data rappresenta l'identificatore dell'utente. E' opzionale, di tipo int o str. Di default assume il valore uid.
- *datetime:* nome dell'attributo che in data rappresenta il timestamp del rilevamento. E' opzionale, è di tipo int o str. Di default vale datetime.
- sequence_id: nome dell'attributo che in data rappresenta l'identificatore della sequenza. E' opzionale, è di tipo int o str. Di default vale sequence_id.
- *order_id:* nome dell'attributo che in data preserva la sequenzialità dei dei dati in una sequenza. E' opzionale, è di tipo int o str. Di default vale order_id.
- *elements:* è una dizionario le cui chiavi rappresentano i nomi degli attributi che caratterizzano il dataset e che vengono raggruppati in una tupla, mentre i valori indicano il tipo. E' opzionale, di default vale None.
- *timestamp:* flag opzionale con cui eseguo il casting dei valori di datetime, se vale True. Di default vale False.

Il costruttore effettua successivamente ulteriori controlli riguardo la consistenza dei dati inseriti: in particolare verifica che tutti gli attributi siano presenti e che le associazioni siano corrette. Inoltre controlla che tutti gli attributi dell'elemento generico abbiano associato un tipo e che il datetime sia coerente con la sequenzialità dei dati. Nel caso in cui, anche un solo confronto ritorna esito negativo l'PrivacyDataFrame non viene costruito, informando l'utilizzatore del tipo di errore.

4.3 Generalità di un Attacco

In questa sezione descriviamo in termini generali le classi di attacchi sviluppate e i parametri necessari a creare ed eseguire un attacco, con particolare attenzione ai metodi di valutazione del rischio di privacy.

4.3.1 Il file Attacks.py

In generale, un attacco è definito da un insieme di procedure condivise e da un insieme di metodi che lo caratterizzano. Ognuno contempla tre diverse modalità di calcolo del rischio modellate tramite una funzione che genera diversi tipi di background.

Il file Attacks.py è composto da sette classi di attacchi. La classe Attack definisce un attacco generico, implementando i metodi condivisi che definiscono le funzioni necessarie alla valutazione del rischio all'interno di un PrivacyDataFrame ed alla generazione delle istanze. Le classi rimanenti, rappresentano tipologie differenti di attacchi, implementando le procedure astratte ereditate da Attack, le quali caratterizzano il tipo di attacco sviluppato.

4.3.2 Creazione ed Esecuzione di un Attacco

La creazione e l'esecuzione di un attacco dipende fondamentalmente da tre parametri, essi sono:

- *PrivacyDataFrame:* la struttura generale costruita sulla base del dataset ricevuto in ingresso. Tale struttura è stata descritta precedentemente.
- *Lunghezza della Conoscenza:* la quantità di informazioni potenzialmente conosciute da un attaccante. Ad esempio, l'aggressore è a conoscenza che un certo individuo ha acquistato durante una spesa tre determinati prodotti. Di conseguenza il responsabile del dataset genererà tutte le configurazioni di tre record a partire dall'insieme di tutti i record di quella sequenza, che identifica quella spesa.

- *Metodo di Calcolo del Rischio:* l'approccio con cui intendo valutare il rischio di privacy.

A seconda del tipo di attacco, possono essere utilizzati ulteriori parametri come ad esempio il margine di tolleranza o la precisione.

4.3.3 Metodi di Calcolo del Rischio

Sulla base delle informazioni studiate nella fase di identificazione di un attacco, distinguiamo tre differenti metodi di valutazione del rischio di privacy. Esse sono:

- *Conoscenza basata sugli elementi:* le informazioni conosciute dall'attaccante riguardano la macrostruttura composta da tutti i record di un individuo. Il background knowledge viene costruito a partire da questa macrostruttura. Successivamente, in base al tipo di attacco, utilizziamo una nuova struttura definita come aggregazione della precedente, contenente le informazioni che caratterizzano tale attacco.

- *Conoscenza basata sulla sequenza:* le informazioni conosciute dall'attaccante riguardano la macrostruttura composta da tutti i record di una determinata sequenza di un certo individuo (i.e., come nell'esempio immediatamente precedente). Il background knowledge di una sequenza di un utente sarà, pertanto, generato a partire dalla rispettiva macrostruttura, mentre il background knowledge di un individuo è rappresentato dall'unione dei background knowledge di ogni sua sequenza dello stesso individuo. Successivamente, in base al tipo di attacco, utilizziamo una nuova struttura definita come aggregazione della precedente, contenente le informazioni che caratterizzano tale attacco.

- *Conoscenza dell'intera sequenza:* le informazioni conosciute dall'attaccante riguardano intere sequenze. A differenza degli altri metodi di calcolo, è un'informazione molto potente. Ad esempio, possiamo immaginare che l'aggressore abbia pedinato un certo individuo per un periodo di tempo, conoscendo esattamente tutte le tappe di una o più traiettorie. Il background knowledge, in questo caso, è generato a partire dall'indice di tutte le sequenze di un certo individuo, considerando per ognuna l'intero contenuto, filtrando le informazioni in base al tipo di attacco.

4.4 Struttura di un Attacco

La classe Attack classe mette a disposizione le procedure principali per il calcolo del rischio di privacy in un PrivacyDataFrame di individui, delegando la caratterizzazione di ogni

tipo di attacco alla rispettiva classe, la cui funzione è ridefinire soltanto la procedura di matching.

La classe è composta da un costruttore e dall'implementazione delle seguenti procedure:

- *_all_risks:* applica la funzione di rischio su ogni individuo dell'PrivacyDataFrame, restituendo un vettore di rischio.

- *_privacy_risk:* valuta il rischio di privacy di un generico individuo dell'PrivacyDataFrame, calcolandone tutte le probabilità di re-identificazione. In particolare tale funzione, in base al metodo di calcolo del rischio, genera prima le istanze del background knowledge e successivamente calcola il rischio, quantificando tutte le probabilità di re-identificazione.

- *_background_generator:* genera le istanze del background knowledge in funzione del metodo di calcolo del rischio scelto dall'utente.

Come possiamo immaginare, quindi, la _all_risk ha il compito di invocare la _privacy_risk su ogni individuo. Quest'ultima, a sua volta, prima di iniziare il calcolo del rischio genera il background knowledge chiamando la funzione _background_generator. Queste procedure rappresentano lo scheletro di ogni tipologia di attacco che andiamo a sviluppare.

Inoltre, la classe dichiara anche le seguenti procedure astratte, implementate da ogni tipologia di attacco.

- *evaluate_risk:* funzione di pre-setting la quale ha il compito di configurare l'PrivacyDataFrame prima di invocare la procedura di calcolo del rischio totale

- *_matching:* rappresenta il cuore di ogni attacco. Effettua il confronto di una generica istanza rispetto ad un insieme di record. Il tipo di confronto è dettato dalla tipologia di attacco per cui è sviluppata ed al metodo di calcolo.

- *full_sequence_matching:* funzione di matching utilizzata in un approccio di calcolo del rischio basato sulla conoscenza di intere sequenze. Il tipo del confronto dipende dal tipo dell'attacco per cui è implementata.

Successivamente descriviamo con maggiore dettaglio l'implementazione delle procedure non astratte, rimandando la trattazione di quest'ultime quando descriviamo le tipologie degli attacchi sviluppati.

4.4.1 Costruzione di un Generico Attacco

Un attacco generico non è altro che una classe composta da un'unica variabile di istanza che rappresenta la lunghezza di ogni singolo elemento del background knowledge, ovvero la quantità di informazioni con cui un malintenzionato può re-identificare un certo individuo.

```
def __init__(self, knowledge_length)
```

Il parametro con cui costruisco l'istanza di un attacco deve essere un intero positivo, altrimenti l'oggetto non viene costruito e l'utente viene opportunamente avvisato dell'errore.

4.4.2 Generazione del Background Knowledge

Il background knowledge, come precedentemente descritto, rappresenta l'insieme di tutti i possibili modi con cui un attaccante può violare la privacy di un certo individuo.

