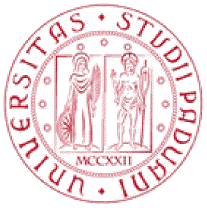


Università degli Studi di Padova

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Laurea in Ingegneria dell'Informazione

Metriche e meccanismi di anonimizzazione spaziale per location based services

Candidato:
Emma Amoroso

Relatore:
Prof. Nicola Laurenti

Co-Relatore:
Laura Crosara

16 Novembre 2023

Anno Accademico 2022/2023

Alla mia famiglia

Indice

Abstract

I continui progressi delle reti cellulari mobili e delle tecnologie di posizionamento GNSS hanno creato una forte spinta del mercato per le applicazioni basate sulla localizzazione. Degli esempi sono interventi nei casi di emergenza, pubblicità e intrattenimento. Una sfida importante per l'ampia diffusione dei servizi basati sulla localizzazione (LBS) è la gestione consapevole della privacy delle informazioni di localizzazione, fornendo garanzie per la privacy dei clienti mobili contro malintenzionati che tentano di acquisire dati. Questa tesi descrive i vari approcci che sono stati sviluppati fino ad ora per proteggere le informazioni relative alla posizione dell'utente e che allo stesso tempo permettono l'utilizzo degli LBS. A questo scopo vengono presentati degli algoritmi di oscuramento che permettono di schermare la posizione dell'utente che utilizza gli LBS. Sono state raccolte le metriche più utilizzate per valutare gli algoritmi di oscuramento dividendole in due categorie: privacy e qualità del servizio. Nella presentazione degli algoritmi sono applicate alcune metriche presentate per mostrarne l'applicazione e verificarne la validità. Infine viene discussa l'applicazione di algoritmi machine learning in quest'ambito, i quali costituiscono un importante sviluppo futuro per garantire maggiore privacy agli utenti di LBS.

Capitolo 1

Introduzione

1.1 Introduzione

Lo sviluppo di interconnessioni tra dispositivi si è maggiormente sviluppato negli ultimi anni grazie alla diffusione della comunicazione wireless e all'uso sempre più necessario della connessione internet per accede a servizi di vario tipo. In particolare, l'evoluzione della comunicazione wireless e della tecnologia di localizzazione ha fatto crescere il numero di attività che utilizzano la posizione dell'utente per fornire dei servizi, chiamati anche Location Based Service (LBS). Il modello di rete di un LBS è costituito da quattro entità come mostrato in figura: utenti mobili, sistema di localizzazione, rete di comunicazione wireless e il fornitore di servizi (LBS server).

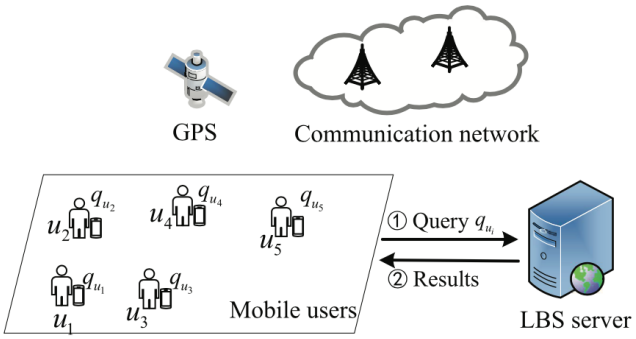


FIGURA 1.1: Schema struttura LBS, da [1].

In un sistema LBS, un utente mobile invia una richiesta ad un server su una rete di comunicazione wireless e di seguito il server LBS ritorna il risultato corrispondente alla richiesta dell'utente. Una richiesta LBS include un timestamp, informazioni sulla localizzazione dell'utente ottenute dal sistema di posizionamento ed infine il contenuto della richiesta. Degli esempi di LBS possono essere le app meteo, Uber e Google Maps. Visto che durante l'utilizzo degli LBS è essenziale che gli utenti forniscano la loro posizione è necessario trovare un equilibrio tra la privacy e la qualità del servizio che si vuole ricevere. Questo equilibrio è

fondamentale per l'utilizzo di questi servizi, infatti se la privacy richiesta dall'utente risulta essere troppo elevata significa che la posizione dovrà essere nascosta il più possibile. Se la posizione risulta essere eccessivamente oscurata le richieste dell'utente non possono essere soddisfatte a pieno perché i risultati si baseranno su una posizione ricevuta approssimativa. Se, per esempio, un utente volesse trovare un servizio vicino alla sua posizione è necessario che essa risulti più accurata possibile perché altrimenti la risposta ricevuta conterrà una locazione che non riflette l'effettiva richiesta dell'utente. Ci sono due tecnologie che le LBS usano per ricavare la posizione dell'utente [2]: una basata sul posizionamento dei satelliti e una basata sulla rete di comunicazione cellulare. La prima basata su sistemi GNSS, tra cui il più noto é il GPS, è la più sofisticata e precisa ed attraverso essa l'utente può ricavare la posizione direttamente tramite i dispositivi di localizzazione locale. Tuttavia, questa tecnologia presenta anche aspetti negativi: lunghi tempi per iniziare il processo di localizzazione, consumi elevati ed è più soggetta a interferenze se sono presenti ostacoli che possono attenuare la potenza del segnale. La seconda tecnologia è basata sull'invio del segnale dal dispositivo mobile verso alcuni ricevitori fissi che utilizzano le informazioni ricevute per calcolarne la posizione che rinvieranno al dispositivo. Una volta ottenuta la posizione, l'utente la può condividere con i server dei quali le LBS si servono, ed essi possono volontariamente o involontariamente fornire la sua posizione. La scoperta della locazione dell'utente può essere un grave danno per la privacy dello stesso perché attraverso essa si può ottenere un profilo della persona individuata, ad esempio: il luogo di residenza, lo stato di salute, religione, abitudini riguardanti sport e tempo libero. Questi dati possono essere sfruttati da diverse società per fare pubblicità mirate, quindi chi ne entra in possesso le può vendere a soggetti terzi. Con l'aumento di questi furti di informazioni è così maturata anche la consapevolezza delle persone su quanto sia importante mantenere un certo riservo sulla posizione in cui si trovano, innescando una reazione a catena che ha aumentato gli studi sulla sicurezza per contrastare la fuga di dati. Riguardo a questo aspetto una associazione senza scopo di lucro per la sicurezza delle informazioni, ISACA, ha svolto uno studio [3], basato su 1000 americani, dal quale è emerso che una buona percentuale di persone presenta grossi dubbi o informazioni incomplete sulla geolocalizzazione. Le principali preoccupazioni espresse riguardano le informazioni fornite per scopi di marketing (24%) e la possibilità che estranei riescano ad acquisire le informazioni senza consenso (24%). Una buona parte degli intervistati ha anche espresso preoccupazione sulla sicurezza personale che deriva dalla divulgazione della loro posizione. Infine si è notato che il 43% delle persone intervistate non legge gli accordi sulle app prima di scaricarle e il 25 % che li leggono non ne capiscono il significato. L'obiettivo di un avversario è quello di carpire la localizzazione di un utente, esso però deve avere delle conoscenze di base da cui partire per riuscire al meglio nel suo intento. Queste conoscenze di base possono essere classificate in tre tipi. La prima riguarda la capacità di un avversario di dedurre la presenza di posizioni fittizie di un utente se esse risultano troppo improbabili e quindi scartarle, per questo si cerca sempre di generare posizioni fittizie che risultino credibili. La seconda riguarda la possibilità di accedere a dati pubblici riguardanti informazioni statistiche che possono aiutare l'attaccante a fare delle ipotesi e migliorare la sua capacità di ricavare informazioni. Degli esempi possono essere informazioni demografiche rilasciate da enti governativi o di ricerca, dati sui trasporti, informazioni geografiche, reti stradali, aree residenziali, distribuzione della popolazione e così via. L'ultima riguarda dati personali che rivelano esplicitamente informazioni sulla posizione di qualcuno, come per esempio i social network. Numerosi studi si sono concentrati sul contrasto di questi attacchi attraverso l'uso

di metodi di oscuramento che proteggono l'utente e la sua posizione. I metodi su cui ci si basa per lo sviluppo di tale scopo sono: *Cloaking*, *Dummy Location*, *Differential Privacy*, *Mix Zones* e *Path Confusion*. Per valutare gli algoritmi che si basano su questi metodi sono presenti delle metriche che però variano molto in base all'articolo e all'argomento. Purtroppo non si è riusciti ancora a creare delle metriche universali per la valutazione del livello di privacy e la qualità del servizio mentre sono in utilizzo le LBS. In questa tesi sono state raccolte e descritte le metriche più usate assieme agli algoritmi in cui sono applicate. Inoltre si è voluto dare un'introduzione sui possibili sviluppi che si potrebbero avere attraverso l'utilizzo del machine learning, che sta prendendo sempre più piede con il miglioramento sempre più importante dell'intelligenza artificiale.

Capitolo 2

Meccanismi di Oscuramento

2.1 Cloaking

Il metodo *cloaking* [4] è basato sul concetto di generalizzazione della posizione dell'utente, attraverso la creazione di aree oscurate, che saranno definite più precisamente in seguito. Per la creazione di queste aree si utilizza un anonimizzatore, che collezionando e processando le richieste dell'utente decide quanto queste aree debbano essere grandi. Con richieste si intende un insieme di timestamp, locazioni degli utenti e contenuti delle richieste. A livello strutturale il metodo prevede che la richiesta di un utente venga inviata all'anonimizzatore che ne definisce l'area oscurata. Essa insieme al contenuto della richiesta viene inoltrata al server LBS, il quale dopo aver elaborato la risposta la rimanderà all'anonimizzatore che a sua volta la riporterà all'utente. Tutto il meccanismo è descritto nella figura 2.1 definita di seguito.

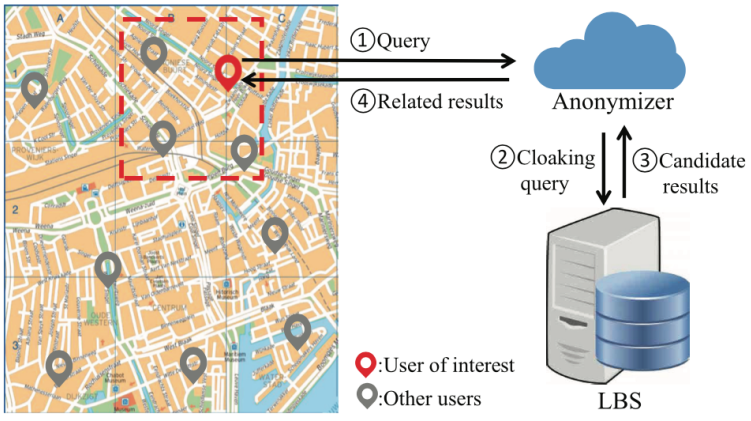


FIGURA 2.1: Tecnica Cloaking,da [1].

L'oscuramento di un'area può essere realizzato sotto il profilo spaziale e temporale. L'oscuramento spaziale prevede l'invio al server di una regione, grande o piccola, al posto della

posizione precisa reale dell'utente, chiamata area oscurata. L'oscuramento temporale prevede l'inserimento di un ritardo all'interno del timestamp della richiesta, per far elaborare al server i dati di un orario differente a quello in cui si sono inviati i dati. La progettazione e il design delle aree di oscuramento si basano su principi definiti per mezzo della tecnica *k-anonymity*. In riferimento all'articolo [5] possiamo affermare che questa tecnica consiste nel rendere i dati riguardanti ogni utente indistinguibili da almeno altri $k-1$ individui. In generale il metodo del *cloaking* basata sulla *k-anonymity* presenta quattro fattori principali che riflettono i requisiti di protezione della privacy [6] : k, A_{min}, A_{max} e T_{max}.

- k rappresenta il minimo numero di utenti richiesti per le impostazioni di anonimato dal servizio richiedente, chiamato grado di anonimato.
- A_{max} è l'area massima della regione di oscuramento. Se la regione risultasse essere troppo grande provocherebbe un allungarsi dei tempi per l'elaborazione dei dati.
- A_{min} è l'area minima della regione di oscuramento. Se la regione risulta essere troppo piccola è presente un rischio molto più elevato di rivelare la posizione.
- T_{max} rappresenta il più lungo periodo di anonimizzazione tollerabile. E' il periodo preciso in cui dovrebbe essere completato il processo di anonimizzazione dall'istante in cui viene effettuata la richiesta dell'utente.

Le variabili k e A_{min} rappresentano elementi che si riferiscono al minimo valore per quanto riguarda la qualità dell'anonimizzazione, mentre A_{max} e T_{max} descrivono la perdita di qualità del servizio causata dal processo di oscuramento. Il meccanismo *k-anonymity* è uno dei metodi più comuni e studiati applicati alla location privacy perché aumenta il grado di indistinguibilità delle informazioni. Tuttavia presenta le seguenti limitazioni. In primo luogo la *k-anonymity* manca di un'adeguata protezione della posizione; per esempio, se abbiamo un'area che presenta un numero basso di persone al suo interno la probabilità che l'attaccante scopra la posizione dell'utente aumenta. La seconda limitazione riguarda la possibilità che gli utenti possano non essere oscurati o che la regione oscurata possa risultare troppo grande, come risultato si osserva la diminuzione della qualità dei servizi LBS in aree scarsamente popolate [7]. Vista la presenza di questi problemi nella $k-anonimity$ gli studi si stanno concentrando sul rendere più difficoltoso per un attaccante violare la privacy di un utente. Per farlo è possibile tenere presente ulteriori elementi che caratterizzano le preferenze di privacy, i quali possono riguardare degli aspetti più personali o delle esigenze in termini di qualità del servizio che si vuole ottenere. Alcune soluzioni studiate sono: $l-$diversity[8], $t-$closeness[9], $p-$sensitivity[10] e $m-$invariance[11]. Attraverso queste modalità si ha una maggiore privacy della richiesta con l'oscuramento degli attributi sensibili per aumentare l'incertezza del collegamento tra l'utente e le sue informazioni.

2.2 Dummy Location

La *Dummy location* è un metodo che cerca di mascherare la posizione dell'utente inviando posizioni multiple, chiamate "dummies", al server LBS insieme alla posizione reale. Le locazioni fittizie sono scelte in maniera randomica; tuttavia cercano di soddisfare alcuni criteri che analizzeremo in seguito. Un esempio per comprendere meglio il funzionamento è visibile in figura 2.2:

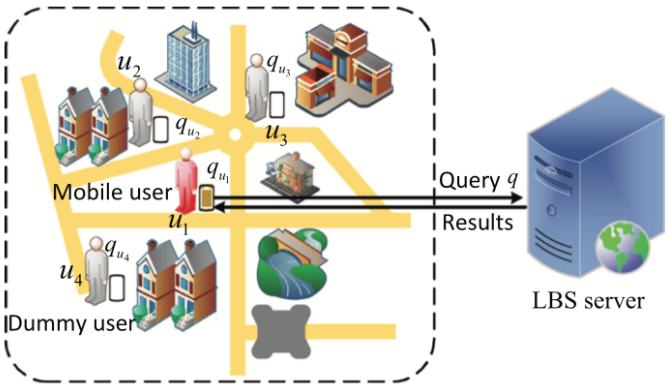


FIGURA 2.2: Tecnica dummy locations,da [1].