La procedura che costruisce l'insieme di tutte le conoscenze è definita come segue:

```
def _background_generator(self, single_priv_df, method)
```

Dove, single_priv_df è la macrostruttura rappresentante il dataframe di record dal quale generare tutte le possibili combinazioni di dimensione knowledge_length. Tale macrostruttura è un PrivacyDataFrame. La struttura contenente tutte le possibili combinazioni di una data dimensione la ottengo utilizzando la procedura itertools.combinations. La dimensione del frame varia in base al metodo di calcolo del rischio, specificato nel parametro method. Ad esempio se calcolo il rischio con una conoscenza basata sugli elementi, la macrostruttura è costituita da tutti i record di un certo individuo.

Vediamo in dettaglio come distinguere caso per caso:

Generazione del Background Knowledge con un approccio Elements Based Knowledge

Come possiamo vedere, calcolando il rischio con una conoscenza basata su tutti gli elementi di un individuo, single_priv_df identifica la macrostruttura contenente tutti i record di un certo individuo dal quale andrò a generare tutte le combinazioni di knowledge_length record.

```
# method == k.ELEMENTS_BASED_KNOWLEDGE:
size = len(single_priv_df)
if self.knowledge_length > size:
```

```
    cases = combinations(single_priv_df.values, size)
else:
    cases = combinations(single_priv_df.values, self.knowledge_length)
```

Generazione del Background Knowledge con un approccio Sequence Based Knowledge

In questo caso, calcolando il rischio con una conoscienza basata sulla sequenza di un individuo, single_priv_df identifica il dataframe contenente tutti i record della sequenza di quell'utente dal quale andrò a generare combinazioni di knowledge_length record. Adesso che abbiamo il background knowledge di una sequenza dell'individuo, reitero la procedura combinatoria per ogni sequenza dello stesso, ottenendo il background knowledge dell'individuo. In pratica non è altro che l'unione disgiunta di tutte le possibili conoscenze di ogni sequenza dello stesso individuo.

```
# method == k.SEQUENCE_BASED_KNOWLEDGE:
cases_list = single_priv_df.groupby(k.SEQUENCE_ID).apply(lambda x:
             self._background_generator(x,k.ELEMENTS_BASED_KNOWLEDGE))
cases = []
for cases4seq in cases_list:
    for case4seq in cases4seq:
        cases.append(case4seq)
```

La reiterazione della stessa procedura, come possiamo leggere dal codice, è stata effettuata raggruppando tutte le sequenze dell'utente su cui abbiamo applicato la stessa funzione di generazione. Successivamente tutte le istanze sono state inserite in una lista.

Generazione del Background Knowledge con un approccio Full Sequence Knowledge

La generazione del background knowledge in un contesto in cui la conoscenza è relativa all'intera sequenza, è leggermente differente. La macrostruttura single_priv_df è un PrivacyDataFrame contenente tutti i record di un generico individuo. In questo caso, in primo luogo considero l'identificatore unico di ogni sequenza e genero tutte le possibili combinazioni di knowledge_length identificatori. In pratica, ottengo un background knowledge le cui istanze contengono gli identificatori delle sequenze dell'individuo.

```
# method == k.FULL_SEQUENCE_KNOWLEDGE:
unique_seqs = single_priv_df[k.SEQUENCE_ID].unique()
size = len(unique_seqs)
```

```
if self.knowledge_length > size:
    cases = combinations(unique_seqs,size)
else:
    cases = combinations(unique_seqs, self.knowledge_length)
seqs_values = []
for case in cases:
    goodseqs = list(case)
    case = single_priv_df[single_priv_df[k.SEQUENCE_ID].isin(goodseqs)]
    seqs_values.append(case)
```

Successivamente, per ogni istanza andrò a sostituire ogni identificatore di sequenza con il rispettivo insieme di record di cui è composta. Alla fine ottengo una lista nel quale ogni elemento contiene tutti i record di knowledge_length sequenze.

4.4.3 Valutazione del Rischio di Privacy Individuale

La procedura che descriviamo permette di calcolare il rischio di privacy individuale, valutando per ogni istanza del background knowledge la probabilità di re-identificazione dello stesso individuo. La procedura di calcolo del rischio utilizza i seguenti parametri:

```
def _privacy_risk(self, single_priv_df, priv_df,
                  method=k.ELEMENTS_BASED_KNOWLEDGE):
```

La macrostruttura single_priv_df è il dataframe su cui invocare la procedura per la generazione del background knowledge. Come al solito il numero di record dipende dal metodo di calcolo del rischio: può contenere tutti i record di un utente, oppure quelli relativi ad una sequenza. La struttura priv_df è l'PrivacyDataFrame di tutta la popolazione di individui. Questo parametro è necessario per calcolare la probabilità di re-identificazione di un certo individuo in funzione di tutta la popolazione. Il parametro method definisce il tipo di rischio che andrò a calcolare. Vediamo in dettaglio gli approcci utilizzati per calcolare i tre tipi di rischio.

Calcolo del Rischio di Privacy con una approccio Elements Based Knowledge.

Ricordiamo che il rischio di privacy è pari alla massima probabilità di re-identificazione, ognuna delle quali è valutata, in questo caso, confrontando una generica istanza con ogni macrostruttura composta da tutti i record che identificano un individuo.

```
# method == k.ELEMENTS_BASED_KNOWLEDGE:
cases = self._background_generator(single_priv_df,
                                   k.ELEMENTS_BASED_KNOWLEDGE)
privacy_risk = 0
for case in cases:
    case_risk = 1.0 / priv_df.groupby(k.USER_ID).apply(lambda x:
                      self._matching(x, case)).sum()
    if case_risk > privacy_risk:
        privacy_risk = case_risk
    if privacy_risk == 1:
        break
return privacy_risk
```

Calcolando il rischio con una conoscenza basata sugli elementi, il single_priv_df contiene tutti i record relativi allo stesso individuo. La prima operazione da effettuare è la generazione del background knowledge e poi calcolare le probabilità di re-identificazione. Andrò, quindi, ad eseguire il matching di ogni istanza rispetto a tutti i record di ogni individuo, il numero di confronti positivi mi restituisce il denominatore del rapporto di rischio. Si noti che il numeratore vale sempre 1,questo perchè ogni istanza del background knowledge è stata generata a partire da tutti i record di quell'individuo di conseguenza il confronto sarà sempre positivo.

Appena viene calcolata una probabilità di re-identificazione pari ad 1, la procedura termina in quanto abbiamo già il massimo rischio. Altrimenti continua a matchare l'istanza con ogni dataframe di ogni individuo tenendo traccia della probabilità più alta. Il tipo di matching si basa sul tipo di attacco utilizzato.

Calcolo del Rischio di Privacy con un approccio Sequence Based Knowledge

Ricordiamo che il rischio di privacy è pari alla massima probabilità di re-identificazione, ognuna delle quali è valutata, in questo caso, confrontando una generica istanza con ogni macrostruttura composta da tutti i record di una sequenza di un individuo. Può accadere in questo caso, che l'individuo sotto attacco, possa contenere più macrostrutture che matchano con la stessa istanza. A differenza del primo metodo di calcolo, il numeratore del rapporto di rischio può essere compreso tra 1 ed il massimo numero di sequenze dell'individuo.

```
# method == k.SEQUENCE_BASED_KNOWLEDGE:
cases = self._background_generator(single_priv_df,
                                   k.SEQUENCE_BASED_KNOWLEDGE)
```

```
privacy_risk = 0
for case in cases:
    num = single_priv_df.groupby([k.SEQUENCE_ID]).apply(lambda x:
          self._matching(x, case)).sum()
    den = priv_df.groupby([k.USER_ID, k.SEQUENCE_ID]).apply(lambda x:
          self._matching(x, case)).sum()
    case_risk = num / den
    if case_risk > privacy_risk:
        privacy_risk = case_risk
    if privacy_risk == 1:
        break
return privacy_risk
```

Calcolando il rischio con una conoscenza basata sulle sequenze, la macrostruttura single_priv_df contiene tutti i record relativi ad una generica sequenza di un individuo. Dopo aver creato tutte le istanze, vengono calcolate tutte le probabilità di re-identificazione per ogni elemento del backgournd knowledge. Successivamente si effettuerà il matching di ogni istanza con tutte le sequenze dell'individuo sotto attacco, il numero di confronti con esito positivo mi restituirà il numeratore del rapporto di rischio. Il denominatore, lo ottengo contando il numero di matching positivi della stessa istanza con rispetto a ogni sequenza di ogni individuo.

Per garantirne l'efficienza, appena trovo una probabilità pari ad 1, la procedura termina. Il tipo del confronto dipende dal tipo dell'attacco che utilizzo.

Calcolo del Rischio di Privacy con un approccio Full Sequence Knowledge

In questo caso ricordiamo che ogni istanza è composta dal contenuto, quindi da tutti i record, di una o più sequenze. Ogni probabilità di re-identificazione la ottengo contando il numero di individui che hanno le sequenze con esattamente lo stesso contenuto. Il rischio di privacy è quantificato dalla massima probabilità.

```
# method == k.FULL_SEQUENCE_BACKGROUND:
cases = self._background_generator(single_priv_df,
                                   k.FULL_SEQUENCE_KNOWLEDGE)
privacy_risk = 0
for case in cases:
    case_risk = 1.0 / priv_df.groupby(k.USER_ID).apply(lambda x:
                      self._full_elements_match(x, case)).sum()
    if case_risk > privacy_risk:
```

```
            privacy_risk = case_risk
        if privacy_risk == 1:
            break
    return privacy_risk
```

Il dataframe single_priv_df contiene tutti i record dell'individuo sotto attacco. Dopo aver generato tutte le istanze effettuo il confronto con il contenuto delle sequenze di ogni individuo. Tale operazione è delegata ad una differente funzione di matching, implementata con una procedura diversa proprio per differenziare questo metodo di calcolo del rischio. A differenza dei precedenti, si matchano clusters di sequenze di record, ovvero intere sequenze tenendo conto dell'ordine.

4.4.4 Valutazione del Rischio di Privacy Totale

La funzione di calcolo del rischio totale ha il compito di estendere il calcolo del rischio di privacy individuale su tutta la popolazione dell'ElementDataFrame.

```
def _all_risks(self,priv_df, uids=None,method=k.ELEMENTS_BASED_KNOWLEDGE,
previous_risk=previous_risk):
```

Tale procedura possiede come parametri il dataframe di tutta la popolazione, priv_df ed la lista uids la quale mi permette di calcolare il rischio di privacy ad insieme selettivo di individui, quindi non necessariamente tutti.

```
if uids == None:
    uids = priv_df
else:
    if isinstance(uids, list):
        uids = priv_df[priv_df[k.USER_ID].isin(uids)]
    if isinstance(uids,PrivacyDataFrame) or isinstance(uids,pd.DataFrame):
        uids = priv_df[priv_df[k.USER_ID].isin(uids[k.USER_ID])]

risks = uids.groupby(k.USER_ID).apply(lambda x: self._privacy_risk
        (x, priv_df, method=method)).reset_index(name=k.PRIVACY_RISK)

return risks
```

Come possiamo leggere, la procedura non fa altro che delegare il calcolo di rischio individuale, con il rispettivo metodo, ad ogni individuo della popolazione. Il parametro

previous_risk mi permette di migliorare l'efficienza della procedura. In pratica se con una lunghezza della conoscenza inferiore il rischio valeva precedentemente 1, allora in questo caso l'utente con rischio precedentemente massimo non viene preso in considerazione nel calcolo, bensì soltanto riportato nel nuovo vettore di rischio. Il metodo termina restituendo un vettore di rischi associato ad ogni utente.

4.4.5 Tipi di Attacchi: Analisi e Struttura

In questa sezione descriviamo la struttura di un generico tipo di attacco, senza ribadirne le funzionalità già descritte nel capitolo delle definizioni. Le tipologie di attacchi rappresentano i potenziali tipi di informazioni che un malintenzionato può conoscere per re-identificare un certo individuo. Durante questo progetto abbiamo sviluppato sei differenti tipi di attacchi, ognuno dei quali con tre differenti metodi di calcolo del rischio. Distinguiamoli senza far riferimento al metodo di calcolo, bensì alla differenti informazioni conosciute da un potenziale attaccante. Essi sono:

- *ElementsAttack*: conoscenza di un sottoinsieme di elementi.
- *Elements Sequence Attack*: l'attaccante è al corrente di una sottosequenza di elementi.
- *ElementsTimeAttack*: conoscenza di un sottoinsieme di coppie: elemento e timestamp.
- *ElementsFrequenceAttack*: l'aggressore conosce la frequenza di un sottoinsieme di elementi
- *ElementsProbabilityAttack*: conoscenza della probabilità di un sottoinsieme di elementi
- *ElementsProportionAttack*: l'aggressore conosce le proporzioni di un sottoinsieme di elementi rispetto a quello di maggiore frequenza.

Ogni tipo di attacco è una classe che eredita da Attack. La sua caratterizzazione è definita nell'implementazione delle procedure astratte dichiarate nella superclasse, esse sono:

- *_matching*
- *_full_elements_matching*
- *evaluate_risk*

Le funzioni di matching definiscono il tipo di un particolare attacco. La loro definizione è la seguente:

```
def _matching(self, single_priv_df, instance):
```

```
def _full_elements_match(self, single_priv_df, case):
```

In entrambi i casi, l'obiettivo è quello di verificare se il confronto dell'istanza con la struttura ritorna esito positivo o negativo. La tipologia del confronto dipende dallo scopo di quel tipo di attacco. La differenza tra le due funzione di matching dipende soltanto dal metodo di calcolo di calcolo del rischio che viene utilizzato, la prima la utilizzo nel caso in cui voglia confrontare l'istanza con un insieme di record, indipendentemente se rappresentano una sequenza o individuo. Tale distinzione viene fatta nella procedura di calcolo del rischio individuale. La seconda procedura, invece, la utilizzo in un contesto full sequence knowledge, dove devo confrontare l'esatto contenuto di intere sequenze. In alcune tipologie di attacco la seconda procedura utilizza a sua volta una funzione lambda di appoggio come supplemento al confronto.

In generale, una volta creata un istanza di un attacco. La procedura di avvio per il calcolo del rischio totale è:

```
def evaluate_risk(self, privacy_df, uids=None,
                  method=k.ELEMENTS_BASED_KNOWLEDGE, previous_risk=NoNe):
```

Tale funzione viene utilizzata per avviare la cascata di procedure definite nella classe Attack. E' una funzione molto breve, il cui ruolo è quello di preparare l'PrivacyDataFrame alle procedure di calcolo. In particolare, tale procedura invoca la procedura di calcolo del rischio totale, la quale a sua volta calcola il rischio di privacy di ogni individuo, generando il rispettivo background knowledge. Il parametro previous_risk fa si che non venga ricalcolato il rischio di individui il cui rischio precedente era già massimo. Tale decisione va ad impattare drasticamente sulle performance, ottenendo un miglioramento significativo in termini di tempo di calcolo.

Un Esempio: FrequencyAttack

Consideriamo un attacco nel quale l'aggressore conosce un sottoinsieme finito elementi (i.e., coordinate, posizioni) e la frequenza con i qualivengono visitati. Tali conoscenze sono relative ad una traiettoria effettuata in un certo periodo. Supponiamo che il responsabile di un provider voglia calcolare il rischio di privacy di un'intera popolazione, sulla base di un ipotetico aggressore che conosce il luogo e il numero di volte che è stato visitato.