Si può notare infatti che viene descritta una generazione di tre posizioni chiamate u_2, u_3, u_4 le quali sono prese in un'area limitrofa a quella dell'utente, chiamata u_1. Il passo successivo è l'invio in contemporanea di quattro richieste provenienti dalle quattro posizioni al server LBS con lo stesso contenuto. Le richieste sono elaborate dall'LBS e vengono rispedite al mittente e filtrate; quindi, l'unica risposta presa in considerazione è quella basata sulla locazione reale dell'utente. La scelta delle *dummy locations* oltre ad essere casuale può presentare delle caratteristiche che possono preservare ulteriormente la privacy dell'utente. Si cerca infatti di fare in modo che gli algoritmi *dummy* seguano delle regole per creare un equilibrio tra la qualità del servizio e la privacy. Per farlo le locazioni fittizie non devono essere troppo vicine all'utente o troppo lontane, perché essere nelle vicinanze aumenta il rischio di essere individuati, mentre essere distanti diminuisce la qualità delle risposte ricevute. Ulteriori fattori che possono essere considerati per rendere le locazioni fittizie più credibili riguardano la possibilità di considerare la frequenza di richieste effettuate, la distribuzione delle locazioni false e la conformazione fisica del territorio considerato. Generalmente, comparato con altri metodi di oscuramento, il metodo basato su dummy location ha alcuni vantaggi:

- E' indipendente da terze parti, poiché la struttura del metodo comprende fisicamente solo dispositivi mobili e server LBS.
- Permette di avere risultati accurati perché, se pur nascosta, viene inviata la posizione dell'utente.
- Non è necessaria la condivisione di chiavi di crittografia tra utenti e server LBS.
- Rimane utile anche se gli attaccanti conoscono il meccanismo di protezione usato, perché dovrebbero localizzare l'obiettivo tra un elenco di altri utenti anonimi.

Tuttavia, questo metodo ha un aspetto negativo da non sottovalutare, infatti comporta un costo alto in termini di traffico di informazioni. L'invio al server LBS di molteplici richieste, di cui solo una verrà considerata, può avere come conseguenza lo spreco di risorse che potrebbero essere usate in maniera più proficua. Inoltre, se al server arrivano troppe richieste simultanee il tempo di risposta può essere elevato, portando a una significativa diminuzione della qualità del servizio.

2.3 Differential Privacy

Sono presenti molti algoritmi di oscuramento che fanno uso della *differential privacy*, introdotto da Dwork [12], per valutarne le prestazioni. Essa infatti è uno dei fattori standard di misurazione della privacy a causa della sua garanzia di privacy dimostrabile rigorosamente. Viene definita formalmente con la seguente definizione:

Definizione 1 *Assumiamo di avere un algoritmo randomizzato A, dove con randomizzato si intende che l'input non corrisponde a un determinato valore di output, e un numero reale positivo ε. L'algoritmo A fornisce ε-differential privacy se, per ogni coppia di database chiamati D_1 e D_2 vicini tra loro che differiscono di un solo elemento, un output arbitrario $S \subseteq Range(A)$,*

$$P_r[A(D_1) \in S] \leq e^{\varepsilon} \cdot P_r[A(D_2) \in S] \tag{2.1}$$

In linea di principio, l'approccio *differential Privacy* trasforma l'interrogazione esatta di un set di dati in una distribuzione, specificando la probabilità di ottenere lo stesso risultato interrogando due set di dati adiacenti. E' possibile applicare la *differential privacy* nel contesto della LBS considerando la locazione come un'informazione sensibile che deve essere protetta il più possibile. Per preservare la privacy in questo contesto è necessario perturbare la locazione dell'utente, per farlo una delle possibilità è aggiungere un rumore nella posizione reale così da comunicare una locazione fittizia al server LBS. Questo approccio è stato studiato nell'articolo [13] e prende il nome di *geo-indistinguishability*, il quale può proteggere la reale locazione dell'utente, mentre permette il rilascio di informazioni approssimate utilizzate per avere un qualche tipo di servizio. Formalmente viene definita come in seguito :

Definizione 2 *Si assuma di avere delle possibili posizioni di un elenco di utenti X e un elenco di pseudo-posizioni probabili Z, con $d(\cdot,\cdot)$ la distanza euclidea. Per ogni due locazioni x_1, $x_2 \in X$, $z \in Z$ e $d(x_1, x_2) \leq r$, un algoritmo K si dice $\varepsilon - Geo{-}differentially\ private$ se:*

$$P_r[K(x_1) = z] \leq e^{\varepsilon \cdot d(x_1, x_2)} \cdot P_r[K(x_2) = z] \tag{2.2}$$

dove ε indica il grado di privacy ad una unità di distanza.

Questa definizione indica che, per delle distanze molto piccole tra locazioni attuali di due utenti x_1 e x_2, le probabilità di generare la stessa posizione z e considerarla come la nuova locazione sono simili. Al contrario, se la distanza è elevata la differenza tra le probabilità aumenta. Questa differenza dipende molto dal parametro $\varepsilon \cdot d(x_1, x_2)$, che rappresenta il livello di protezione della privacy. La definizione sopra riportata formalizza la nozione di protezione della posizione dell'utente entro un raggio r. La $Geo - indistinguishability$ presenta un meccanismo pratico per l'applicazione nel contesto degli LBS e la possibilità di miglioramento, affrontata in diversi studi. Nell'articolo [14], per esempio, vengono riutilizzate le locazioni fittizie se la posizione reale dell'utente risulta ancora nelle vicinanze. Un altro approccio è la possibilità di adattare l'inserimento di rumori in una mappa [15] considerando la densità di popolazione intorno alla posizione corrente dell'utente. La $differential\ privacy$ presenta i seguenti vantaggi e svantaggi [1]:

- Contrariamente alle altre tecniche definisce un rigoroso modello dell'avversario, fornisce una rappresentazione qualitativa stringente e una prova di preservazione della privacy. Questo può mitigare il rischio di violazioni della privacy mentre si garantisce l'usabilità dei dati.
- Dopo l'inserimento controllato di rumore, si nota che la quantità di rumore aggiunta è indipendente dalla taglia del dataset. Come conseguenza, per dataset grandi la minima quantità di rumore può ancora risultare in un alto livello di privacy.
- Questa tecnica non funziona bene se i dati sensibili dell'utente sono correlati in un dataset; perché in questo scenario il rischio di danneggiare i dati di localizzazione è alto.
- Non viene considerata la possibilità che l'attaccante abbia acquisito delle conoscenze precedenti e questo comporta una protezione debole contro un avversario preparato.

2.4 Mix Zones

Questo metodo viene inizialmente introdotto nell'articolo [16] nel contesto degli LBS, qui le *mix zones* vengono definite come delle regioni spaziali dove le applicazioni non possono avere accesso ad alcuna informazione riguardante gli utenti all'interno. Vengono quindi realizzate delle zone completamente oscurate in cui gli utenti, mentre sono al loro interno, non possono fare richieste a nessun server LBS esterno. Inoltre, sono regioni in cui un certo numero di utenti entrano, cambiano pseudonimo ed escono in modo tale che non si riesca a collegare il loro nuovo pseudonimo con quello precedente. Questo metodo di cambio frequente di pseudonimi è molto efficace nel proteggere la privacy dell'utente. La letteratura più recente per la costruzione delle *mix zones*, che utilizza figure rettangolari o circolari, è vulnerabile agli attacchi temporali e in questo contesto non viene garantita la privacy dell'utente. Infatti considerando quattro utenti u_1, u_2, u_3 e u_4 e il cui istante di ingresso differisce di molto, un attaccante potrebbe collegare facilmente lo pseudonimo con l'utente. Per resistere agli attacchi temporali in alcuni articoli viene suggerito l'utilizzo di una finestra temporale applicata alla mix zone, durante la quale gli utenti rimangono all'interno dell'area ed quindi evitare di uscire troppo distanziati. Ad esempio, nell'articolo [17] si propone una *cryptographic mix zones* dove le zone oscurate sono situate nelle aree in cui sono presenti più utenti; di conseguenza, la finestra temporale è determinata dal ritardo degli utenti che entrano nella mix zone. Osservando la Figura 2.3, si può vedere come l'ingresso e l'uscita degli utenti nello stesso periodo creino confusione nel server LSB.
Tuttavia il metodo delle *mix-zones* non protegge dagli attacchi basati su informazioni ricavate da "canali laterali", i quali si riferiscono a dati che possono essere dedotti o estrapolati da comportamenti, modelli o informazioni secondarie. Inoltre gli utenti all'interno di una *mix-zone* non possono usare alcuni servizi, come, ad esempio, quelli che richiedono l'utilizzo di un'identità coerente nel tempo, infatti nella *mix zone* c'è un continuo cambio di pseudonimi che può causare problemi. Nello specifico, se un utente mobile sottopone una richiesta ad un LBS per cercare degli amici nelle vicinanze e simultaneamente invia un'altra richiesta di informazioni riguardante alcuni ristoranti vicini in base al suo profilo, il server LBS richiederà una identità coerente dell'utente per poter determinare in modo esatto il suo profilo e quello degli amici. Oltretutto il metodo *mix zones* fallisce anche nel supporto dei servizi LBS quando

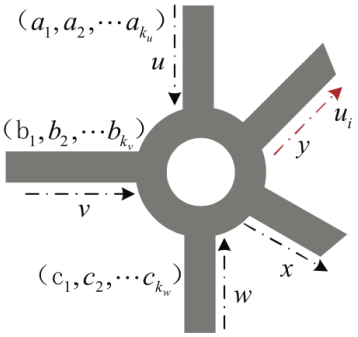


FIGURA 2.3: Tecnica mix-zone, da [1].

un utente mobile è dentro alla mix zone, poiché la sua locazione non può essere inviata al LBS. Per minimizzare la degradazione della qualità del servizio, causata da interruzioni verso gli utenti mobili, la distribuzione delle *mix zone* nel sistema dovrebbe essere ottimizzata al fine di limitarne il numero totale richieste così da ottenere un certo grado di anonimato.

2.5 Path Confusion

Path confusion è un metodo di preservazione della privacy principalmente adottato per gestire le traiettorie dei veicoli nel traffico, che cerca di evitare il collegamento tra campioni di posizioni consecutive e gli individui. In contrasto con l'approccio *mix zone*, nel quale una specifica area aveva bisogno di essere allocata, *path confusion* usa le interazioni tra utenti per raggiungere la protezione della privacy. In questo processo, gli attaccanti possono ottenere campioni di posizioni, ma falliscono all'identificazione del percorso completo dell'utente che vogliono tracciare. In questo approccio le identità possono non essere anonime in quanto il metodo di basa sullo scambio di identità tra utenti. Per esempio, quando 2 veicoli si incontrano, si scambiano le identità con una probabilità specifica tale che il server non può collegare campioni di locazioni consecutive al proprietario. Tuttavia, questo metodo limita il livello della protezione della privacy per la traiettoria. Perciò, la tecnica del *path confusion* di solito introduce un anonimizzatore che agisce come intermediario tra un utente e un server di destinazione. L'anonimizzatore ritarda le richieste degli utenti in un periodo per aumentare significativamente la probabilità, degli utenti, di attraversare i percorsi utilizzati da altri veicoli, aumentando la confusione del server LBS. Il server anonimizzatore non condividerà le locazioni accumulate finché non avrà raggiunto il livello di anonimato cercato, nel lasso di tempo prefissato. Come conseguenza esiste la possibilità che, durante un lungo intervallo temporale, la posizione dell'utente non sarà inviata al server LBS, il quale porterà ad avere una diminuzione della qualità del servizio. Infatti, durante questo lasso di tempo, il server non riceve informazioni quindi non può fornire delle risposte. Per mitigare questo effetto, nell'articolo [18] viene presentato un modo per fare una predizione di mobilità. L'idea principale è che l'anonimizzatore predica le intersezioni tra percorsi di utenti, sottoponga le richieste ai server LBS in base alla locazione prevista e salvi i dati. Quando l'utente arriva nella locazione effettiva, il server di anonimizzazione controlla se i dati salvati sono già stati elaborati. L'accuratezza e la lunghezza della previsione del percorso sono i principali fattori che hanno effetto su questo meccanismo.

Capitolo 3

Metriche

3.1 Metriche riguardanti la Privacy

Relative anonymity level:

É una misura del livello di anonimato fornito dall'algoritmo di *cloaking*, normalizzato dal livello di anonimato richiesto dal messaggio. Con la dicitura m_s si intende un messaggio appartenente ad un insieme S di messaggi a cui è possibile associare almeno un messaggio m_t, indicato come un messaggio anonimizzato. Il messaggio m_s è caratterizzato da parametri quali:

- k un numero intero maggiore di zero, che indica il livello di anonimato che si vuole ottenere su quel determinato messaggio. Con questo valore si definisce che il messaggio per essere anonimizzato richiede di avere le coordinate all'interno di un'area che contenga almeno $k-1$ altri utenti, per cui più il valore di k risulta elevato e più la privacy aumenterà.
- Le coordinate reali (x, y, t) della posizione e dell'istante in cui utente vuole inviare la richiesta.
- I valori (dx, dy, dt) indicano la quantità di spostamento massima da (x, y, t) che le informazioni spazio-temporali possono avere durante l'anonimizzazione del messaggio m_s in m_t.

Il messaggio m_t viene correttamente anonimizzato da una funzione suriettiva definita $R(m_s)$. L'immagine di uno specifico m_s è il sottinsieme F' di F che indica l'insieme dei messaggi anonimizzati. Con il termine $B_{c_l}(\cdot)$ si intende un contenitore al cui interno è presente una descrizione della posizione e del ritardo ottenuti su m_t e m dopo il processo di anonimizzazione, con m messaggio generico contenuto in F. E' possibile definire *relative anonymity level*, riferendosi all'articolo [19], con la seguente formula:

$$R_l = \frac{1}{|F'| \cdot k} \sum_{m_t = R(m_s) \in F'} |\{m | m \in F \wedge B_{c_l}(m_t) = B_{c_l}(m)\}| \tag{3.1}$$

Per il calcolo del R_l si inizia prendendo un messaggio m_s non anonimizzato e attraverso la funzione $R(m_s)$ si trovano le immagini m_t appartenenti al sottinsieme F'. Ricordando che la funzione è suriettiva è necessario fare una sommatoria su tutte le immagini m_t trovate. All'interno della sommatoria vengono contati quanti messaggi m appartenenti a F presentano la stessa descrizione del messaggio m_t considerato. Una volta eseguita la sommatoria su tutti gli elementi il risultato viene diviso per la dimensione dell'intero insieme F' considerato e per il livello di anonimato richiesto dal messaggio m_s, preso inizialmente. Un valore più alto del valore 1 di anonimità relativa significa che i messaggi sono anonimizzati con un valore di k più grande di quello richiesto.

Short-term Disclosure (SD):

In riferimento al metodo *dummy location* [20] si può ricavare la seguente metrica. Si prenda il caso in cui un utente si muove nel tempo e dopo ogni intervallo comunica con un LBS. Per anonimizzare l'utente, in questo caso è necessario inserire degli utenti fittizi che negli stessi istanti siano posizionati in luoghi diversi e inviino a loro volta delle richieste. Con T viene indicato l'insieme degli istanti t in cui, sia utenti fittizi che utenti reali, inviano le richieste mentre con L^t viene indicato l'insieme di locazioni, sia reali che fittizie, che nell'istante t sono presenti. La metrica è calcolata come segue:

$$SD = \frac{1}{|T|} \sum_{t=1}^{|T|} \frac{1}{|L^t|} \tag{3.2}$$

La sommatoria prende in considerazione per ogni istante considerato il numero di locazioni che l'attaccante vede e quindi la probabilità, se attaccasse in quel momento, di individuare l'utente reale. Si può affermare che SD è la probabilità media di successo di identificare la vera posizione dell'utente tra le locazioni fittizie.

Long-term Disclosure (LD):

Un'ulteriore metrica utilizzata nella *dummy location* può essere ricavata dall'articolo [20]. Date z traiettorie totali, tra le quali v traiettorie hanno delle intersezioni tra loro, si può dedurre che $z - v$ traiettorie non hanno intersezioni. Per queste v traiettorie è possibile numerare tutti i possibili percorsi che l'utente può intraprendere, il cui numero totale viene indicato con p. Si può quindi indicare LD come:

$$LD = \frac{1}{p + (z - v)} \tag{3.3}$$

Questa funzione dà come risultato un valore che indica la probabilità di un attaccante di identificare la traiettoria giusta dell'utente, se sono presenti più intersezioni tra percorsi. Per far comprendere meglio la metrica si prenda come esempio 3 possibili traiettorie z con 2 intersezioni che coinvolgono tutte e tre le traiettorie, quindi v avrà come valore 3. Questi valori presentano come conseguenza quella di avere 8 possibili percorsi p che un utente può intraprendere. Come risultato l'attaccante avrà come probabilità $\frac{1}{8}$ di indovinare quella giusta e quindi intercettare l'utente.

Location Privacy:

Questa metrica, presa per analizzare un algoritmo basato su *differential privacy* [21], esprime il livello di privacy che un utente presenta nei vari contesti in cui si trova. Per questo in seguito si è dato un modello con il quale realizzare questo concetto così da avere un metro di giudizio più preciso. Per iniziare, si definisce $a(t) = \langle t, r \rangle$ l'attuale locazione dell'utente al tempo t. Questo dato cercherà di essere acquisito dall'attaccante attraverso delle osservazioni riguardanti le pseudo-locazioni dell'utente al tempo t definite come $o(t) = \langle t, r' \rangle$. Si supponga inoltre che l'attaccante sia a conoscenza del profilo dell'utente $\phi(\cdot)$ e che utilizzi le sue precedenti esperienze per infliggere un attacco rivolto alle posizioni osservate, ottenendo una stima $\hat{r}$ di r. Formalmente la probabilità di fare una stima $\hat{r}$ come posizione effettiva dell'utente r in base alla posizione osservata r' si indica come:

$$h(\hat{r}|r') = Pr\{a(t) = \langle t, \hat{r} \rangle | o(t) = \langle t, r' \rangle\} \tag{3.4}$$

In particolare, le informazioni acquisite dall'attaccante ricavate da attacchi precedenti, riguardano la probabilità che un utente sia in una determinata locazione quando esso accede a un determinato LBS. Questa informazione si riflette sul calcolo della stima $\hat{r}$ al tempo t che è collegata alla pseudo-locazione r' osservata allo stesso tempo t. Si nota che la formulazione dell'attacco è indipendente dal fatto se venga considerato o meno che l'evento sia stato anonimizzato da un LPPM. Per poter misurare la privacy viene ipotizzato di conoscere l'identità dell'utente e la probabilità che quell'utente si trovi nella posizione r, indicato con $\phi(r)$. Con $f(r'|r)$ si definisce la probabilità di sostituire la posizione r dell'utente con una posizione r' fittizia, derivante dall'uso di un LPPM. Inoltre, $d_p(\hat{r}, r)$ indica una distanza tra la posizione stimata $\hat{r}$ e la posizione dell'utente r. La funzione d_p dipende dalla semantica del luogo o dalle richieste di sicurezza che l'utente vuole quando utilizza certi servizi, per cui deve essere scelta in base al caso che si vuole studiare. Per esempio, se l'utente vuole nascondere solo la sua esatta posizione una distanza adatta potrebbe essere la distanza di Hamming tra la posizione stimata $\hat{r}$ e la posizione reale r descritta in questo modo:

$$d_p(\hat{r}, r) = \begin{cases} 0, & se \;\; \hat{r} = r \\ 1, & altrimenti \end{cases} \tag{3.5}$$

Alternativamente, la $d_p(\hat{r}, r)$ potrebbe dipendere dalla distanza fisica tra $\hat{r}$ e r, in questo caso potrebbe essere utile modellare la funzione come una distanza Euclidea tra queste due locazioni, come la distorsione dell'errore quadratico:

$$d_p(\hat{r}, r) = (\hat{r} - r)^2 \tag{3.6}$$

Con la formula descritta di seguito si quantifica la privacy di un utente come valore atteso della distanza $d_p(\hat{r}, r)$. L'aspettazione si calcola sui valori di r, r' e $\hat{r}$:

$$Privacy(\phi, f, h, d_p) = \sum_{\hat{r}, r', r} \phi(r) f(r'|r) h(\hat{r}|r') d_p(\hat{r}, r) \tag{3.7}$$

Se viene sostituita una delle due definizioni date della $d_p(\hat{r}, r)$ nella (3.7) si osserva che si avrà un maggior valore di privacy quando la stima della locazione $\hat{r}$ e la posizione reale dell'utente r sono diverse o risultano essere distanti tra loro.

Entropy:

Un'altra misura presa in considerazione è l'entropia, la quale cattura l'incertezza dell'attaccante nell'intercettare gli utenti. In applicazione al metodo *mix zones* [22] possiamo dire che l'attaccante cerca di individuare il collegamento tra il nuovo pseudonimo dell'utente, ottenuto all'ingresso della nuova mix-zone, e quello ottenuto in precedenza in un'altra mix-zone. Con ai e aj si definiscono gli pseudonimi ricevuti nella prima mix-zone appartenente ad un insieme A di utenti anonimizzati. Quando avviene l'ingresso in una nuova *mix zone* lo pseudonimo cambia quindi il nuovo nome di ai sarà ai', la stessa cosa vale per aj che verrà chiamato aj'. Con $p_{ai' \to aj}$ si indica la probabilità di mappare l'utente ai' in un qualsiasi utente aj. La misura dell'entropia in questo contesto è così definita:

$$H(ai') = -\sum_{aj \in A} p_{ai' \to aj} \cdot \log_2(p_{ai' \to aj}) \tag{3.8}$$

La sommatoria eseguita viene fatta rispetto a tutti gli utenti anonimi perché bisogna considerare tutte le possibili associazioni tra utenti. Più il valore dell'entropia è elevato e più sarà difficile per l'attaccante catturare le informazioni. Tuttavia l'entropia calcolata in questo modo non risulta molto precisa se la mappa delle associazioni non è uniforme. Infatti se sono presenti nel calcolo dell'entropia delle probabilità più alte di altre il valore può risultare accettabile ma ci saranno associazioni più probabili di altre che può favorire l'attaccante nell'individuare correttamente l'utente.

Normalized Entropy:

E' conosciuta come grado di entropia e viene definita come il rapporto tra l'entropia ottenuta sperimentalmente divisa per l'entropia teorica che si ottiene a parità di utenti anonimi considerati, appartenenti ad un insieme A di utenti anonimizzati. In altre parole è la misura di quanto il valore ottenuto sperimentalmente sia vicino al valore ottenuto teoricamente.

Pairwise Entropy:

In riferimento all'articolo [22] per migliorare il calcolo dell'entropia, descritta in precedenza, si è cercato un modello che si riconducesse ad un risultato in cui le probabilità di mappatura degli pseudonimi presenti potessero essere considerate quasi uniformi. Per farlo è stato proposto un modello di calcolo a coppie chiamato H_{pair}. Essendo questa metrica utilizzata nel contesto del metodo *mix zone* sono presenti utenti i quali cambiano pseudonimi ogni volta che entrano in contatto con una nuova *mix zone*. Si ricorda che con ai e aj si indicano gli pseudonimi ottenuti nella prima *mix zone* mentre con ai' e aj' gli pseudonimi ottenuti in un'altra *mix zone*, sempre riferiti agli stessi utenti. La costruzione del modello prevede di prendere due utenti alla volta e procedere con il calcolo di due entropie che tengano conto di tutte le possibili probabilità di associazione tra gli pseudonimi ottenuti nella prima mix zone e quelli ottenuti in seguito. E' possibile definire H_{pair} come:

$$H_{pair}(ai, aj, t) = -(p_{ai' \to ai}(t) \log p_{ai' \to ai}(t) + p_{ai' \to aj}(t) \log p_{ai' \to aj}(t)) \tag{3.9}$$

$$H_{pair}(aj, ai, t) = -(p_{aj' \to ai}(t) \log p_{aj' \to ai}(t) + p_{aj' \to aj}(t) \log p_{aj' \to aj}(t)) \tag{3.10}$$

L'entropia *pairwise* tra due utenti ai e aj è l'entropia ottenuta considerando ai e aj come unici membri dell'elenco di utenti anonimi A. Così facendo abbiamo due eventi da analizzare: il primo caso riguarda l'uscita dell'utente ai dalla mix zone ma cambiato di nome ai', mentre il secondo riguarda la stessa azione ma con l'utente aj indicato con aj'. Per il primo evento abbiamo le seguenti mappe di probabilità $p_{ai' \to ai}$ e $p_{ai' \to aj}$, mentre per il secondo abbiamo $p_{aj' \to ai}$ e $p_{aj' \to aj}$. Se la probabilità di mappare lo pseudonimo ai' in ai è uguale alla probabilità di mapparlo in aj allora possiamo affermare che ai' è ugualmente probabile ad essere associato a ai o aj. Con questi dati l'attaccante ha la più bassa certezza di collegare lo pseudonimo ai' con il suo reale utente. Se una delle due probabilità è maggiore dell'altra avremo uno sbilanciamento e l'attaccante sarà più propenso a scegliere l'utente con una probabilità più alta. Per il secondo evento il ragionamento è analogo solo che lo pseudonimo a cui ci riferiamo è aj'. Di conseguenza se si riesce attraverso uno dei due eventi ad associare in maniera corretta lo pseudonimo con l'utente, si riuscirà a dedurre anche l'altra associazione. Per mitigare questo rischio si deve fare in modo che entrambe le entropie $H_{pair}(ai, aj, t_{out,ai'})$ e $H_{pair}(aj, ai, t_{out,aj'})$ abbiano come valore 1. Per misurare l'efficacia dell'entropia *pairwise* tra gli utenti ai e aj possiamo prendere il valore minimo tra le due entropie calcolato al tempo t_{out} in cui i due utenti escono dalla mix zone:

$$H_{pair}(ai, aj, t) = min\{H_{pair}(ai, aj, t_{out,ai'}), H_{pair}(aj, ai, t_{out,aj'})\} \tag{3.11}$$

Si può espandere le osservazioni fatte precedentemente e affermare che devono valere per tutte le coppie di possibili utenti presenti nella lista anonima.

Location Privacy per Path Confusion:

Si definisce *location privacy* l'accuratezza con cui un attaccante può individuare un determinato utente, prendendo in considerazione la distanza tra la posizione osservata di un utente e la sua posizione effettiva. Il concetto di privacy è strettamente collegato con il concetto di incertezza, per questo, solitamente, si utilizza come misura l'entropia. Tuttavia non considera se le posizioni degli utenti siano effettivamente diverse. Nel contesto del path confusion uno degli approcci per cercare di aumentare il più possibile la confusione di un attaccante sfrutta i punti in cui le traiettorie risultano essere vicine. Se esse non si sovrappongono vengono creati degli algoritmi che perturbano le traiettorie in modo tale che l'attaccante veda un incrocio e non capisca a quale traiettoria è associato l'utente. Per definire meglio la metrica riferita in questo contesto viene precisato che le posizioni dell'utente n sono indicate con coordinate nel piano (x, y), in cui con $(x_n(t), y_n(t))$ si indica la posizione reale dell'utente mentre con $(\tilde{x}_{m_j(n)}(t), \tilde{y}_{m_j(n)}(t))$ si indica la j-esima ipotesi di locazione dell'utente n in base alla posizione osservata dopo la perturbazione. Con $m_j(n)$ viene indicato un vettore multivariato m, che contiene tanti elementi quanti sono il numero degli utenti e ogni campo del vettore contiene l'indice dell'utente che questa ipotesi assegna ad un campione di localizzazione. Con i valori U, T, H e O vengono indicati rispettivamente l'insieme di utenti, l'insieme totale di istanti in cui faccio le osservazioni, il totale delle ipotesi effettuate dall'attaccante e il totale delle posizioni osservate dell'utente mentre con $f_n(\cdot)$ viene indicata

una distribuzione di probabilità. Attraverso tutti questi elementi si è sviluppata una metrica che coinvolge un valore atteso che quantifica l'errore commesso dall'avversario. Questa misura dà un'indicazione di quanto lontane, in termini di distanza, siano le ipotesi fatte dall'avversario rispetto alla reale posizione dell'utente. Definiamo per prima cosa $P({H_t}^j|O_t)$ e $d({H_j}^t, U^t)$ come segue:

$$P({H_j}^t|O^t) = \prod_{n=1}^{|U|} f_n(\tilde{x}_{m_j(n)}(t), \tilde{y}_{m_j(n)}(t)) \tag{3.12}$$

$$d({H_j}^t, U^t) = \sum_{n=1}^{|U|} \sqrt{(\tilde{x}_{m_j(n)}(t) - x_n(t))^2 + (\tilde{y}_{m_j(n)}(t) - y_n(t))^2} \tag{3.13}$$

e il valore atteso viene definito:

$$E[d] = \frac{1}{|U||T|} \sum_{t=1}^{|T|} \sum_{j=1}^{|H|} P({H_j}^t|O^t) d({H_j}^t, U^t) \tag{3.14}$$

Con $d({H_j}^t, U^t)$ si intende specificare la somma, rispetto a tutti gli utenti, delle distanze euclidee che derivano dalla differenza tra la posizione osservata in accordo con l'ipotesi j-esima associato all'utente n e l'effettiva posizione dell'utente n-esimo, entrambe considerate all'istante t. Per calcolare il valore di $P({H_j}^t|O^t)$ si considerano le posizioni degli utenti osservati indipendenti tra loro. Viene fatta un'ipotesi di locazione j, in base a delle osservazioni, in cui ogni utente considerato potrebbe essere localizzato. Attraverso il vettore multivariato è possibile considerare tutte le permutazioni degli utenti associandoli alla posizione ipotizzata. Tutte le densità di probabilità associate ad ogni permutazione vengono moltiplicate e se ne ricava una densità congiunta. Dopo aver fatto una serie di ipotesi si cercherà quella che massimizza il prodotto e che quindi che permetterà all'attaccante di eseguire un attacco più preciso. La $f(\cdot)$, in riferimento all'articolo [23], viene presa come una distribuzione di probabilità gaussiana multivariata. Si conclude che più il valore di $E(d)$ risulta elevato, più l'attaccante sbaglia nel processo di individuazione degli utenti.