Per prima cosa, viene creata un'istanza di FrequencyAttack, supponendo che si conoscano due coppie di informazioni. Successivamente avviamo il calcolo del rischio, passandogli il PrivacyDataFrame precedentemente creato ed il metodo di calcolo del rischio, nel quale supponiamo che le informazioni che conosce siano relative ad una certa traiettoria.

```
# Import data
data_mob = from_file(\../data/mobility.csv")
df_mob = PrivacyDataFrame(data_mob,user_id='user', sequence_id="seq",
                          order='order', elements={'pname':str})

# Make ElementsAttack instance then compute risk
at = attacks.ElementsAttack(knowledge_length=2)
risk = at.evaluate_risk(d,method=k.SEQUENCE_BASED_KNOWLEDGE)
```

Questa è la procedura con la quale calcoliamo i rischi dell'intera popolazione. Vediamo in questo caso come è stata implementata la funzione di valutazione del rischio:

```
def evaluate_risk(self,privacy_df,uids=None,
                  method=k.ELEMENTS_BASED_KNOWLEDGE,previous_risk=NoNe):
    freq = frequency_data(privacy_df, method=method)
    return self._all_risks(freq, uids=uids, method=method,
                           previous_risk=previous_risk)
```

Come sappiamo, in un attacco basato sulla frequenza, il rischio deve essere calcolato sul vettore di frequenza relativo ad un certo dataset. La procedura, infatti, converte il PrivacyDataFrame in un vettore di frequenza. Successivamente avvia il metodo di valutazione del rischio totale:

```
risks = uids.groupby(k.USER_ID).apply(lambda x: self._privacy_risk
        (x, priv_df, method=method)).reset_index(name=k.PRIVACY_RISK)
```

la quale calcola per ogni individuo il rispettivo rischio di privacy. Il metodo prima deve generare il background knowledge dell'individuo e solo successivamente inizia ed effettuare il matching per calcolare il rischio.

```
# method == k.SEQUENCE_BASED_KNOWLEDGE:
cases_list = single_priv_df.groupby(k.SEQUENCE_ID).apply(lambda x:
             self._background_generator(x,k.ELEMENTS_BASED_KNOWLEDGE))
cases = []
```

```
for cases4seq in cases_list:
    for case4seq in cases4seq:
        cases.append(case4seq)



cases = self._background_generator(single_priv_df,
                                   k.SEQUENCE_BASED_KNOWLEDGE)

privacy_risk = 0
for case in cases:
    num = single_priv_df.groupby([k.SEQUENCE_ID]).apply(lambda x:
                                self._matching(x, case)).sum()
    den = priv_df.groupby([k.USER_ID, k.SEQUENCE_ID]).apply(lambda x:
                           self._matching(x, case)).sum()
    case_risk = num / den

    if case_risk > privacy_risk:
        privacy_risk = case_risk
    if privacy_risk == 1:
        break
```

La matching in questo preciso caso confronterà la coppia di informazioni: punto e frequenza, con ogni sequenza. La matching è:

```
def _matching(self, single_priv_df, case):

    occ = pd.DataFrame(data=case, columns=single_priv_df.columns)
    occ.rename(columns={k.FREQUENCY: k.FREQUENCY + "case"}, inplace=True)
    merged = pd.merge(single_priv_df, occ, left_on=[k.ELEMENTS],
                      right_on=[k.ELEMENTS])

    if len(merged.index) != len(occ.index):
        return 0
    else:
        cond1 = merged[k.FREQUENCY + "case"] >= merged[k.FREQUENCY] -
                merged[k.FREQUENCY] * self.error
        cond2 = merged[k.FREQUENCY + "case"] <= merged[k.FREQUENCY] +
```

```
                merged[k.FREQUENCY] * self.error

        if len(merged[cond1 & cond2].index) != len(occ.index):
            return 0
        else:
            return 1
```

Nel caso in cui il responsabile voglia calcolare il rischio di privacy, considerando la conoscenza di due intere sequenze, quindi ogni punto ed ogni sequenza per ognuna, il procedimento fino alla chiamata della _privacy_risk è lo stesso se non che quest'ultima in questo caso esegue la _full_elements_match. Quest'ultima, come possiamo leggere contiene nella sua istanza il dataframe delle due intere sequenze mentre single_priv_df contiene il dataframe di tutti i record di un individuo. La procedura cerca in ogni sequenza dell'individuo l'esatta corrispondenza di entrambe le sequenze dell'istanza. Per far ciò si appoggia ad una lambda function.

```
def _full_elements_match(self, single_priv_df, case):

    ordered_single_priv = single_priv_df.sort_values(by=[k.SEQUENCE_ID,
                          k.ELEMENTS])
    ordered_case = case.sort_values(by=[k.SEQUENCE_ID, k.ELEMENTS])
    case_seqs = ordered_case.groupby(k.SEQUENCE_ID)
    for _, case_seq in case_seqs:
        match = ordered_single_priv.groupby(k.SEQUENCE_ID).apply
                (lambda x: self._lambda_full_freq_match(x,case_seq))
        if not match.any():
            return 0
    return 1
```

dove la funzione lambda di supporto effettua il seguente controllo:

```
def _lambda_full_freq_match(self, single_priv, case):
    cond1 = list(case[k.ELEMENTS]) == list(single_priv[k.ELEMENTS])
    cond2,cond3=False,False
    if cond1:
        cond2 = full_list_compare(list(case[k.FREQUENCY]),
                list(single_priv[k.FREQUENCY] -
                (single_priv[k.FREQUENCY] * self.error)))
        cond3 = full_list_compare(list(single_priv[k.FREQUENCY] +
```

```
                (single_priv[k.FREQUENCY] * self.error)),
                list(case[k.FREQUENCY]))
    return cond1 and cond2 and cond3
```

Capitolo 5

Testing sperimentale

Abbiamo utilizzato PRIVSeq per valutare i rischi su un insieme di dati di retail in larga scala. In particolare, abbiamo analizzato le tracce degli acquisti di migliaia di customer, la maggior parte di cui grossisti, relativamente ad un web store localizzato in UK. In questo capitolo descriviamo il tipo di dato e successivamente utilizziamo la libreria per calcolare i rischi e plottare la distribuzione.

5.1 Tipo di Dato

Il dataset che abbiamo analizzato riguarda tutte le transazioni relative alla vendita al dettaglio di prodotti ad uso personale. L'esercente è un web store con sede nel Regno Unito, il cui insieme di dati fornito prende in considerazione tutti gli acquisti a partire dal 01/12/2009 al 09/12/2011. La maggior parte dei prodotti venduti riguarda articoli da regalo per tutte le occasioni, i cui principali acquirenti sono grossisti. Infatti, la maggior parte dei basket, come vedremo, prende in considerazione sequenze di considerevole dimensione poco adatte a finalità private. Ogni record del dataset è composto da otto attributi, vediamone la descrizione:

- *InvoiceNo*: è l'invoice number, ovvero l'identificatore della ricevuta. Un numero intero di sei cifre che identifica in modo univoco la transazione. Qualora tale valore iniziasse con la lettera 'c', la ricevuta indicherebbe uno storno.

- *StockCode*: è il codice del prodotto. Un numero intero di cinque cifre che identifica univocamente il prodotto acquistato.

- *Description*: contiene una stringa di lunghezza arbitraria contenente il nome del prodotto.

- *Quantity*: numero intero che indica la quantità acquistata di quel determinato prodotto.
- *InvoiceData*: è una stringa contenente la data e l'ora nel quale è stata effettuata la transazione relativa al prodotto.
- *UnitPrice*: il prezzo unitario del prodotto, rappresentato da un numero in virgola mobile.
- *CustomerID*: l'identificatore univoco dell'acquirente, rappresentato da un numero intero di cinque cifre.
- *Country*: stringa che indica il paese di residenza dell'acquirente.

Il seguente esempio mostra la composizione del dataset dopo averlo importato in un generico dataframe per alleggerire la lettura.

	Invoice	StockCode	Description	Quantity	
30	C493430	21527	RETRO SPOT TRADITIONAL TEAPOT	-1	...
31	C493430	85123A	WHITE HANGING HEART T-LIGHT HOLDER	-1	...
32	C493431	21533	RETRO SPOT LARGE MILK JUG	-1	...
33	C493431	22066	LOVE HEART TRINKET POT	-4	...
34	C493431	21432	SET OF 3 CASES WOODLAND DESIGN	-1	...
35	493432	21488	RED WHITE SCARF HOT WATER BOTTLE	1	...
36	493432	84029E	RED WOOLLY HOTTIE WHITE HEART.	3	...
37	493432	21357	TOAST ITS - DINOSAUR	5	...
38	493432	37448	CERAMIC CAKE DESIGN SPOTTED MUG	2	...
39	493432	20726	LUNCH BAG WOODLAND	20	...