Anonymity set size:

Si tratta di un metodo molto semplice di misurare l'anonimità utilizzato nel metodo *mix-zone* [22] su cui ci si può basare inizialmente. Fa riferimento al numero di possibili utenti resi anonimi che in un determinato momento sono situati nella stessa zona oscurata. Più utenti sono presenti e più sarà difficile ottenere le informazioni ricercate perché l'attaccante sarà costretto a cercare tra molti più utenti quello che gli interessa. Tuttavia questa metrica da sola non è sufficiente, dato che le probabilità di associazione degli pseudonimi degli utenti all'interno di una mix zone possono risultare non uniformi.

3.2 Metriche riguardanti la Qualità del servizio

Success Rate:

E' un'importante misura per valutare l'efficacia del modello basato sulla *k-anonymity*. Concretamente, l'obiettivo primario dell'algoritmo basato sul *cloaking* è di massimizzare il numero di messaggi anonimizzati correttamente m_t in conformità con i suoi vincoli di anonimato, come ad esempio il livello di privacy richiesto k. E' possibile prendere i valori definiti nella metrica *relative anonimity level* e prendere un sottinsieme del messaggi totali S indicato con S'. Il tasso di successo può essere formulato nel seguente modo:

$$T_s = \frac{|\{m_t | m_t = R(m_s), m_t \in F, m_s \in S'\}|}{|S'|} \cdot 100 \tag{3.15}$$

La formula si definisce come una divisione tra il numero di messaggi anonimizzati correttamente dalla funzione $R(m_s)$, quindi appartenenti all'insieme F, diviso per la cardinalità del sottinsieme S'. Il valore viene espresso come percentuale per dare un'idea più chiara e far capire in maniera più immediata se ci sono stati dei miglioramenti. Possiamo osservare che più il numero di messaggi anonimizzati è elevato, rispetto allo stesso numero di messaggi inizialmente considerati, più il tasso sarà elevato.

Relative spatial resolution:

Prendendo come riferimento l'oscuramento con il metodo *cloaking* è possibile ricondursi all'utilizzo dei valori definiti nella metrica del *relative anonimity level*. Viene indicata come risoluzione spaziale l'area rettangolare massima definita dalle misure limite del messaggio m_s relative agli assi x e y. Possiamo inoltre indicare con X_{m_t} e Y_{m_t} rispettivamente la coppia di coordinate x e y iniziali e finali ottenute, relative al processo di anonimizzazione del messaggio. Definiamo matematicamente la relativa risoluzione spaziale come:

$$R_s = \frac{1}{|F'|} \sum_{m_t = R(m_s) \in F'} \sqrt{\frac{2 \cdot dx \cdot 2 \cdot dy}{||X_{m_t}|| \cdot ||Y_{m_t}||}} \tag{3.16}$$

Dove $|| \cdot ||$ viene applicato alle coordinate riferite agli assi x e y, dando come risultato la misura dello spostamento effettivamente eseguito nell'anominizzazione. Il numeratore nell'equazione rappresenta l'area rettangolare che descrive lo spazio totale possibile nel quale il messaggio può essere anonimizzato, deriva dalla moltiplicazione delle dimensioni massime dx e dy moltiplicate entrambe per 2. Con il denominatore viene indicata l'area effettivamente utilizzata nello spostamento del messaggio m_s, dalla coordinata della posizione reale alla coordinata della posizione fittizia. Viene eseguita una sommatoria su tutte le possibili immagini della funzione suriettiva $R(m_s)$ dentro F' e si normalizza rispetto al numero di elementi all'interno di $|F'|$ così da ottenere un valore medio. Un alto valore della risoluzione spaziale implica che l'anonimizzazione è realizzata con una piccola regione di oscuramento spaziale rispetto alla disponibilità.

Relative temporal resolution:

In riferimento al metodo di oscuramento basato sul *cloaking* per questa metrica è possibile utilizzare i valori definiti nella *relative anonymity level*. Ricordando che anche sotto il profilo temporale si può oscurare un messaggio si è deciso di sviluppare una metrica che consideri questo aspetto. Innanzitutto viene definito il termine di risoluzione temporale come il ritardo massimo che può essere introdotto nel messaggio da anonimizzare. Con il simbolo I_{m_t} si indicano due istanti di tempo che definiscono rispettivamente l'istante reale, in cui l'utente invia il messaggio e l'istante fittizio nel quale si vuole inviare la richiesta al server LBS. Definiamo matematicamente la relativa risoluzione temporale come:

$$R_t = \frac{1}{|F'|} \sum_{m_t = R(m_s) \in F'} \frac{2 \cdot dt}{||I_{m_t}||} \tag{3.17}$$

Viene eseguito il rapporto tra il ritardo massimo che è possibile introdurre e il ritardo effettivamente introdotto calcolato con $||I_{m_t}||$, che indica la lunghezza dell'intervallo. La sommatoria è eseguita su tutte le immagini ottenute dalla funzione suriettiva $R(m_s)$ dentro F' e si normalizza rispetto al numero di messaggi $|F'|$ totali così da ottenere un valore medio. Un alto valore della relativa risoluzione temporale implica che l'anonimizzazione è realizzata con il più piccolo intervallo temporale e quindi con un piccolo ritardo dovuto all'anonimizzazione.

Message processing time:

E' una misura delle prestazioni in fase di esecuzione del massaggio durante la sua anonimizzazione. Il tempo per processare il messaggio può diventare un problema critico se la potenza di compilazione non è abbastanza per gestire l'arrivo di un alto numero di messaggi.

Distance Deviation:

In riferimento al metodo *dummy location* [20] possiamo introdurre un'ulteriore metrica. Si definisca L^t_{re} e $L^t_{d,n}$ come la locazione dell'utente reale e le locazioni degli utenti fittizi, tutte considerate nell'istante t. Questa metrica può essere descritta come la distanza media tra le traiettorie false e quella vera dell'utente. Viene indicata in questo modo:

$$dst = \frac{1}{|T|} \frac{1}{|L_d|} \sum_{n=1}^{|L_d|} \sum_{t=1}^{|T|} dist(L^t_{re}, L^t_{d,n}) \tag{3.18}$$

Il calcolo delle due sommatorie parte considerando la prima dummy location $L^t_{d,1}$ ed esegue la somma di tutte le distanze tra la locazione dell'utente L^t_{re} e la dummy location, considerando tutti gli istanti di tempo $|T|$ presi in esame. La sommatoria interna quindi tiene conto del numero di istanti considerati in una traiettoria mentre la sommatoria esterna prende in considerazione tutte le *dummy location*, create per confondere l'avversario. Il valore ottenuto viene diviso per il numero totale di fasce orarie e per il numero delle locazioni fittizie così da ottenere un valore medio della distanza. Se questa misura risulta essere troppo elevata avremo che il servizio a cui stiamo inviando le richieste può fornire delle risposte inutilizzabili a causa del fatto che le *dummy location* risultino essere troppo distanti dalla locazione reale. Mentre se la distanza risulta essere molto piccola possiamo avere delle conseguenze in termini di sicurezza perché l'attaccante può individuare la nostra posizione più facilmente.

Service Quality Metric:

La risposta che otteniamo da un server LBS dipende dalla pseudo-locazione r' in uscita dal LPPM e non dalla reale locazione dell'utente r. La distorsione introdotta per determinare una pseudo-locazione ha delle conseguenze sulla qualità del servizio che ogni utente sperimenta. Più la posizione dell'utente è simile alla pseudo-locazione inviata e più la qualità del servizio sarà elevata a discapito della privacy. La perdita di qualità dovuta al LPPM dipende dalle probabilità $f(r'|r)$ e $\phi(r)$ definite nella metrica *Location privacy* 3.7. Inoltre viene introdotto un nuovo parametro $d_q(.)$ che indica il valore che quantifica la perdita di qualità se viene sostituita la posizione r con r'. La metrica viene calcolata come valore media di $d_q(r', r)$ su tutte le r e r':

$$Q_{loss}(\psi, f, d_q) = \sum_{r,r'} \psi(r) f(r'|r) d_q(r', r) \tag{3.19}$$

Questa metrica definisce un valore preciso della media, riguardante la perdita di qualità del servizio dando un'indicazione su quanto bene le risposte ricevute siano accurate. Si basa sul concetto della similitudine tra la posizione dell'utente e la posizione fittizia: quando infatti la posizione fittizia viene fornita simile alla posizione reale degli utenti avremo una piccola perdita mentre se la posizione fornita è molto diversa avremo una perdita importante di qualità del servizio. In molte applicazioni infatti la qualità del servizio è inversamente proporzionale alla distanza fisica tra r e r'. Si può considerare il caso in cui si ricercano dei servizi nelle vicinanze, se la posizione fittizia dista anche solo pochi km si ottengono delle risposte molto diverse. Ci sono però anche servizi LBS nei quali la qualità del servizio è basata su altri criteri. Se vengono prese come esempio le applicazioni meteo, le previsioni che riceviamo si possono basare su uno spazio più grande quindi la pseudo-locazione r' basta che sia situata dentro una determinata area. Infatti i risultati ottenuti se r' si trova nella stessa città di r risultano comunque molto attendibili. Ipotizziamo che l'utente possa imporre un massimo di Q_{loss}^{max} che può essere tollerato prima di avere una perdita troppo grande di qualità del servizio e quindi avere delle risposte inutilizzabili. E' possibile imporre che per ogni LBS valga :

$$Q_{loss}(\phi, f, d_q) \leq Q_{loss}^{max} \tag{3.20}$$

Questa equazione vincola la funzione LPPM di oscuramento $f(r'|r)$ a non avere come output delle pseudo-locazioni che risultano, in media, in una perdita troppo grande di qualità del servizio. Si nota inoltre che, influenzando la soglia Q_{loss}^{max}, sul LPPM si ha una variazione anche sulla $d_q(.)$ che a sua volta dipende dalla distanza tra r e r'. Questo significa che aumentando o diminuendo Q_{loss}^{max} si ha un'influenza diretta sulla distanza tra la pseudo-locazione e la locazione reale dell'utente. Queste variazioni dipendono dal LBS con il quale vogliamo interagire e dalle richieste che vogliamo sottoporgli. Si riprenda l'esempio delle applicazioni meteo che, come detto in precedenza, non hanno bisogno di una qualità del servizio molto elevata. In questo caso il valore di Q_{loss}^{max} che possiamo imporre può essere aumentato fino a quando non si creano delle regioni in cui la distanza tra r e r' siano massime pur mantenendo gli stessi risultati. Possiamo quindi concludere che il livello di Q_{loss}^{max} dipenda

molto dalle applicazioni usate, se è troppo basso perdiamo in privacy, se troppo alto non riusciamo ad usare certi tipi di applicazioni.

Quality of Service:

La qualità dei dati ottenuti dagli LBS dipendono molto dall'accuratezza della posizione fornita quindi bisogna trovare un equilibrio tra le due cose. Le applicazioni presentano livelli variabili di robustezza in base al servizio fornito e dal fatto se hanno più o meno bisogno che la posizione dell'utente sia precisa o meno. Per questo motivo si è deciso nell'articolo [23] di misurare la qualità del servizio con una misura generica che non presenta degli elementi troppo precisi vista la difficoltà di questa misurazione. In riferimento ai valori definiti nella metrica *Location Privacy per Path confusion* descritta in precedenza è possibile definire l'errore di locazione medio come:

$$QoS = \frac{1}{|U||T|} \sum_{n=1}^{|U|} \sum_{t=1}^{|T|} \sqrt{((\tilde{x_n}(t) - x_n(t))^2 + (\tilde{y_n}(t) - y_n(t))^2} \tag{3.21}$$

Le sommatorie vengo fatte considerando Il numero di utenti U e il numero di istanti della traiettoria $|T|$. In pratica viene misurata semplicemente la differenza media tra la pseudo-locazione trovata dopo la perturbazione e la posizione reale di un utente. Una distanza bassa indica una maggiore precisione, quindi una QoS bassa, mentre una distanza alta indica una minore precisione, quindi una QoS alta.

Capitolo 4

Esempi Meccanismi di Oscuramento

In questo capitolo verranno descritti due algoritmi uno basato sul metodo *cloaking* e l'altro sul metodo *dummy location* entrambi definiti nel capitolo due.

4.1 Algoritmo basato sul metodo Cloaking

4.1.1 Scopo dell'algoritmo

L'articolo [19] afferma che la perturbazione della locazione è una tecnica efficace per supportare il metodo della *k-anonymity* e per gestire la violazione della privacy relativa alla posizione degli utenti. Se ogni locazione inviata da utenti mobili viene perturbata e la posizione rimpiazzata con una più generica, nella quale sono presenti altri $k-1$ utenti, l'avversario avrà un'incertezza nel collegare l'utente mobile con la sua posizione reale. L'incertezza cresce all'aumentare del valore di k, che procura un alto grado di privacy per gli utenti mobili. L'obiettivo dell'articolo è la personalizzazione del modello di *k-anonymity* per proteggere la posizione. Con il termine personalizzazione si intende che i requisiti di privacy variano in base al contesto e alle esigenze di ogni individuo; quindi, si sono utilizzati questi dati per creare un modello più efficace nell'adattamento alle preferenze degli utenti. Questo approccio risulta essere innovativo perché la maggior parte degli studi effettuati sui rischi di accesso agli LBS utilizza un modello rigido e poco personalizzabile. Un metodo per ottenere la personalizzazione è quello di permettere agli utenti di decidere il loro livello di privacy k in diversi istanti di tempo. Se il livello di privacy è alto si avrà bisogno di un maggior oscuramento, nello spazio e nel tempo, al fine di anonimizzare il messaggio. Come conseguenza si otterranno una bassa risoluzione spaziale e temporale inviata a un server LBS che, a sua volta, porterà ad avere dei problemi in termini di qualità del servizio ricevuto. Al contrario, se il valore di k é basso si avranno una risoluzione spaziale e temporale elevata, che comporterà problemi in termini di privacy; quindi, è necessario trovare un trade-off tra il livello di privacy e la qualità del servizio. Il modello di location privacy progettato in [19] ha due caratteristiche principali. La prima riguarda la possibilità del modello di far decidere all'utente il *minimum level of anonymity* desiderato e le *maximum temporal and spatial tolerances* che si è disposti ad accettare al momento della richiesta ad un LBS; con il valore

di k viene data un'indicazione sulla privacy, mentre con le massime tolleranze si identifica la qualità del servizio che si desidera ottenere. La seconda caratteristica riguarda la realizzazione di un sistema efficiente per la perturbazione dei messaggi, la quale è eseguita da un server di anonimizzazione. Per realizzarla è stato sviluppato un algoritmo di oscuramento che tiene conto del 'trade-off' sopra descritto. Tale algoritmo permette, inoltre, di elaborare un flusso di messaggi continuo e di collaborare con altri algoritmi di cloaking che perturbano le informazioni contenute nei messaggi inviati da utenti mobili prima di essere inoltrati ai server LBS.

4.1.2 Struttura base dell'algoritmo

Le informazioni che definiscono un messaggio m_s sono: il livello k di anonimato richiesto dal messaggio, le coordinate reali (x, y, t) che indicano la posizione reale dell'utente e l'istante in cui viene inviata la richiesta e infine i valori (dx, dy, dt) che indicano la quantità di spostamento massima che le coordinate reali possono avere durante il processo di offuscamento. Il processo di perturbazione dei messaggi m_s attuato dal server anonimizzatore viene diviso in quattro fasi. La prima fase, chiamata *zoom in*, comporta la localizzazione di un sottoinsieme di utenti, ognuno dei quali invia uno specifico messaggio m_s, che rimane in attesa di essere analizzato dall'anonimizzatore. Questo sottoinsieme, infatti, contiene i messaggi che potranno poi essere potenzialmente utilizzati per l'anonimizzazione di ulteriori nuovi messaggi ricevuti. La seconda fase, chiamata *detection*, è responsabile della ricerca di un particolare gruppo di messaggi all'interno del sottoinsieme trovato precedentemente, che saranno anonimizzati insieme ad ulteriori nuovi m_s messaggi ricevuti. Se questo gruppo di messaggi viene trovato, allora è possibile eseguire la perturbazione nella fase successiva. Quest'ultima fase, chiamata *perturbation* ha il compito di perturbare i messaggi e inoltrarli ai server LBS. Nella quarta e ultima fase vengono utilizzate le deadline di un messaggio che rappresentano il più alto valore dell'intervallo di tempo deciso dall'utente in cui il messaggio può essere anonimizzato, sulla base delle tolleranze scelte. Questa fase, chiamata *expiration*, controlla i messaggi in sospeso le cui deadline sono state superate e li elimina dall'elenco dei messaggi. Inoltre, all'interno dell'articolo viene sviluppato un teorema, chiamato CliqueCloak, sul quale si basa la scelta dell'insieme di messaggi da anonimizzare come gruppo. Gli algoritmi di cloaking che utilizzano questo teorema vengono definiti algoritmi di *CliqueCloak*.

4.1.3 Definizione Teorema *CliqueCloak*

Per prima cosa si definisce $B_{cn}(m_s)$ l'area spazio-temporale data del messaggio m_s che contiene: le coordinate e l'istante in cui il messaggio viene inviato all'anonimizzatore (x, y, t) e le tolleranze massime (dx, dy, dt) riguardanti area e ritardo le quali possono essere sfruttate dall'anonimizzatore. L'obiettivo principale dell'articolo è lo sviluppo di un algoritmo efficiente che trovi l'area di oscuramento dei messaggi più piccola possibile. Il problema viene, perciò, diviso in due sotto problemi:

- Dato un insieme M di messaggi anonimizzabili, come trovare la minima area di oscuramento per tutti i messaggi.
- Preso un messaggio m_s in un insieme di messaggi S, come trovare un insieme M contenente m_s e il gruppo di messaggi anonimizzabili con m_s.

Per lo sviluppo della prima parte del problema viene definito un elenco di $M \subset S$ messaggi che è possibile raggruppare e anonimizzare insieme; in altre parole, vengono assegnati alla stessa area di oscuramento. La migliore strategia per trovare l'area minima di oscuramento per tutti gli elementi dentro M è di utilizzare il rettangolo di delimitazione minimo comune a tutti i messaggi. Questo rettangolo, definito $B_m(M)$, prende il nome di *minimum spatio-temporal cloaking box* ed è riferito all'insieme M; mentre con il termine $L(m_s)$ si indicano le coordinate spazio-temporali precise del messaggio prima dell'anonimizzazione (x, y, t). Per la seconda parte del problema è necessario modellare i vincoli di anonimizzazione di tutti i messaggi contenuti in S in un *Constraint graph*, così definito:

Definizione 3 *Preso $G(S, E)$ un grafo non orientato, dove S è l'insieme dei nodi ognuno dei quali rappresenta un messaggio ricevuto, ed E è l'elenco dei collegamenti degli archi. Nel grafo esiste un arco $e = (m_{si}, m_{sj}) \in E$ tra due nodi m_{si} e m_{sj} se e solo se valgono le seguenti condizioni: 1) $L(m_{si}) \in B_{cn}(m_{sj})$ 2) $L(m_{sj}) \in B_{cn}(m_{si})$ 3) $u_{id}.m_{si} \neq u_{id}.m_{sj}$ Si definisce questo grafo Constraint graph*

Insieme le condizioni 1, 2 e 3 fanno in modo che i messaggi siano connessi in un grafo se e solo se: provengono da due utenti con identificativo diverso u_{id} e le loro coordinate spazio-temporali sono contenute nelle reciproche aree, limitate dalle loro tolleranze.
Definito il concetto di *Contraint graph* si può introdurre il teorema *CliqueCloak* nel seguente modo:

Teorema *CliqueCloak* 1 *Preso $G(S, E)$ un constraint graph $M = m_{s1}, m_{s2}, ..., m_{sl} \subset S$ e $\forall_{1 \leq i \leq l}$, $m_{ti} = \langle u_{id}.m_{si}, r_{no}.m_{si}, B_m(M), C.m_{si} \rangle$. Si afferma che $\forall_{1 \leq i \leq l}, m_{ti}$ è una valida perturbazione $k - anonymous$ di m_{si}, $m_{ti} = R(m_s)$, se solo se l'insieme M dei messaggi forma un $l - clique$ nel constraint graph $G(S, E)$ tale che $\forall_{1 \leq i \leq l}$, $k_{m_s} \leq l$*

Per spiegare meglio il teorema si utilizza l'esempio mostrato in Figura 4.1 che considera quattro messaggi (m_1,m_2,m_3,m_4), provenienti da utenti diversi. Ipoteticamente, all'inizio sono presenti all'interno del sistema solo i primi tre messaggi. Si costruisce quindi il *constraint graph* (c), che mostra come questi messaggi possano essere collegati tra loro. Si osserva che le posizioni spaziali dei messaggi m_1 e m_2 sono contenute l'una nello spazio dell'altra e risultano, quindi, essere connesse nel *constraint graph*; al contrario, m_3 è scollegato. Si osserva che $k_{m_2} = 3$ mentre m_1 e m_2 formano un $2 - clique$. Dal teorema si ricava che il numero di elementi che costituiscono un $l - clique$ deve avere come valore minimo il più alto valore del livello di privacy presente tra i messaggi che si considerano, in questo caso 3. Di conseguenza si osserva che non viene soddisfatto il teorema *CliqueCloak* e quindi i messaggi m_1 e m_2 non possono essere anonimizzati insieme e vengono rimossi. Si costruisce poi un ulteriore ipotetico *constraint graph*, mostrato in (d), aggiungendo il messaggio m_4. Con l'arrivo del nuovo messaggio l'unico modo per soddisfare il teorema è prendere un $3 - clique$ costituito da $\{m_1, m_2, m_4\}$. Così facendo l risulta essere pari al massimo valore di k. Ora è possibile costruire la minima area di oscuramento, mostrata in figura con la linea continua, escludendo il messaggio m_3. Quest'ultimo potrà essere utilizzato in seguito per trovare un'altra area o eliminato se non risulta possibile anonimizzarlo con altri messaggi.

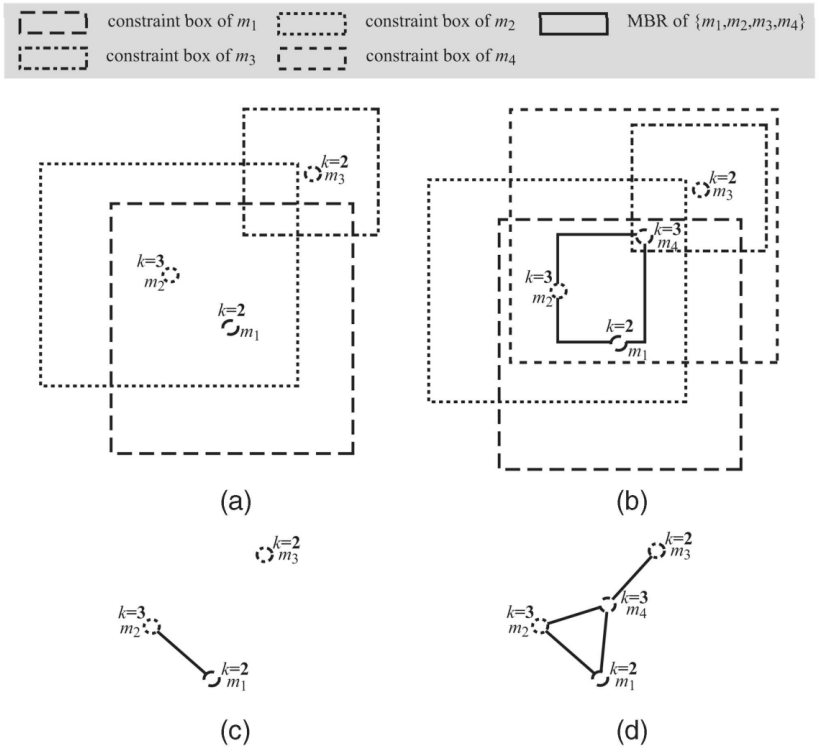


FIGURA 4.1: Esempio spiegazione *Clique theorem*, da [19].

4.1.4 Risultati ottenuti

Utilizzando alcune delle metriche definite nel capitolo tre si procede con la discussione delle performance dell'algoritmo proposto. La valutazione è stata eseguita utilizzando le metriche *success rate* (3.15), *relative temporal resolution* (3.17), *relative sparial resolution* (3.16) e *relative anonymity level* (3.1). Per prima cosa si precisa che con local-k si intende l'algoritmo base utilizzato, riferito al teorema *CliqueCloak*. Con nbr-k, invece, si indica un algoritmo di miglioramento, implementato all'interno del local-k, che aumenta le performance ottenute: un numero maggiore di messaggi viene anonimizzato in una sola volta, creando gruppi più numerosi di messaggi anonimizzati insieme. Infine con algoritmo online si intende un algoritmo che lavora in maniera incrementale, quindi appena riceve un elemento in input ne genera uno in output.

La Figura 4.2 mostra il *success rate* ottenuto utilizzando l'algoritmo local-k, nbr-k e con un algoritmo online. Per gli algoritmi sopra nominati sono stati raccolti i dati e suddivisi in base al livello di privacy richiesto dall'utente, in un range che va da 2 a 5. Nel grafico le barre più scure raffigurano i tassi di successo ottenuti con l'algoritmo nbr-k mentre quelle più chiare sono riferite ai risultati dati dall'algoritmo local-k. Ogni tasso è calcolato utilizzando uno specifico valore di k, ad esclusione delle barre all'estrema sinistra del grafico, che rappresentano il tasso di successo medio per tutti i messaggi. Con la barra bianca più sottile viene rappresentata la percentuale di messaggi non anonimizzati se si utilizza l'algoritmo online.

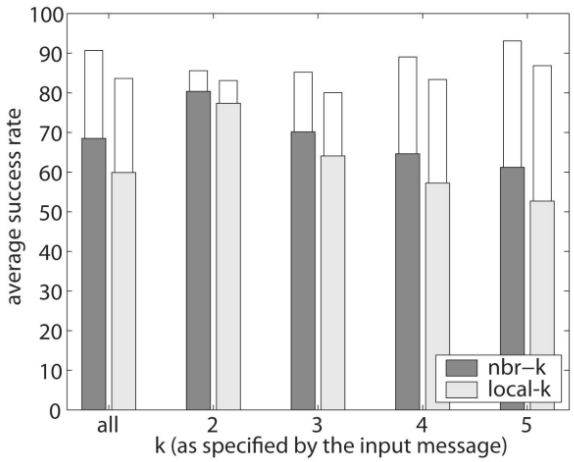


FIGURA 4.2: Success Rate con diversi valori di k, da [19].

Si possono fare le seguenti osservazioni:

- L'approccio nbr-k procura un tasso di successo medio più elevato del 15% rispetto all'approccio local-k.
- Il miglior valore medio ottenuto osservando tutti i livelli di privacy k è circa del 70%, e si ottiene con l'utilizzo del nbr-k. Il restante 30% dei messaggi viene eliminato perchè non rispetta il teorema *CliqueCloak*.
- I messaggi con un più alto valore di k sono i più difficili da anonimizzare. Si può notare che la differenza in percentuale tra l'anonimizzare con $k = 2$ e con $k = 5$ è del 30%.
- Considerando i valori rappresentati dalle barre bianche, non è possibile realizzare un algoritmo online ottimale a causa della necessità di quest'ultimo di avere accesso a delle informazioni future, che non possono essere note a priori in questo contesto.

Di seguito, nella Figura 4.3 è mostrata, confrontando le performance di local-k e nbr-k, la media della relative anonymity level (R_l) ottenuta in base al livello di privacy k voluto. Prendendo i valori ottenuti per l'algoritmo nbr-k si osserva che se il messaggio richiede un livello $k = 2$ si ottiene un valore R_l di 1.7, che implica l'anonimizzazione dei messaggi con un livello $k = 3.4$ da parte dell'algoritmo. Per lo stesso livello di privacy, nel caso dell'algoritmo local-k il valore di R_l risulta essere di 1.4; questo comporta che l'anonimizzazione è avvenuta con un livello $k = 2.8$. La differenza di valori che riguardano questi due approcci si attenua mano a mano che il livello di privacy viene aumentato. Infatti, se si prende $k = 5$ si ottiene $R_l = 1$ per entrambi gli approcci, ossia il livello richiesto e quello effettivamente usato per anonimizzare risultano uguali. Con i dati ricavati da questo grafico si conclude che più il livello di privacy richiesto risulta elevato e più il livello con il quale si anonimizza attraverso l'algoritmo diminuisce.

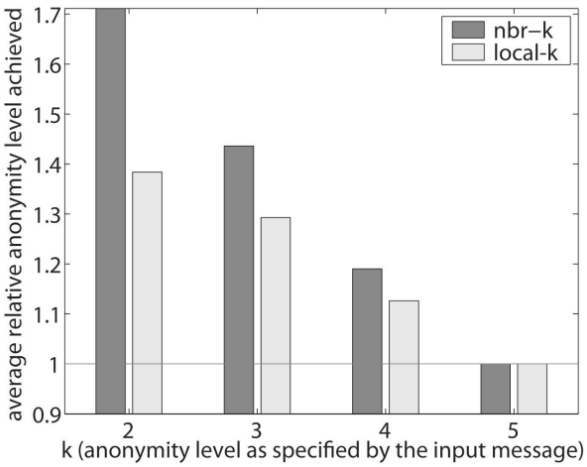


FIGURA 4.3: Relative anonymity level con diversi valori di k, da [19].

Di seguito vengono analizzati i rispettivi valori di *Spatial resolution* e *Temporal resolution*, in corrispondenza della frequenza di messaggi che presentano quei determinati risultati. La tolleranza presa per la *Temporal resolution* è di 30s mentre per la *Spatial resolution* è di 100m.

Le osservazioni che si possono ricavare dalla Figura 4.4a per la temporal resolution sono:

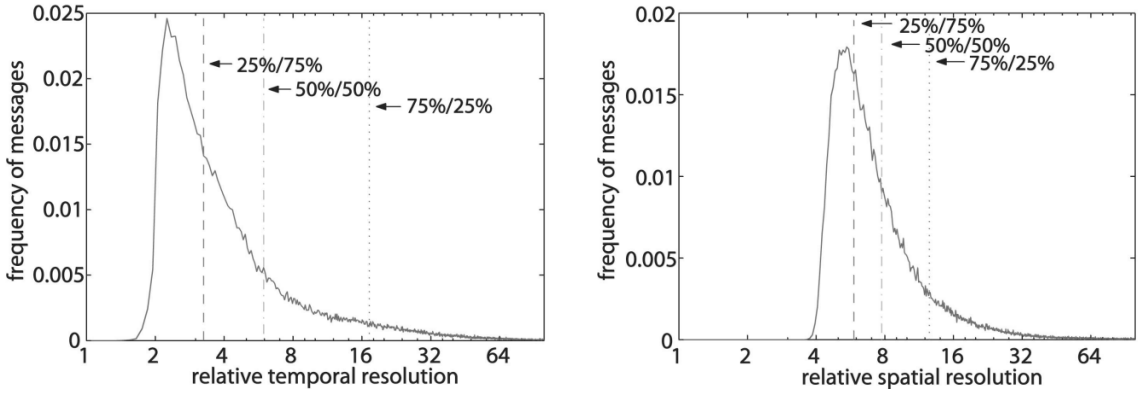
- Nel 75% dei casi risulta essere circa maggiore di 3.25, con un'accuratezza temporale di un valore circa minore di 10s.
- Per il 50% dei casi risulta avere un valore circa maggiore di 5.95.
- Per il 25% dei casi risulta avere un valore circa maggiore di 17.25.