	...	InvoiceDate	Price	CustomerID	Country
30	...	2010-01-04 11:43:00	7,95	14680	United Kingdom
31	...	2010-01-04 11:43:00	2,95	14680	United Kingdom
32	...	2010-01-04 11:46:00	4,95	17372	United Kingdom
33	...	2010-01-04 11:46:00	1,45	17372	United Kingdom
34	...	2010-01-04 11:46:00	5,95	17372	United Kingdom
35	...	2010-01-04 12:30:00	3,95	14680	United Kingdom
36	...	2010-01-04 12:30:00	3,75	14680	United Kingdom
37	...	2010-01-04 12:30:00	1,25	14680	United Kingdom

```
38  ...     2010-01-04 12:30:00     1,49        14680   United Kingdom
39  ...     2010-01-04 12:30:00     1,65        14680   United Kingdom
```

5.2 Analisi dei Dati e Scelte Progettuali

Ad una prima analisi potremmo pensare di creare un PrivacyDataFrame che contiene tutti gli attributi. Ad esempio associando i riferimenti principali come nel seguente dizionario $\{'uid' : 'CustomerID','datetime' : 'InvoiceData','sequence' : 'Invoice'\}$ e inserire come elemento generico i restanti attributi. Ma quest'ultimo ragionamento è controintuitivo. Infatti un malintenzionato potrebbe essere un conoscente che è venuto a sapere cosa stato acquistato oppure anche un aggressore che non abbiamo mai visto e che è riuscito ad accedere al sistema vedendo la cronologia degli acquisti sapendo addirittura la cronologia di intere transazioni. É poco probabile che un conoscente, oltre allo StockCode, sappia anche la descrizione, quanti ne sono stati acquistati, il prezzo unitario e il paese di origine dell'acquirente. Noi supporremo che l'attaccante conosca lo StockCode, il quale identifica il prodotto ed indirettamente anche il prezzo e sicuramente la descrizione. I restanti attributi non sono rilevanti per i nostri fini. Avremmo potuto inserirli come attributi supplementari al PrivacyDataFrame, ma per non appesantire ulteriormente la grande quantità di dati abbiamo preferito evitare. L'elemento generico è composto dall'identificatore del prodotto.

Una volta definito come generalizzare il set di dati possiamo pensare a quanti e quali dati analizzare. Scalare la valutazione empirica del rischio di privacy nell'intero periodo di due anni sarebbe impensabile. La complessità computazionale e l'analisi combinatoria di oltre un milione di record non permetterebbe in tempi accettabili nessun tipo di valutazione. In questo caso dovremmo adottare strategie di machine learning, ad esempio utilizzando classificatori che mi permettono di stimare il rischio e non di calcolarlo per ogni istanza. Abbiamo, quindi, deciso di analizzare tutte le transazioni relative al primo semestre del 2010, ovvero dal 01/01/2010 al 30/06/2010. Il dataset analizzato è composto da circa 175.000 record, con un totale di circa 126000 record significativi e tiene traccia delle transazioni effettuate da 2000 customer differenti. Il numero totale delle ricevute emesse è circa 7000. Di conseguenza il numero medio di record associati ad un individuo è di 63. In generale, un cliente è quindi composto da 3.5 transazioni, ognuna delle quali prevede l'acquisto di 18 prodotti nel carrello.

Adesso che abbiamo definito come strutturare i dati e scelto quali transazioni analizzare possiamo importare l'insieme di dati nella libreria. Costruiamo il PrivacyDataFrame,

ovvero la struttura in grado di accoglierli su cui successivamente simulare gli attacchi calcolandone i rischi. La struttura del dataframe è rappresentata nel seguente esempio.

	uid	datetime	sequence	order	elements
12474	2010-06-13	11:19:00	512063	7	22489
12474	2010-05-10	11:58:00	C507486	1	21432
12474	2010-05-10	11:58:00	C507486	2	20703
12474	2010-05-10	11:58:00	C507486	3	22320
12477	2010-03-26	14:01:00	502764	1	48189
12477	2010-03-26	14:01:00	502764	2	48194
12477	2010-03-26	14:01:00	502764	3	48195
12477	2010-03-26	14:01:00	502764	4	84815
12477	2010-03-26	14:01:00	502764	5	22431

In particolare associamo gli attributi generici come segue:

- *uid*: l'identificatore dell'individuo è associato allo *CustomerID*
- *datetime*: il *Pandas.Timestamp* contenente la data e l'ora della transazione è rappresentato dal *InvoiceDate*. In fase di importazione devo convertire l'attributo in istanza di tale classe.
- *order*: la sequenzialità delle transazioni è dettata dall'*InvoiceDate*. Pertanto, tale attributo viene costruito in fase di costruzione del PrivacyDataFrame. In tal caso, in un attacco ElementsTimeAttack con un margine di precisione ampio (i.e., $precision =' Month'$), l'ordine dei prodotti nel cluster generato verrebbe preservato.
- *sequence*: l'attributo generico sequenza, in questo contesto, identifica una successione ordinata di elementi, ovvero prodotti identificati dallo StockCode.
- *elements*: l'elemento è lo StockCode. Possiamo immaginarlo con una tupla di dimensione unitaria. Qualora l'attaccante conoscesse ad esempio anche la quantità acquistata allora avremmo potuto inserire un ulteriore elemento ottenendo una coppia (StockCode,Quantity). Ma in tal caso sarebbe stato superfluo complicare la struttura, in quanto se avesse un'informazione del genere sarebbe sufficente attaccare l'individuo sulla base di una conoscenza della frequenza di acquisto.

Un PrivacyDataFrame è un Pandas.DataFrame, pertanto la costruzione di tale struttura e l'importazione di un generico dataset mantiene pressochè le stesse performance

della superclasse. I controlli ed il casting degli attributi sono operazioni tra Pandas.Series ed anche in questo caso non abbiamo un degrado di prestazioni. Un peggioramento del tempo di costruzione lo otteniamo quando deve essere realizzato l'attributo order perchè non presente nel dataset (proprio come nel dataset in questione). In tal caso avremmo un degrado del 130% circa. Su un numero di record pari a quelli di questo dataset, la costruzione può impiegare 8.5-13 minuti anzichè 4-8 secondi. Tale degrado rimane in un contesto in cui il calcolo del rischio impiega 4-5 giorni ad essere valutato, quindi, in ogni caso, un degrado irrisorio.

Adesso non rimane altro che simulare gli attacchi ed avere valutazioni del rischio sulla base di conoscenze differenti. Per ogni attacco che utilizzeremo, dovremmo decidere il metodo di calcolo, la dimensione della conoscenza e i livelli di tolleranza e di precisione necessari in determinati attacchi. Lo schema seguente mostra gli attacchi e i parametri con i quali simulare potenziali aggressioni e valutare il rischio.

	Elements Based Knowledge	Sequence Based Knowledge	Full Sequence Knowledge	Attacchi Totali
Elements Attack	K=1,2,3,4	K=1,2,3,4	K=1,2,3	13
ElementsSequence Attack	K=1,2,3,4	K=1,2,3,4	K=1,2,3	13
ElementsFrequency Attack	K=1,2,3,4 T=[0.5]	K=1,2,3,4 T=[0.5]	K=1,2,3 T=[0.5]	13
ElementsProbability Attack	K=1,2,3,4 T=[0.5]	K=1,2,3,4 T=[0.5]	K=1,2,3 T=[0.5]	13
ElementsTime Attack	K=1,2,3,4 P=[D]	---	---	4

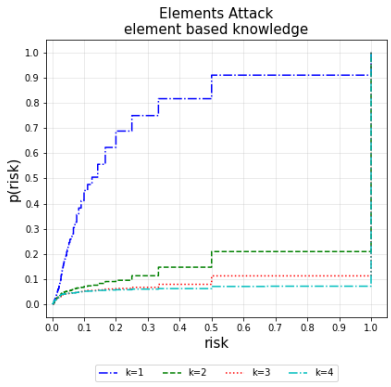
Figura 5.1: Tipi di attacchi eseguiti sul dataset

La simulazione degli attacchi è stata effettuata su un calcolatore macOS avente cpu 3.5 GHz Intel Core i5 e memoria 16 GB 1600 MHz DDR3. Di conseguenza la valutazione delle performance rappresenta una misurazione indicativa con macchine aventi lo stesso tipo di architettura. Ogni attacco è stato eseguito con tutti i metodi di calcolo del rischio. La conoscenza dell'informazione o delle informazioni può riguardare tutti i prodotti acquistati da un cliente, quelli relativi ad una precisa transazione oppure conoscere addirittura intere transazioni di acquisto. Quest'ultimo potrebbe essere il caso di un hacker che ha trovato il

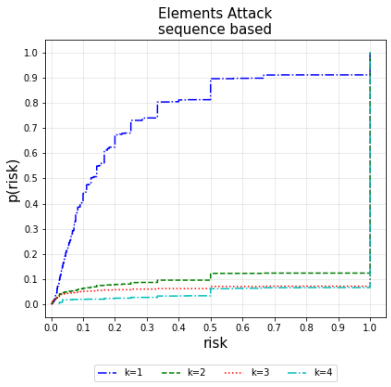
modo di accedere nello store online e vedere la cronologia degli acquisti. Per ogni attacco si considera un range di conoscenza fino ad un massimo k=4. Abbiamo notato che un valore superiore sarebbe poco interessante in quanto porterebbe al valore massimo del rischio per l'85-90% degli individui. Un malintenzionato può sapere cosa è stato acquistato e la data con un certo margine di precisione.