Dato che il valore della risoluzione temporale aumenta più il ritardo inserito risulta breve, si conferma che le performance, per quanto riguarda la risoluzione temporale, sono migliori rispetto al caso peggiore definito dalla tolleranza.

Le osservazioni che è possibile ricavare dalla Figura 4.4b per la Spatial resolution sono:

- nel 75% dei casi si ha un valore circa maggiore di 5.85, ricavando un'accuratezza spaziale che ha valore circa minore di 18m.
- Nel 50% dei casi si ottiene un valore circa maggiore di 7.75
- Nel 25% dei casi si ottiene un valore circa maggiore di 12.55

Dato che se il valore della risoluzione spaziale aumenta significa che l'area utilizzata è più piccola, si conferma che le performance, per quanto riguarda la risoluzione spaziale, sono migliori rispetto al caso peggiore definito dalla tolleranza.



(a) *Temporal resolution* in base alla distribuzione della frequenza dei messaggi

(b) *Spatial resolution* in base alla distribuzione della frequenza dei messaggi

FIGURA 4.4: Confronto tra *Spatial* e *Temporal resolution*, da [19]

In conclusione, in [19] si è proposta la personalizzazione di un modello, basato sulla *k-anonymity*, che garantisce all'utente un grado di libertà più alto potendo decidere il livello di privacy k, la *Spatial* e la *Temporal tolerance*. Inoltre in [19], è stato implementato un metodo efficiente per la perturbazione dei messaggi, basato sul teorema *CliqueCloak*. A partire dall'osservazione dei dati sperimentali, ricavati ed analizzati in precedenza, è stato possibile confermare la validità di questo metodo.

4.2 Meccanismi di generazione dummy trajectories

La necessità di comunicare la traiettoria di un utente ad un LBS deriva dall'esigenza di ottenere un servizio che dipenda dal percorso che l'utente sta percorrendo in quel momento, per questo uno dei modi sviluppati per proteggere l'utente è la generazione di traiettorie fittizie che confondano un eventuale attaccante. Il problema della generazione di locazioni fittizie *dummy location*, presentata nel capitolo due, è da sempre oggetto di studi perché presenta molti approcci possibili. Uno di questi si basa sullo studio della posizione reale dell'utente e della sua traiettoria; tuttavia, non tiene conto di come le traiettorie percorse da un utente possano seguire dei pattern precisi, o comunque più probabili. Questo aspetto viene affrontato nell'articolo [20].

Prendendo la Figura 4.5 come esempio, è possibile analizzare quattro casi interessanti, in cui con la linea continua T è indicata la traiettoria dell'utente mentre con le linee tratteggiate (d1,d2) si indicano le traiettorie fittizie. Partendo dalla 4.5a, si nota la presenza di un problema: le traiettorie d1 e d2 risultano essere troppo diverse da un normale comportamento umano; pertanto, verrebbero subito scartate da un eventuale aggressore. Di conseguenza, per ridurre la facilità di identificazione è necessario che le traiettorie fittizie siano credibili. Un'ulteriore elemento da prendere in considerazione è la possibilità per gli attaccanti di collezionare dei $long - term\ movement\ pattern$, che potranno essere utilizzati per filtrare le traiettorie e trovare quella dell'utente. Se si confrontano gli esempi (c) e (b) e si ipotizza che siano riferiti allo stesso utente ma in momenti diversi, un eventuale attaccante potrebbe dedurre che T sia la traiettoria reale. Pertanto, risulta importante che le traiettorie generate non solo riproducano il movimento di un utente, ma siano coerenti nel tempo; questo perché, se ad un certo punto eseguono un movimento inconsueto, c'è maggior rischio che vengano

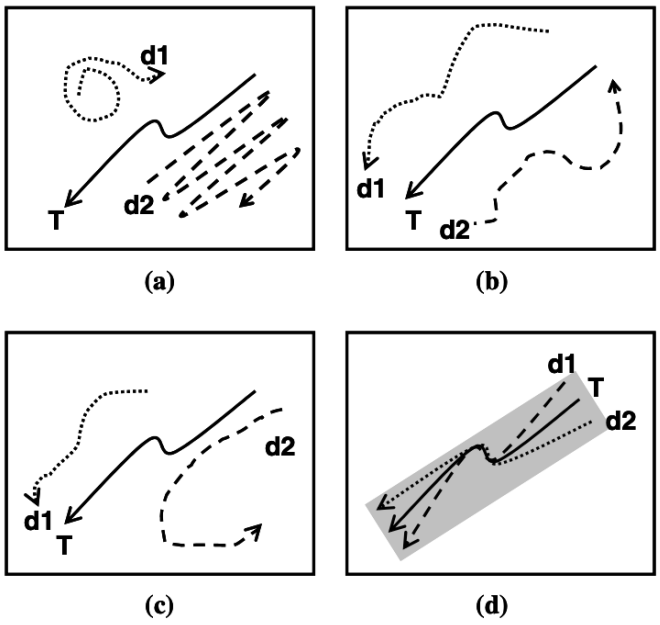


FIGURA 4.5: Esempi Traiettorie Dummy, da [20].

scartate. Infine, il caso (d) mostra un approccio differente, in cui è presente un'intersezione delle traiettorie che, al tempo stesso, risultano essere troppo simili tra loro, provocando una diminuzione della privacy. Nell'articolo [20], preso in esame, si sono studiati dei pattern che presentino due caratteristiche importanti: la prima riguarda l'aumento delle intersezioni tra le traiettorie e la seconda un aumento della distanza tra esse. Combinando questi due elementi, è possibile diminuire le probabilità dell'attaccante di indovinare il percorso corretto.

4.2.1 Generazione di dummy location con patterns

Inizialmente, si definisce un profilo di privacy di un utente, specificando tre parametri definiti nel capitolo tre e ripresi di seguito: Short-term Disclosure (SD) 3.2, Long-term Disclosure (LD) 3.3 e Distance deviation (dst) 3.18. Questi parametri vengono decisi dall'utente e vengono impostati come valori minimi accettabili per l'utilizzo di un servizio. Per semplificare la spiegazione viene ipotizzato che l'utente si possa muovere liberamente su una mappa divisa in una griglia di celle quadrate, identificate con coordinate (x,y). Di seguito vengono presentati due schemi, chiamati *random pattern* e *rotation pattern*, per generare le traiettorie fittizie.

Random pattern:

Per la realizzazione di questo schema, vengono inizialmente fornite le coordinate di partenza e di arrivo, di cui l'utente reale è a conoscenza. Queste posizioni serviranno per generare delle traiettorie che siano coerenti dopo un lungo periodo di osservazione. Secondo questo metodo, il movimento dalla posizione iniziale a quella finale, eseguito dagli utenti fittizi, sarà sempre randomico. Tuttavia, questo schema risulta essere troppo semplice perché non contiene tanti fattori, per esempio non tiene in considerazione fattori come la distanza di deviazione; quindi, l'utilizzo di un numero maggiore di utenti fittizi è l'unico modo per aumentare la privacy richiesta.

Rotation pattern:

Il secondo schema, al contrario, risulta essere più complesso del precedente, assicurando anche una maggiore privacy. L'idea principale è l'aggiunta di intersezioni tra traiettoria reale dell'utente e quelle fittizie. Data la traiettoria dell'utente, vengono generate delle nuove traiettorie, per gli utenti fittizi, ruotando la traiettoria originale. Ricordando che devono essere soddisfatte le tre richieste definite nel profilo dell'utente (SD, LD, dst), l'approccio usato prevede, inizialmente, di derivare lo spazio delle soluzioni che soddisfano il requisito della *dst*. In seguito, all'interno dello spazio delle soluzioni si ottengono gli altri due valori, SD e LD attraverso le formule 3.2 e 3.3. Ottenuti tutti i dati, vengono selezionate come traiettorie fittizie quelle che presentano le richieste con i valori più bassi. Al fine di ricavare lo spazio delle soluzioni per la *dst*, è necessario considerare due elementi, definiti come *rotation angle* e *rotation point*.

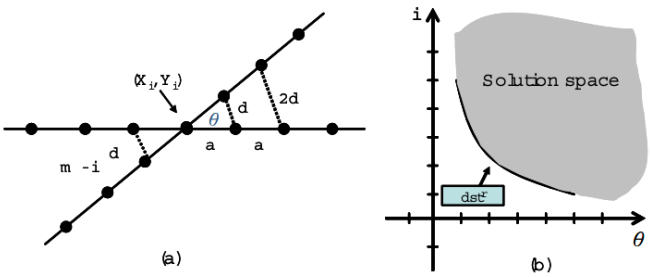


FIGURA 4.6: Esempio traiettoria reale e fittizia, da [20].

Per semplificare la derivazione della *dst*, si assume di avere due traiettorie, una reale e una fittizia, come in Figura 4.6. Si possono definire i seguenti parametri: *rotation point*, chiamato (X_i, Y_i), indica il punto in cui la traiettoria reale e fittizia si intersecano all'istante i-esimo; La d rappresenta la distanza tra la locazione reale e quella fittizia all'istante $(i+1)$; $|T|$ è il numero di istanti totali presi in considerazione; infine, il *rotation angle* è l'angolo θ. Per il teorema del coseno, si ha che: è possibile ricavare d vista l'uguaglianza tra i due lati che comprendono l'angolo θ come:

$$d = \sqrt{2}|a|\sqrt{1 - cos(\theta)} \tag{4.1}$$

Da cui si deriva che la distanza di deviazione tra due traiettorie è definita come:

$$dst^r = \sqrt{2}|a|\sqrt{1 - cos(\theta)} \left(\sum_{j=0}^{i} j + \sum_{j=0}^{|T|-i} j \right) \tag{4.2}$$

La formula 4.2 denota la dipendenza dal *rotation angle* e dal *rotation point* e la loro importanza nel ricavare dst^r. Lo schema delle traiettorie inizia non presentando alcun utente fittizio: essi, infatti, verranno aggiunti uno per volta in base alle necessità, derivate dai vincoli sul profilo dell'utente. Si definisce il valore di dst_n come la distanza di deviazione delle n traiettorie fittizie, presenti nello schema, fino all'istante i considerato. Il ragionamento

usato per l'inserimento di utenti fittizi nello schema delle traiettorie è il seguente: si fanno delle ipotesi sul numero della fascia oraria i in cui si potrebbe inserire un *rotation point* e su quanto l'angolo θ potrebbe valere. Quindi, per ogni ipotesi, si verifica se:

$$\frac{n}{n+1}dst_n + \frac{1}{n+1}dst^r \geq dst \tag{4.3}$$

Di conseguenza, è possibile riscrivere la formula nel modo seguente:

$$dst^r \geq (n+1)dst - n(dst_n) \tag{4.4}$$

Una volta che l'equazione è verificata, si esegue il calcolo delle SD 3.2 e LD 3.3, provvisorie che si avrebbero con i dati ipotizzati. Viene scelta l'ipotesi più soddisfacente, ossia che quindi presenta i dati di SD e LD più vicini a quelli forniti dall'utente. Se i valori ottenuti risultano superiori a quelli voluti, sarà necessario inserire un'ulteriore locazione fittizia per diminuirli, aggiornare il valore di dst_n e ripetere la procedura fino al raggiungimento dell'obiettivo. Per comprendere meglio il metodo, si prenda in considerazione il seguente esempio numerico. Si consideri il profilo dell'utente rappresentato dai seguenti valori: $dst = 2.1$, $SD = 40\%$ e $LD = 10\%$. Inizialmente, non sono presenti utenti fittizi ($n = 0$), quindi, $dst_n = 0$. Vengono fatte delle ipotesi e sono riportate in Figura 4.7a quelle relative alla verifica della validità:

θ	i	SD	LD
120	5	56.25%	25%*
50	3	56.25%	25%
180	1	56.25%	25%

(a). Solution space when n=0

θ	i	SD	LD
170	8	37.5%	16.67%
120	7	37.5%	12.5%
80	6	39.6%	8.33%*

(b). Solution space when n=1

FIGURA 4.7: Dati ipotesi numeriche, da [20].

Successivamente, in seguito si calcolano i valori SD e LD per tutte le ipotesi e, a causa dell'uguaglianza tra tutti i valori, si decide di prenderne uno a piacere. In questo caso, nella *rotation point* $i = 5$ avverrà un incrocio con una traiettoria fittizia ruotata di $\theta = 120°$. Il valore dst_n va ora aggiornato perché n = 1: si eseguono gli stessi passaggi svolti nel punto precedente, utilizzando le ipotesi in Figura 4.7b. Dalla figura si ricavano i valori di SD e LD che soddisfano l'obiettivo, corrispondenti all'ipotesi di rotation angle 80° e rotation point 6. Una volta raggiunto l'obiettivo, il processo di anonimizzazione è completato e si ferma la generazione di utenti fittizi, come illustrato in Figura 4.8.

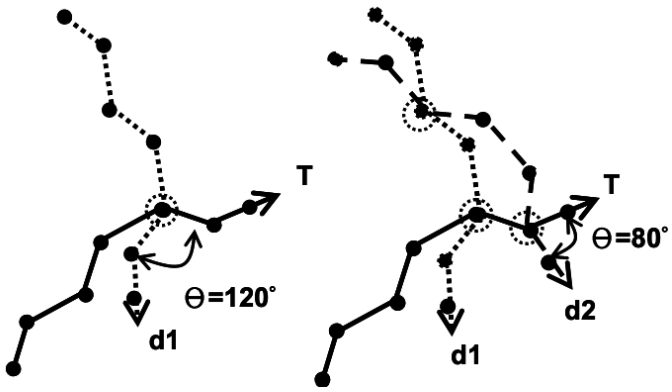


FIGURA 4.8: Esempio numerico, da [20].

4.2.2 Risultati ottenuti

I risultati sono ottenuti impostando come profilo di privacy i seguenti valori: $SD = 20\%$, $LD = 10\%$ e $dst = 2.8$. Nella Figura 4.9 vengono messi a confronto il *rotation pattern* proposto e uno schema realizzato nell'articolo [24], che si basa su una generazione randomica di posizioni fittizie. In 4.9a è possibile vedere: sull'asse x il numero di giorni in cui sono state fatte le misurazioni, mentre sull'asse y si può osservare la percentuale di SD misurata in un determinato giorno. Nel caso dell'approcio con *rotation pattern*, la percentuale di SD rimane sempre costante al 20%, anche per un'osservazione a lungo termine di 12 giorni. Al contrario, utilizzando lo schema randomizzato, verso la fine del periodo di osservazione l'SD raggiunge il 100%; in quest'ultimo caso, la traiettoria dell'utente sarà completamente scoperta dall'attaccante. Passando alla figura 4.9b, il grafico mostra il valore percentuale di LD che si ottiene per ciascun periodo di osservazione riportato sull'asse x. Qui si ripete il comportamento osservato per la SD. In particolare, la percentuale di LD, per il rotation pattern, mantiene andamento costante al 10%. Diversamente, se ci si riferisce allo schema randomizzato LD raggiunge il 100% verso la fine dell'osservazione.

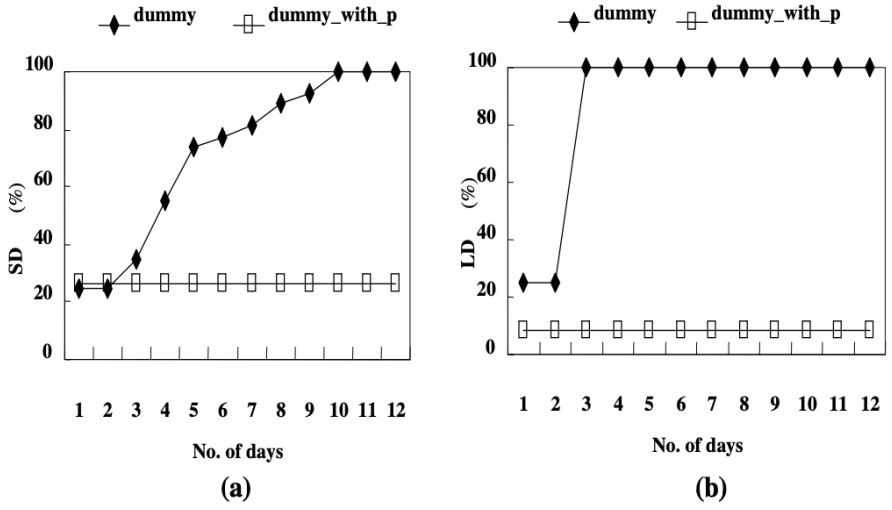


FIGURA 4.9: Comparazione di Dummy-based Schemes, da [20].

Entrambi i grafici confermano che il *rotation pattern* mantiene una migliore anonimizzazione dell'utente rispetto ad uno schema randomizzato; quindi, sotto il profilo della privacy *rotation pattern* risulta efficiente a lungo termine.
In figura 4.10 è presentato un confronto tra il *rotation pattern* e il *random pattern* che riporta il numero di locazioni fittizie necessarie a mantenere i valori di LD, SD e dst richiesti nel profilo dell'utente. Nei grafici è possibile osservare come sull'asse y di entrambi siano indicati il numero di locazioni fittizie mentre sull'asse x del grafico 4.10a sia indicata la percentuale SD mentre sul grafico 4.10b sia rappresentata la percentuale di LD. Prendendo il grafico 4.10a, con LD fissato al 50% e dst a 2.8, è possibile vedere che il numero di locazioni fittizie generate con i due approcci risulta molto simile, nonostante l'aumento della SD. Prendendo il grafico 4.10b, con SD fissato al 50% e dst a 2.8, è possibile vedere che il numero di locazioni fittizie generate con il metodo *rotation pattern* è inferiore a quelle ottenute con il *random pattern*.

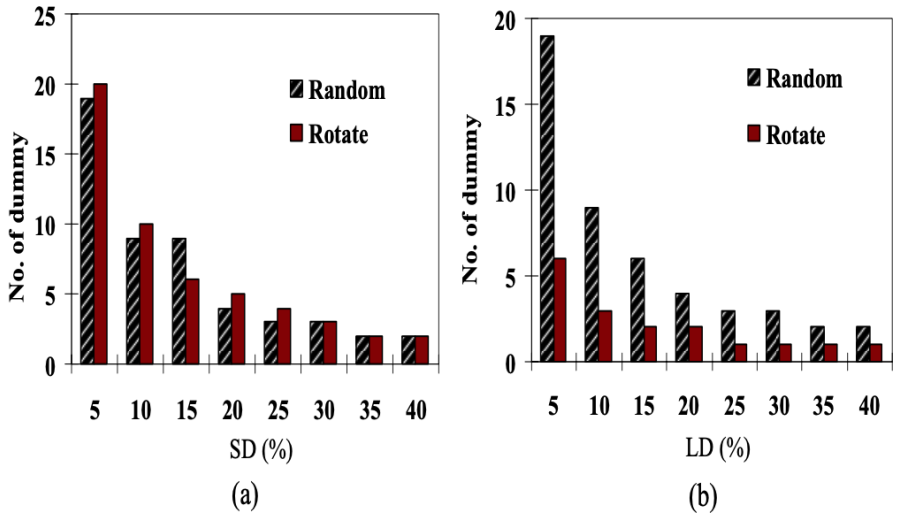


FIGURA 4.10: Numero di locazioni fittizie generate in base allo schema usato, da [20].

Dall'analisi dei grafici in Figura 4.10, si verifica la maggiore efficacia del *rotation pattern* rispetto al *random pattern*, perché una minore presenza di locazioni fittizie implica un minor tempo di elaborazione delle richieste da parte dell'LBS.

Capitolo 5

Applicazioni nel machine learning

5.1 Introduzione

La velocità con la quale si stanno evolvendo la raccolta, la trasmissione e la distribuzione dei dati mette a rischio sempre di più l'utente. Dunque, con lo sviluppo dell'intelligenza artificiale, si è cominciato ad esplorare la sua applicazione nel campo della location privacy. L'IA(intelligenza artificiale), tramite la costruzione di politiche di decisione rivolte agli utenti, supporta la protezione delle loro posizioni, garantendo la regolazione delle informazioni e la personalizzazione delle preferenze di privacy. Per analizzare più nel dettaglio l'applicazione dell'IA, in questa sezione verranno presentati due articoli; entrambi utilizzano il concetto di *machine learning* al fine di rendere più elastico il processo di protezione, adattandolo, quindi, all'utente.

5.2 Applicazione del Machine learning nel contesto dei social network

Nell'ambito dei social network il concetto di privacy è un tema delicato; infatti, a causa loro le posizioni degli utenti sono diventate maggiormente accessibili. L'articolo [25] classifica come gli utenti condividono la loro locazione in modo diretto e modo indiretto:

- Il modo diretto consiste nella condivisione volontaria tramite post o commenti da parte dell'utente.
- Il modo indiretto consiste nel ricavare la posizione dell'utente avendo a disposizione, per esempio, la sua distanza da un determinato posto.

Quindi, sulla base della tipologia di condivisione vengono formulati due tipi di attaccanti, *casual traking attacker* e *continuous traking attacker*:

- Il primo, applicabile nel caso di una condivisione diretta, ha sempre accesso allo storico delle informazioni sull'obiettivo necessarie a costruirne il profilo. L'attacco può essere attuato in qualsiasi momento e in maniera del tutto casuale.

- Il secondo, nel caso di una condivisione indiretta, realizza un tracciamento a lungo termine dell'obiettivo. In questo modo ottiene una serie di posizioni riguardanti l'utente, che genera un profilo di mobilità.

Per realizzare un algoritmo di difesa contro questi attaccanti è necessario per prima cosa definire degli algoritmi di attacco, in questo caso ne sono stati realizzati due, di cui uno utilizza un approccio di *machine learning*. Al fine di rendere i due algoritmi di attacco efficaci, gli autori dell'articolo hanno condotto un esperimento utilizzando due dataset. Uno costituito da un piccolo gruppo di utenti, usato per il primo attacco, mentre l'altro si basa su uno studio su larga scala, usato per il secondo attacco. In particolare, il primo dataset è il risultato di un esperimento durato 3 settimane, dove sono state collezionate le condivisioni dirette ed indirette di 30 partecipanti. Il secondo è stato ricavato attraverso il coinvolgimento di 22,843 utenti per un periodo di tempo di cinque mesi, nei quali, registrando le loro traiettorie, sono stati studiati gli accessi alla rete Wi-Fi, l'identificativo dell'utente, l'orario e il social network utilizzato.

5.2.1 Metodi di attacco

L'articolo descrive due tipologie di attacco. Il primo utilizza un algoritmo che tiene conto della più grande traccia comune tra gli utenti MCT(maximal common trace), dove con traccia si intente la più lunga sequenza di posizioni comuni tra due utenti. Il secondo, invece, sfrutta un algoritmo di machine learning il cui modello di apprendimento è collezionare più modelli di mobilità possibili.

Algoritmo Maximal Common Target:
Nel primo approccio proposto, l'attaccante effettua le seguenti azioni:

- Colleziona le tracce di mobilità di diversi utenti nella stessa regione.
- Trova la più grande traccia comune tra uno specifico target, scelto dall'attaccante, e gli altri utenti.
- Calcola la similitudine tra il target e gli altri utenti per poi classificarli in base al loro punteggio di somiglianza.
- Deduce gli attributi demografici del target dai primi k utenti.

Algoritmo Machine Learning:
Il secondo algoritmo è costituito dai seguenti passaggi:

- Si inizia con la fase di apprendimento, in cui si utilizzano i dati raccolti nel dataset più grande. Si prendono tutte le locazioni raccolte e, se un utente in quel luogo ha condiviso la sua posizione, viene assegnato a tale locazione il valore 1, altrimenti viene assegnato 0.
- Una volta allenato il modello, l'algoritmo è usato per dedurre informazioni di altri utenti in base ai loro profili di localizzazione.

Con l'utilizzo di un campione di 4000 utenti, 2000 in numero uguale tra uomini e donne e altri 2000 distribuiti ugualmente tra laureandi triennali, magistrali e dottori, si è verificata l'efficacia di questi due approcci; in particolare, si è studiato il tasso di successo nel dedurre il genere e il livello di studio di ogni individuo preso in esame. Con il primo approccio si è

riscontrato che i tassi di successo nel determinare genere e livello di studio sono rispettivamente del 73% e del 78%; mentre, con il secondo algoritmo si ha, per gli stessi, il 63% ed il 65%. I risultati dimostrano che il primo approccio risulta essere più efficace del secondo sotto il profilo della qualità dei risultati. Tuttavia, con una grande mole di dati, l'algoritmo basato sul *Machine learning* è, in termini computazionali, 496 volte più veloce.

5.2.2 Metodo di difesa SmartMask

Per contrastare gli attacchi, si è sviluppato un metodo di controdifesa basato sul ML(Machine Learning). L'idea principale dell'algoritmo SmartMask è di bilanciare la protezione e l'utilità, assegnando un certo livello di privacy in base alle locazioni in cui l'utente si trova. L'intuizione deriva dalla maggior tendenza, da parte degli utenti, all'utilizzo dei social nei luoghi pubblici e da una minore nei luoghi più privati, come a casa o a lavoro. L'implementazione del modello si basa su quattro fasi:

- Context Generator: si collezionano le informazioni di locazione e si immagazzinano gli storici di mobilità di un database locale. Quindi, si estrae il profilo di localizzazione dell'utente dai dataset. Tale profilo registra la frequenza di visita dell'utente, il periodo e la durata di permanenza in un luogo, al quale in seguito sarà assegnato un livello di privacy tra basso, medio o alto.
- Privacy-level Generation: si assegna un livello di privacy differente per ogni posizione geografica e app LBS. Per decidere il livello viene usata una struttura come quella mostrata in Figura 5.1.
- User specified interface: si fornisce un'interfaccia per gli utenti per permettere l'inserimento delle preferenze di privacy. L'utente può specificare: il livello di privacy da avere in ogni contesto geografico o app LBS che soddisfi le sue esigenze.
- Obfuscation Engine: questa fase si occupa dell'implementazione di una tecnica di offuscamento attraverso lo sviluppo di una funzione, ObfuscateLocation(). Essa utilizza come parametri il nome dell'applicazione a cui l'utente vuole accedere, la locazione originale dell'utente e il livello di privacy voluto.

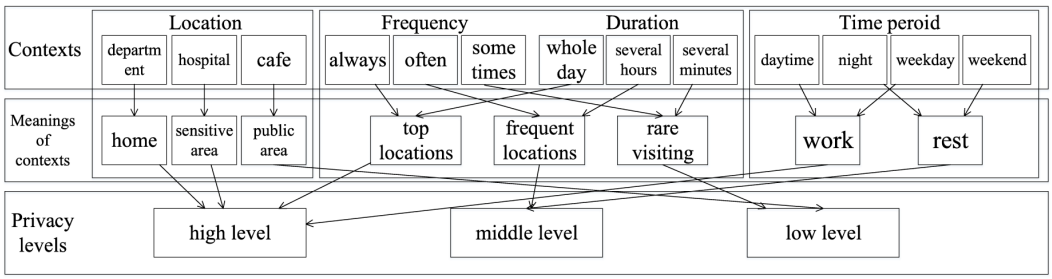


FIGURA 5.1: Attribuzione livelli di privacy, da [25].

Per valutare l'efficacia di SmartMask, si conduce un esperimento di durata 3 settimane, a cui partecipano 15 volontari di un campus che utilizza il sistema Android modificato, che incorpora SmartMask. Nelle prime due settimane, si raccolgono i dati di localizzazione di ogni utente per generare un profilo di localizzazione. Nell'ultima settimana, SmartMask comincia

a offuscare ogni posizione in base al profilo e alle preferenze degli utenti. I dati racconti nel dataset su larga scala vengono quindi usati per verificare l'efficacia della protezione ottenuta. Vengono provati vari raggi di oscuramento e viene valutata l'accuratezza di un attacco nel determinare i dati demografici riguardanti il genere e il livello di studio. Come mostrato in Figura 5.2 è possibile vedere che con l'aumento del raggio di offuscamento i valori diminuiscono di conseguenza.

obfs.radii(m)	100	300	500	1,000	no obfs.
Gender	0.57	0.56	0.54	0.47	0.73
Edu	0.62	0.61	0.57	0.51	0.76

FIGURA 5.2: Risultati efficacia modello SmartMask, da [25].

Si conclude che l'approccio SmartMask aiuta a contrastare le violazioni che riguardano il tentativo di dedurre delle informazioni riguardati l'utente. SmartMask può imparare automaticamente e generare in maniera autonoma i livelli di privacy basati sul contesto in cui l'utente si trova. Tuttavia per migliorare il modello sarebbe necessaria l'acquisizione di dati su larga scala che amplierebbero la capacità del modello ad adattarsi a più situazioni in cui l'utente può trovarsi.

5.3 Applicazione del machine learning con l'utilizzo di sensori interni del dispositivo mobile

Lo sviluppo del GPS ha reso la condivisione della posizione sempre più facile e precisa. Tuttavia, il GPS non è efficiente sotto il profilo energetico del telefono, in particolar modo quando lo si usa in continuazione per condividere la posizione. Nell'articolo [26] si è cercato di affrontare questo problema utilizzando altri sensori presenti nel dispositivo mobile, per raggiungere lo stesso scopo. Per la realizzazione di questo studio sono stati presi 15 volontari a cui è stato chiesto, per un periodo di 25 giorni, di accendere il cellulare appena svegli e di spegnerlo quando sarebbero andati a dormire. Inoltre, in vari momenti della giornata, venivano inviati dei messaggi dove veniva chiesto all'utente se desiderava condividere la sua posizione attuale; una volta ottenuta la risposta, i dati raccolti con i sensori venivano salvati.