5.3 Risultati e Distribuzione dei Rischi

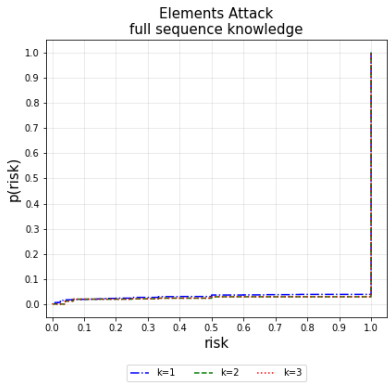
In questa sezione mostriamo i risultati degli esperimenti simulati sui nostri dati sperimentali di retail. Il primo set di risultati mostra l'andamento della funzione di distribuzione cumulativa per ogni attacco al variare di k, distinguendo i tre metodi di calcolo del rischio. Ogni esperimento utilizza un $k = 1, 2, 3, 4$, a parte per gli attacchi simulati con la conoscenza dell'intera sequenza nei quali ci siamo limitati a considerare i primi tre valori. Il secondo set di grafici mostra l'andamento degli attacchi tenendo costante il valore di k.



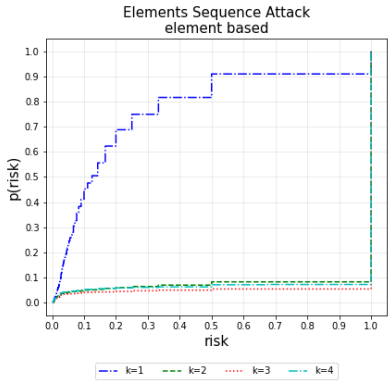
(a)



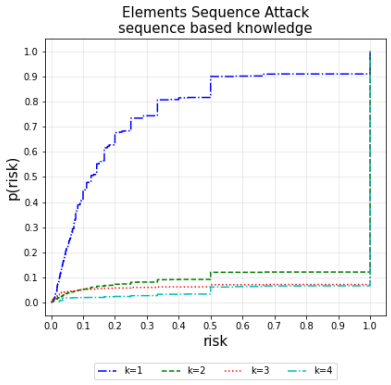
(b)



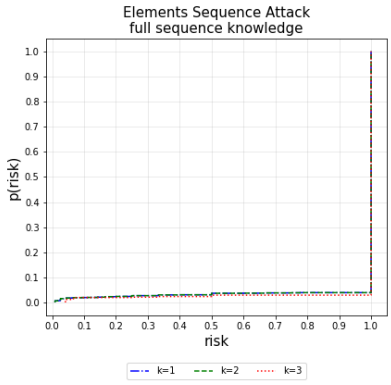
(c)



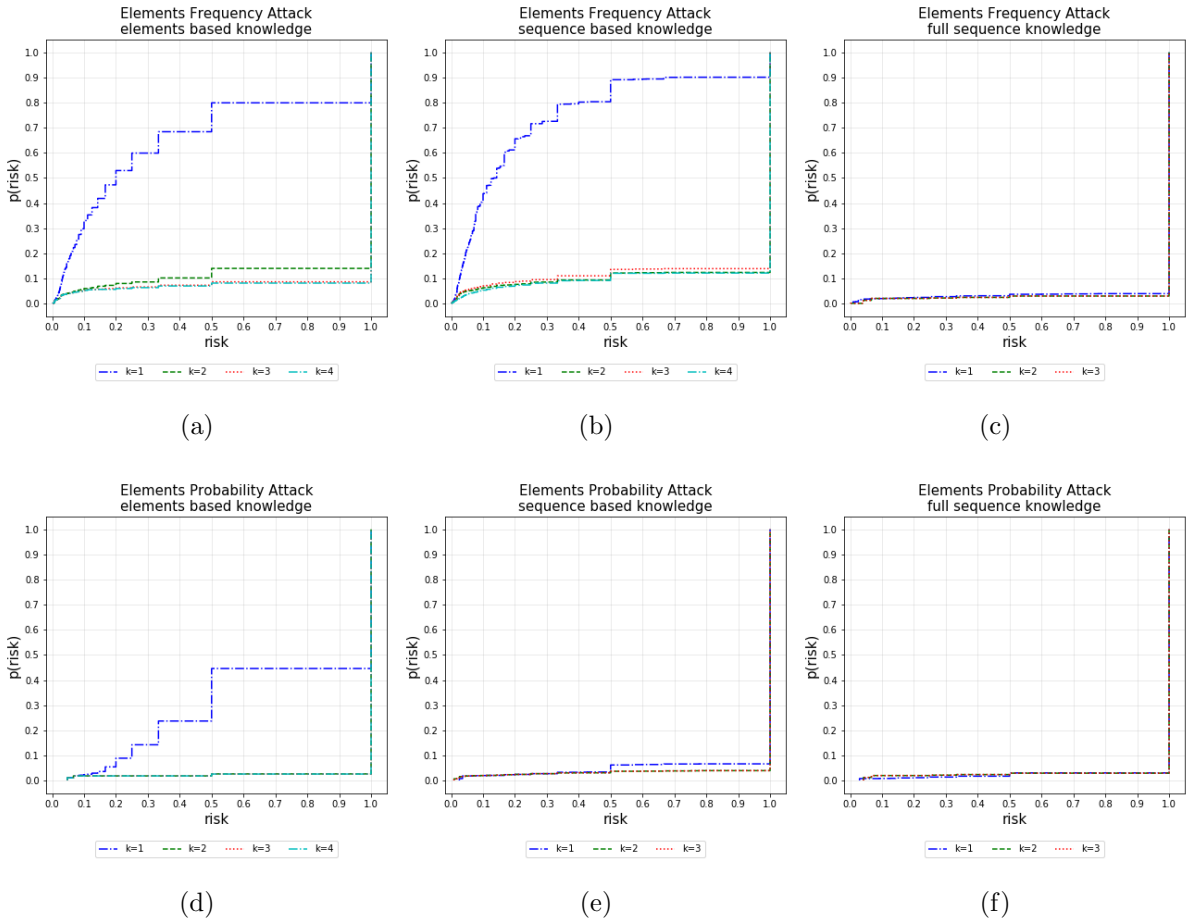
(d)



(e)



(f)



(a) (b) (c)

(d) (e) (f)

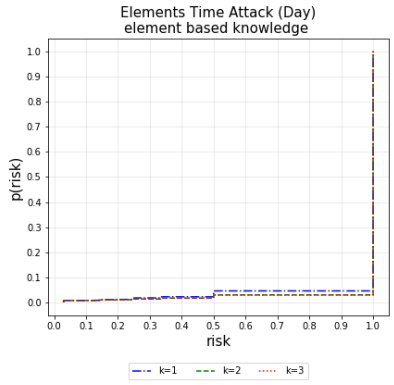
Figura 5.2: Plotting delle funzioni di distribuzione cumulativa del rischio

Ad una prima analisi possiamo notare che, indipendentemente dal tipo di attacco, il metodo di calcolo basato sulla conoscenza dell'intera sequenza non mostra risultati apprezzabili. Infatti, circa il 95% degli individui al variare di k possiede sempre rischio massimo, indipendentemente dal tipo di attacco. Questo risultato dipende dalla distribuzione dell'insieme dei dati. Il dataset utilizzato tiene traccia delle transazioni effettuate nella maggior parte dei casi da grossisti e non da clienti finali. Di conseguenza la maggioranza delle transazioni è composta da lunghe sequenze di acquisti. Pertanto la conoscenza di un intera sequenza di prodotti acquistati è più che sufficiente a re-identificare l'acquirente. Il restante 5% dei clienti con rischio non massimo rappresenta individui occasionali con pochissime transazioni, ognuna delle quali è composta da un numero molto limitato di prodotti acquistati.

É interessante notare il plateau disegnato dalla funzione di distribuzione una volta raggiunto il rischio 0.5, ottenuto da tutti gli attacchi che utilizzano il metodo di calcolo basato sugli elementi. Questo valore è dettato dalla definizione di probabilità di re-identificazione

modellata con il metodo di calcolo in oggetto. Tale probabilità, come descritto nel capitolo delle definizioni, vale $PR_D(r = u|b) = \frac{1}{|M(D,b)|}$, pertanto non può assumere valori strettamente compresi tra 0.5 e 1. Questo può non essere valido per gli attacchi utilizzati con il metodo avente una conoscenza basata sulle sequenze. Infatti, la probabilità di re-identificazione in tal caso può assumere al numeratore valori naturali positivi: $PR_D(r = u|b) = \frac{|M_N(D_u,b)|}{|M_D(\cup_{i=1}^{n} D_i,b)|}$. Pertanto il risultato può assumere qualsiasi valore reale positivo compreso tra 0 e 1.

Complessivamente, tutti gli esperimenti simulati con il metodo di calcolo del rischio basato sulla conoscenza degli elementi mostrano un andamento abbastanza simile. In particolare, gli attacchi ElementsAttack e ElementsSequenceAttack con $k = 1$ possiedono soltanto il 10% di individui con rischio massimo. Il rimanente tiene un rischio minore o uguale a 0.5. L'attacco sulla frequenza e quello considerando le probabilità, pur mantenendo un andamento simile, si discostano maggiormente per il numero di individui con rischio massimo: il primo ne possiede il 20% circa mentre il secondo oltre la metà della popolazione ha rischio pari ad 1.0. All'aumentare di k tutti gli attacchi mostrano un imminente aumento dei livelli di rischio portando il numero di individui con rischio massimo al 90%. L'unico attacco con un risultato leggermente più apprezzabile è l'ElementsAttack, infatti con $k = 2$ possiede un buon 20% di individui con rischio minore o uguale a 0.5.
É interessante notare come l'ElementsTimeAttack, avendo già una conoscenza di base molto dettagliata, l'aumento di k non modifica così tanto i livelli di rischio. Inoltre, nei nostri esperimenti abbiamo utilizzato un livello di precisione giornaliero. Non sarebbe banale poter valutare l'andamento della distribuzione utilizzando un livello di precisione maggiore, ad esempio su base mensile.



(a)

Figura 5.3: ElementsTimeAttack: funzione di distribuzione

Gli attacchi simulati con il metodo di calcolo basato sulla conoscenza della sequenza

mostrano complessivamente lo stesso andamento, a parte che per l'ElementsProbabilityAttack. Tutti gli esperimenti con $k = 1$ mostrano che soltanto il 10% circa degli individui possiede un rischio massimo di re-identificazione. Inoltre, analizzando la funzione possiamo leggere che l'80% ha un rischio inferiore a 0.35, il 10% circa è compreso tra 0.35 e 0.5 e infine abbiamo una minima parte, circa l'1% con un rischio non massimo ma superiore a 0.5. All'aumentare di k possiamo notare che la funzione scende notevolmente portando all'85-90% il numero di individui con rischio massimo. L'ElementsProbabilityAttack mostra risultati non apprezzabili già a partire da $k = 1$, infatti complessivamente al variare di k abbiamo fin da subito il 90% degli individui con rischio massimo.

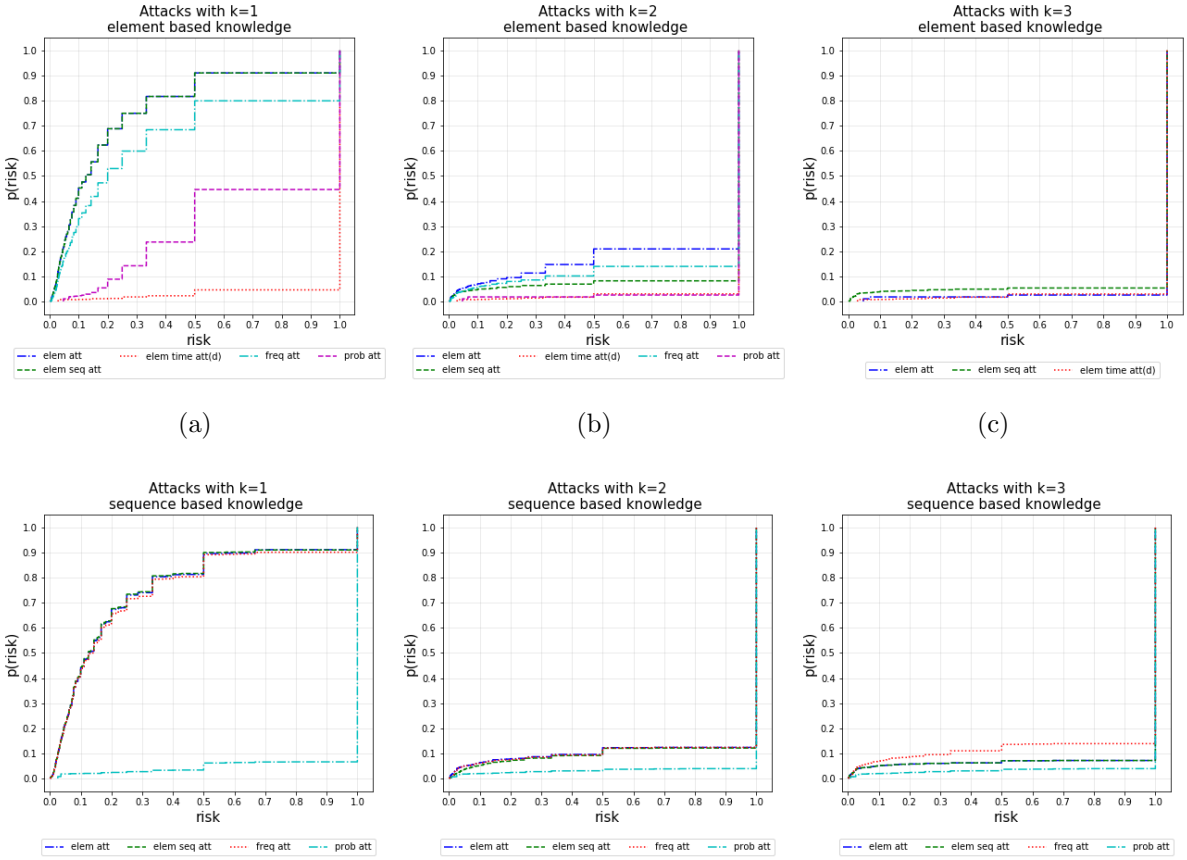
La seguente tabella mostra i tempi di calcolo del rischio relativamente ad un insieme di attacchi eseguiti con il metodo di calcolo con la conoscenza basata sugli elementi.

	k=1	k=2	k=3	k=4
ElementsProbabilityAttack	107,5 hour	5,3 hour	6,2 min	6,1 min
ElementsFrequencyAttack	115,12 hour	5,9 hour	6,6 min	6,3 min
ElementsAttack	119,34 hour	7,5 hour	6,5 min	6,2 min

Figura 5.4: Valutazione delle performance di un set di attacchi

Come si può notare all'aumentare della dimensione della conoscenza i tempi si riducono drasticamente. Questa rapida riduzione della complessità di calcolo è dovuta dal fatto che ogni calcolo del rischio viene eseguito tenendo conto degli utenti precedenti con rischio massimo. In particolare non viene rieseguito il calcolo del rischio nei confronti di tutti quegli utenti il cui rischio precedente era già massimo.