5.3.1 Features utilizzate nello studio

WiFi Scan

Da questo sensore possono essere ricavate diverse informazioni:

- Attraverso l'utilizzo degli AP(Access Point), può essere ricavata la distanza tra casa e il luogo di lavoro. Inoltre, è possibile determinare la frequenza di visita di un luogo.
- Le precedenti preferenze di privacy che l'utente aveva impostato in determinati punti di interesse può influenzare le impostazioni che vorrà avere in futuro.
- La distanza da altri punti di interesse in cui si è impostata una privacy più alta o più bassa. Per esempio, se si è vicini a un luogo come un ospedale, in cui la privacy è elevata, sarà difficile che nelle vicinanze la privacy voluta sia molto minore.

- Infine, è ricavabile anche l'entropia degli AP. Se essa risulta bassa, in questo caso minore di 2.5, allora si è in presenza di luoghi visitati con una certa frequenza, ad orari e giorni precisi in cui l'utente fa una certa azione. Se l'entropia è alta, quindi circa di 2.5, allora sono luoghi poco frequentati dall'utente, in cui non si riesce a trovare uno schema di visite preciso.

Acceleration Data

Con l'utilizzo di questo sensore è possibile ricavare se un utente è fermo o in movimento. In particolare, i valori presi in esame sono :

- La varianza, che misura l'intensità dei movimenti eseguiti dall'utente.
- La media, che misura la postura del dispositivo mobile.

Tramite questi valori si è infatti osservato che, se una persona rimane ferma per un lungo lasso di tempo, è meno propensa a condividere la sua posizione, quindi il livello di privacy risulterà più elevato. Al contrario, se i valori raccolti indicano che l'individuo si sta muovendo, probabilmente si troverà in un luogo pubblico, quindi risulterà essere più disponibile a condividere la sua posizione.

Sound

Un altro fattore importante è il suono ambientale, che ha permesso di ricavare ulteriori proprietà di una locazione. I valori acquisiti dal microfono del dispositivo mobile sono:

- La pressione del livello sonoro, presente nell'area.
- I coefficienti definiti nel MFCC (Mel-Frequency Cepstral Coeffiecient).
- La determinazione della conversazione di un utente con un individuo o dei rumori ambientali.

Si è ricavata una relazione tra condivisione della posizione e suono di un luogo. Se il suono risulta basso gli utenti sono propensi a non condividere la posizione, perché è possibile che siano in un luogo privato o in un ospedale. Se, invece, i suoni captati risultano elevati gli utenti si trovano, molto probabilmente, in un luogo pubblico insieme a molte altre persone, quindi sono meno rigidi sulla condivisione della loro posizione.

Wi-Fi Indoor Positioning

Questo meccanismo di localizzazione cerca di stimare le coordinate precise di un utente all'interno di un edificio usando la potenza del segnale, basandosi sulla tecnica delle impronte digitali. Un elenco di AP e la rispettiva potenza del segnale diventano un impronta digitale che risulta unica per quelle coordinate. Vengono ricavati i seguenti valori:

- La distanza dalla più vicina posizione rivelata da un altro utente.
- La distanza dalla più lontana posizione rivelata da un altro utente.
- La distanza tra una posizione rivelata e una oscurata.

Time

Ogni volta che avviene una collezione di dati e la relativa risposta, risulta importante registrare anche il periodo preciso in cui è avvenuta la misurazione. Il periodo che viene salvato riguarda l'ora e il giorno precisi della *disclose* (l'utente ha accettato di condividere la sua posizione) e *not-disclose* (l'utente si è rifiutato di condividere la sua posizione). Usando questi dati è possibile fare un confronto con dati precedentemente salvati e vedere se sono presenti schemi ricorrenti riguardanti gli utenti.

Location

Per confrontare le performance dei sensori interni e il GPS, prima si sono estratte le caratteristiche delle coordinate, prese con il Wi-Fi, e poi sono state usate per predire le preferenze di privacy dell'utente. Quindi, sono stati estratti i dati più simili tra GPS e Wi-Fi, come ad esempio la distanza da casa, dal lavoro e la frequenza di visita.

5.3.2 Spiegazione algoritmo machine learning

Dai sensori appena descritti si ottengono dei vettori di features e, per ognuno di essi, la rispettiva risposta data dall'utente riguardante la condivisione o meno della posizione. I dati acquisiti vengono utilizzati per allenare un modello di machine learning basato sulla classificazione in due classi, chiamate *disclose* e *not-disclose*. La procedura di costruzione del modello si divide in quattro fasi:

- Per ogni feature viene calcolato un *information gain*, che tiene conto di quanto la feature contribuisce alla classificazione: più è alto il valore, più la componente ha maggiore importanza nella classificazione.
- La definizione del gain permette, poi, di definire in maniera più precisa la distanza tra i vettori. Moltiplicando, infatti, ogni feature per il corrispondente gain, la distanza tra due vettori che hanno valori differenti, relativi a una feature distinguibile, aumenta. In altre parole la distanza tra vettori di classi diverse diventa elevata.
- Per non incorrere nel problema della *"curse of dimensionality"*, quando vengono calcolate le distanze tra i vettori, è necessario ridurre la dimensione di questi ultimi. Per farlo, si è scelto di scartare le feature con *information gain* che sono trascurabili, mantenendo quelle con valori elevati.
- Infine, viene utilizzato un modello a GMM (Gaussian Mixture Model) per modellare i vettori di feature per ogni classe. Utilizzando questa specifica distribuzione è possibile modellare le politiche di privacy dell'utente che variano in base alle situazioni dell'utente.

Pertanto, con il modello appena definito, per ogni vettore di caratteristiche si associa la classe con il valore di maximum likelihood maggiore, che rappresenta la sua probabile classe di appartenenza.

Successivamente, per la fase di valutazione, è stata utilizzata la *F-measure* [26] e sono state testate numerose combinazioni di tutti i sensori coinvolti nello studio, al fine di evitare di utilizzarli tutti contemporaneamente che, altrimenti, comporterebbe un alto consumo energetico. Dalla figura 5.3, si può notare che la combinazione Sound(Mic), WiFi e Time

risulta la combinazione più accurata, per l'acquisizione di informazioni relative all'utente, se confrontata con i risultati che si avrebbero utilizzando il solo GPS.

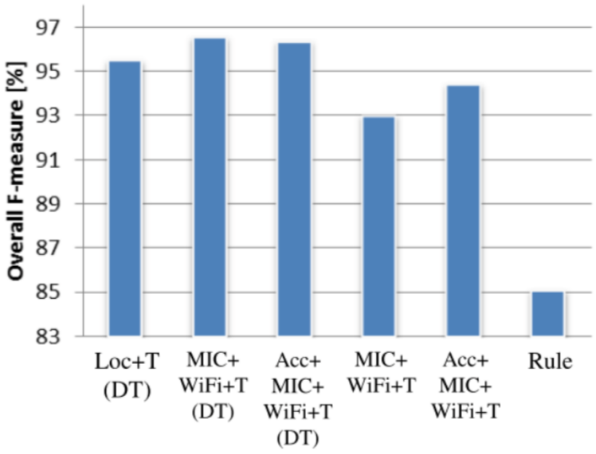


FIGURA 5.3: Risultati ottenuti, da [26].

Capitolo 6

Conclusioni

In questa tesi si è descritto, in primo luogo, la struttura di un LBS e come si sviluppa lo scambio di informazioni tra un utente e un server LBS che fornisce il servizio. E' stato evidenziato come l'utilizzo di queste applicazioni comporti dei rischi perché, se la posizione dell'utente risulta essere scoperta, è possibile costruire un profilo dell'individuo, quindi ricavare informazioni sensibili.

Nel secondo capitolo sono stati presentati dei metodi su cui basare la realizzazione dell'oscuramento di una posizione e, così facendo, preservare la privacy di un utente. Si è visto come gli approcci hanno in comune l'obiettivo di nascondere la posizione quanto basta per confondere un eventuale attaccante, ma non abbastanza da non poter utilizzare le LBS. Questo "trade-off" non è facile da trovare e, per misurarlo, sono state elaborate diverse metriche.

Sebbene non esista ancora uno standard uniforme per la valutazione di tutti i meccanismi, nel terzo capitolo sono state raccolte le metriche più comuni. Sono state divise in due tipologie, una che misura la privacy e l'altra che misura la qualità del servizio.

Nel capitolo quattro sono stati spiegati due algoritmi relativi ai metodi *cloaking* e *dummy location*. Sono stati analizzati e per la valutazione delle prestazioni sono state usate alcune metriche del capitolo due. Dall'analisi degli algoritmi si evince la necessità di inserire entrambe le tipologie di metriche di valutazione. Per entrambi gli algoritmi descritti si sono ottenuti buoni risultati, in accordo con le metriche usate, quindi se ne conferma la validità e la possibilità di applicare queste metriche per valutare altri algoritmi. La ricerca del trade-off tra privacy e qualità del servizio continua ad essere oggetto di studio e lo sviluppo del *machine learning* ha aperto la strada ad un nuovo approccio al problema.

L'utilizzo del *machine learning* si sta espandendo rapidamente, soprattutto grazie alla sua capacità di gestire grandi quantità di dati; tuttavia, l'apprendimento automatico presenta ancora dei limiti. Una delle questioni più importanti riguarda la difficoltà nel definire e misurare il livello di privacy, poiché l'impatto delle conoscenze in possesso degli aggressori è difficile da valutare. Inoltre, anche la reperibilità dei dati risulta essere un problema: infatti, è necessario introdurre un numero sufficiente di campioni che simulino gli attacchi per allenare efficacemente il modello a difendere l'utente. Infine, è necessario un approfondimento sul *training* dell'algoritmo per quanto riguarda l'adattamento della privacy dell'utente, in base alla posizione. Questo perché non è ancora stato creato un modello che sia in grado di gestire qualunque tipo di situazione in cui l'utente si trova. Nonostante questi problemi, l'approccio

basato sul *machine learning* è risultato essere il più promettente per la sua adattabilità. Un possibile sviluppo futuro di questa tesi prevede lo studio delle metriche descritte applicate a problemi reali, per capire se è possibile ricavare un metodo standard per qualsiasi situazione. Una volta trovato un metodo generalizzato, la verifica dell'efficacia degli algoritmi risulterà più immediata e veloce.

Bibliografia

[1] H. Jiang, P. Z. Jie Li, Z. X. Fanzi Zeng e A. Iyengar, «Location Privacy-preserving Mechanisms in Location-based Services: A Comprehensive Survey,» *IEEE Signal Processing Magazine*, 2020.

[2] T. F. Jiangshui Zhang Yixin Hua, «The Embedded GIS Technology in LBS,» *IEEE Signal Processing Magazine*, 2010.

[3] N. Eddy, «Location-based applications popular, despite privacy concerns: ISACA. eWEEK,» *https://www.eweek.com/mobile/location-based-applications-popular-despite-privacy-concerns-isaca*, 2012.

[4] X. Wang, J. L. Pengrou Shi1, F. Y. Yingxue Yang e J. W. Hongnian Yu, «Privacy-preserving Mechanisms of Continuous Location Queries Based on LBS: A Comprehensive Survey,» *IEEE Signal Processing Magazine*, 2022.

[5] P. Samarati e L. Sweeney, «Protecting Privacy When Disclosing Information: k-anonymity and Its Enforcement through Generalization and Suppression,» *Technical Report. SRI International.*, 1998.

[6] B. Gedik e L. Liu, «Location Privacy in Mobile Systems: A Personalized Anonymization Model,» *IEEE Signal Processing Magazine*, 2005.

[7] T. Xu e Y. Cai, «Exploring Historical Location Data for Anonymity Preservation in Location-Based Services,» *IEEE Signal Processing Magazine*, 2008.

[8] A. Machanavajjhala, D. K. Johannes Gehrke e M. Venkitasubramaniam, «L-diversity: Privacy beyond k-anonymity,» *IEEE Signal Processing Magazine*, 2006.

[9] N. Li, T. Li e S. Venkatasubramanian, «t-closeness: Privacy beyond k-anonymity and l-diversity,» *IEEE Signal Processing Magazine*, 2007.

[10] J. X. Zhen Xiao e X. Meng, «p-Sensitivity: A semantic privacy-protection model for location- based services,» *In Proceedings of the International Conference on Mobile Data Management Workshops*, 2008.

[11] X. Xiao e Y. Tao, «M-invariance: Towards privacy preserving re-publication of dynamic datasets,» *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2007.

[12] C. Dwork, «Differential privacy,» *In Proceedings of the International Colloquium on Automata, Languages and Programming. 1–12.*, 2006.

[13] M. E. Andrés, K. C. Nicolás Emilio Bordenabe e C. Palamidessi, «Geo- indistinguishability: Differential privacy for location-based systems,» *In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security 901-914*, 2013.

[14] C. P. Konstantinos Chatzikokolakis e M. Stronati, «A predictive differentially private mechanism for location privacy,» *CoRR 1311, 4008*, 2013.

[15] C. P. Konstantinos Chatzikokolakis e M. Stronati, «Constructing elastic distinguishability metrics for location privacy,» *Proc. Priv. Enhanc. Technol. 2015, 2 (2015)*, 2015.

[16] A. R. Beresford e F. Stajano, «Mix zones: User privacy in location-aware services,» *In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops. 127–131*, 2004.

[17] J. Freudiger, M. F. Maxim Raya, P. Papadimitratos e J.-P. Hubaux, «Mix-zones for location privacy in vehicular networks,» *In Proceedings of the ACM International Workshop on Wireless Networking for Intelligent Transportation Systems*, 2007.

[18] J. Meyerowitz e R. R. Choudhury, «Hiding stars with fireworks: Location privacy through cam- ouflage,» *In Proceedings of the International Conference on Mobile Computing and Networking*, 2009.

[19] B. Gedik, «Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms,» *IEEE Signal Processing Magazine*, 2008.

[20] T.-H. Y. W.-C. Peng e W.-C. Lee, «Protecting Moving Trajectories with Dummies,» *IEEE Signal Processing Magazine*, 2007.

[21] R. Shokri, C. T. George Theodorakopoulos e J.-Y. L. B. Jean-Pierre Hubaux†, «Protecting Location Privacy: Optimal Strategy against Localization Attacks,» 2007, in `http://carmelatroncoso.com/papers/Shokri-CCS2012.pdf`.

[22] B. Palanisamy, «Attack-Resilient Mix-zones over Road Networks: Architecture and Algorithms,» *IEEE Signal Processing Magazine*, 2015.

[23] B. Hoh e M. Gruteser, «Protecting Location Privacy Through Path Confusion,» *IEEE Signal Processing Magazine*, 2005.

[24] H. Kido, Y. Yanagisawa e T. Satoh, «An Anonymous Communication Technique using Dummies for Location-based Services,» *In Proc. of the Second International Conference on Pervasive Services (ICPS), pages 88–97*, 2005.

[25] H. Li, S. D. H. Zhu, X. Liang e X. S. Shen, «Privacy leakage of location sharing in mobile social networks: Attacks and defense,» *In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*, 2016.

[26] N. Y. T. Maekawa e Y. Sakurai, «How well can a user's location privacy preferences be determined without using GPS location data?» *IEEE Trans. Emerg. Topics Comput., vol. 5, no. 4, pp. 526–539*, 2017.