Quest'ultimo set di attacchi è un'interessante tipo di visualizzazione che, fissato il valore di k, mostra l'andamento della probabilità di rischio al variare del tipo di attacco. Ogni esperimento viene fatto per ogni tipo di calcolo del rischio. Un risultato interessante lo otteniamo fissando $k = 1$ e considerando il metodo basato sugli elementi. La funzione ci mostra chiaramente la differente evoluzione di tutti gli attacchi. L'attacco in funzione dell'elemento e del tempo (con precisione giornaliera) è il più penalizzato, a seguire abbiamo l'ElementsProbabilityAttack con circa il 45% con rischio minore o uguale a 0.5. A seguire abbiamo l'ElementsFrequencyAttack e i due ElementAttack (con e senza ordine) aventi rispettivamente 20% ed il 10% degli utenti con rischio massimo.



(a) (b) (c)

(d) (e) (f)

Figura 5.5: Plotting delle distribuzioni cumulative del rischio con dimensione fissa

Capitolo 6

Conclusioni

La privacy dei dati personali è attualmente uno degli argomenti più discussi nell'analisi dei dati. Questa preziosa risorsa attira molti interessi e proprio per questo i fenomeni di violazione dei dati personali stanno crescendo giorno dopo giorno. Allo stesso tempo, l'interesse di aziende, imprese ed analisti è giustificato dal fatto che i dati permetto loro di scoprire molte più sfaccettature sulle dinamiche della società. Il miglioramento del benessere sociale e le nuove scoperte sono soltanto alcuni degli obbiettivi che possono essere raggiunti attraverso una attenta analisi dei dati. E' quindi interesse di entrambe le parti trovare metodologie che proteggano la privacy individuale, permettendo allo stesso tempo l'analisi significativa dei dati personali.

Riteniamo che la valutazione del rischio di privacy sia uno dei passaggi fondamentali nella costruzione di un ecosistema di protezione della privacy. Tramite gli strumenti di valutazione del rischio sviluppati, il gestore di un provider è in grado di quantificare il rischio dei propri dati e capire quali individui siano maggiormente a rischio. Alcuni dei più tradizionali framework quantificano il rischio prendendo in considerazione lo scenario peggiore, ad esempio, assumendo che l'avversario conosca il maggiore numero di informazioni possibili per re-identificare un individuo. Ma questo come possiamo immaginare è surreale. Il framework PRUDEnce consente di effettuare una valutazione sistematica del rischio di privacy a partire da un sottoinsieme di conoscenze generato matematicamente, permettendo ai fornitori di calcolare il rischio su background di conoscenze differenti.

Sebbene questo sia un notevole passo avanti nella giusta direzione, l'obiettivo di questo progetto è stato quello di sviluppare una libreria in Python, definendo nuovi modelli di attacchi ed una struttura che permettesse di gestire dati sequenziali di tipo generico. In questo modo le procedure sono in grado di valutare il rischio di privacy a partire da qualsiasi tipo di dato sequenziale, indipendentemente se appartiene alla famiglia dei dati di mobilità o di retail, abbassando notevolmente l'investimento in termini di tempo necessario nell'implementazione e classificazione di procedure ad hoc per un certo tipo di

dato sequenziale.

Un interessante contributo aggiuntivo potrebbe essere quello di ampliare il ventaglio di attacchi. In questo modo avrei a disposizione un background di conoscenze più vasto con cui esprimere ulteriori valutazioni del rischio. Sarebbe interessante affiancare al modulo della valutazione del rischio anche un modulo che permetta di stimare la qualità dei dati, ad esempio, cercando di capire in maniera automatica quali e quanti dati impattano fino ad un certo limite di rischio.

In conclusione, ritengo che un'attenta analisi dei dati rappresenti la frontiera necessaria al miglioramento del benessere personale e sociale. Il susseguirsi di nuove scoperte non fa altro che alimentare l'interesse di aziende e professionisti nel ricercare nuove soluzioni a nuovi problemi, permettendo ad entrambi di raccoglierne il beneficio.

Bibliografia

[1] Charu C. Aggarwal e Philip S. Yu, cur. *Privacy-preserving data mining: models and algorithms*. eng. Advances in database systems 34. OCLC: 255823401. New York, NY: Springer, 2008. ISBN: 978-0-387-70991-8.

[2] Rakesh Agrawal e Ramakrishnan Srikant. "Privacy-preserving data mining". en. In: *ACM SIGMOD Record* 29.2 (giu. 2000), pp. 439–450. ISSN: 01635808. DOI: 10.1145/335191.335438. URL: http://portal.acm.org/citation.cfm?doid=335191.335438 (visitato il 04/09/2019).

[3] Rui Chen, Benjamin C. M. Fung e Bipin C. Desai. "Differentially Private Trajectory Data Publication". In: *CoRR* abs/1112.2020 (2011).

[4] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel e Wouter Joosen. "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements". In: *Requir. Eng.* 16.1 (2011), pp 3–32. DOI: 10.1007/s00766-010-0115-7.

[5] "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data". In: *OJ* L281/31 (October 1995), pp. 1–9.

[6] Josep Domingo-Ferrer. "A Three-Dimensional Conceptual Framework for Database Privacy". In: *Secure Data Management: 4th VLDB Workshop, SDM 2007, Vienna, Austria, September 23-24, 2007. Proceedings*. A cura di Willem Jonker e Milan Petković. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 193–202. ISBN: 978-3-540-75248-6. DOI: 10.1007/978-3-540-75248-6_14. URL: https://doi.org/10.1007/978-3-540-75248-6_14.

[7] Fosca Giannotti, Anna Monreale e Dino Pedreschi. "Mobility Data and Privacy". In: *Mobility Data Modeling, Management, and Understanding*. A cura di E. Zimanyi C. Renso S. Spaccapietra. 2013, pp. 174–193.

[8] C. S. E. Institute. "OCTAVE – (Operationally Critical Threat, Asset, and Vulnerability Evaluation). http://www.cert.org/octave/". In: ().

[9] Frank McSherry e Kunal Talwar. “Mechanism Design via Differential Privacy”. In: *FOCS*. IEEE Computer Society, 2007, pp. 94–103.

[10] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen e Vincent D. Blondel. “Unique in the Crowd: The privacy bounds of human mobility”. In: *Scientific Reports* 3 (mar. 2013), 1376 EP -. URL: http://dx.doi.org/10.1038/srep01376.

[11] OWASP. “Risk rating methodology.” In: (). url=http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf.

[12] Francesca Pratesi, Anna Monreale, Roberto Trasarti, Fosca Giannotti, Dino Pedreschi e Tadashi Yanagihara. “PRUDEnce: a System for Assessing Privacy Risk vs Utility in Data Sharing Ecosystems”. In: *Transactions on Data Privacy* 11.2 (2018), pp. 139–167. URL: http://www.tdp.cat/issues16/tdp.a284a17.pdf.

[13] “Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)”. In: *OJ* L119 (May 2016), pp. 1–88.

[14] P. Samarati. “Protecting Respondents’ Identities in Microdata Release”. In: *IEEE Trans. on Knowl. and Data Eng.* 13.6 (nov. 2001), pp. 1010–1027. DOI: 10.1109/69.971193.

[15] Pierangela Samarati e Latanya Sweeney. “Generalizing Data to Provide Anonymity when Disclosing Information (Abstract)”. In: *PODS*. ACM Press, 1998, p. 188.

[16] Scikit-Mobility. “skmob mobility library”. In: (). URL: https://github.com/scikit-mobility/scikit-mobility.

[17] Latanya Sweeney. “k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY”. en. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (ott. 2002), pp. 557–570. ISSN: 0218-4885, 1793-6411. DOI: 10.1142/S0218488502001648. URL: http://www.worldscientific.com/doi/abs/10.1142/S0218488502001648 (visitato il 12/07/2019).

[18] Frank Swiderski e Window Snyder. *Threat Modeling*. O’Reilly Media, 2004. ISBN: 9780735637696. URL: https://books.google.it/books?id=qWjoUuFSmf8C.

[19] Yu Zheng, Licia Capra, Ouri Wolfson e Hai Yang. “Urban Computing: Concepts, Methodologies, and Applications”. In: *ACM Trans. Intell. Syst. Technol.* 5.3 (set. 2014), 38:1–38:55. ISSN: 2157-6904. DOI: 10.1145/2629592. URL: http://doi.acm.org/10.1145/2629592.