

# Time Is Running Out

Assessing Temporal Privacy of Privacy Zones in Fitness Tracking Social Networks

**Wout DELEU**

Promotor: Prof. dr. ir. Stijn Volckaert

Begeleiders: Ing. Karel Dhondt,  
Ing. Alicia Andries  
Ing. Jonas Vinck

Masterproef ingediend tot het behalen van  
de graad van master of Science in de  
industriële wetenschappen: Elektronica/ICT  
Optie Smart Applications

Academiejaar 2022 - 2023

©Copyright KU Leuven

Deze masterproef is een examendocument dat niet werd gecorigeerd voor eventuele vastgestelde fouten.

Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, kan u zich richten tot KU Leuven Technologiecampus Gent, Gebroeders De Smetstraat 1, B-9000 Gent, +32 92 65 86 10 of via e-mail [iiw.gent@kuleuven.be](mailto:iiw.gent@kuleuven.be).

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

# Voorwoord

Ik had graag eerst en vooral mijn ouders bedankt voor het financieren van mijn studies, en de ondersteuning gekregen in de periode. Daarnaast had ik graag Karel Dhondt, Stijn Volckaert, Alicia Andries en Jonas Vinck bedankt voor hun hulp en ondersteuning tijdens het schrijven van deze scriptie. Daarnaast in het bijzonder had ik ook graag Thomas Gruyaert bedankt, die tijdens het werken aan zijn eigen thesis ook een enorm grote hulp was. Graag had ik ook enkele van mijn kotenoten en vrienden bedankt voor de nodige afleiding tijdens de stressvolle perioden gedurende het academiejaar. In het bijzonder Lennert Beele, Angelo Pattyn en Jakob Sabbe, die zelf ook aan hun thesis werkten! Ook Sam Boeve, voor de erg hulpvolle adviezen gedurende het proces. Ook als laatste een bijzondere vermelding voor mijn ouders en broer, die mij ondersteunden doorheen het volledige proces!

# Samenvatting

In een maatschappij waar sociale media alomtegenwoordig is, zijn de privacybezorgdheden hier rond evenzeer erg actueel. Bij het ontwikkelen van applicaties moeten privacywetgevingen en -bezorgdheden in acht genomen worden. Dit neemt echter niet weg dat in heel wat applicaties nog gebreken te vinden zijn in de uitvoering van het privacybeleid. In deze scriptie wordt de focus gelegd op de uitvoering van het beleid binnen de fitnesstrackers. Dit zijn platformen met als doel gegevens (die betrekking hebben op sportactiviteiten) op te slaan en te delen met andere gebruikers. Dit zijn gegevens zoals hartslag, gps-locaties, etc. Sommige van deze gegevens kunnen mogelijk gevoelige informatie bevatten of vrijgeven. Gedurende deze thesis wordt getracht om deze gevoelige informatie uit te buiten, met de nadruk op gps-gerelateerde data. Het grootste gevaar bij het delen van deze locaties is het vrijgeven van plekken die je liever niet deelt met de buitenwereld, zoals bv. een woonplaats.

Heel wat van deze fitnesstrackers zijn zich bewust van de mogelijke gevaren en gaan op gelijkaardige manieren te werk om de privacy van de gebruiker te garanderen. Dit gaat echter ten koste van gebruiksvriendelijkheid. Vanuit het perspectief van ontwikkelaars wordt de trade-off tussen privacy en gebruiksvriendelijkheid constant gemaakt. Op de meeste platformen zoals Strava<sup>1</sup> en Garmin worden gelijkaardige privacy features geïmplementeerd. Bijvoorbeeld het verbergen van activiteiten voor andere gebruikers, of enkel activiteiten weergeven voor je volgers. Een andere veelgebruikte techniek is gekend als het gebruik van *EPZs* (Endpoint Privacy Zones). Een EPZ is een cirkel, of bij uitzondering een polygoon, opgezet rond een gevoelige locatie. Deze cirkels worden opgesteld met een radius gekozen door de gebruiker. Het centrum van de EPZ zal een willekeurig punt zijn in de buurt van de locatie in kwestie. Deze kan niet verder dan 70% van de radius verwijderd zijn van de verbergen locatie in het geval van Strava. Elk stuk van het afgelegde traject dat binnen deze zone ligt zal worden verborgen voor de andere gebruikers.

Het verbergen van delen van de route is echter geen waterdicht beveiligingsmechanisme, want hierbij worden bijhorende gegevens niet aangepast of mee verborgen. De bijhorende afgelegde afstand wordt bijvoorbeeld niet aangepast. Voorafgaand onderzoek toonde aan dat het mogelijk is om gevoelige locaties te achterhalen door het gebruik van de totale duur en totale afgelegde

---

<sup>1</sup><https://www.strava.com/>

---

afstand van de activiteiten, in combinatie met het stratenplan van het gebied. Dit soort aanvallen worden *inferentieaanvallen* genoemd. Het traject afgelegd binnenvin de EPZ kan worden afgeleid met behulp van de totale afstand van de activiteit en de zichtbare afstand, afgelegd buiten de EPZ. De afstand binnenvin de EPZ kan dan worden gemapt op het stratenennetwerk, om zo alle mogelijke routes te bekomen die de gebruiker kan afgelegd hebben binnenvin de EPZ. Door dit mechanisme toe te passen op alle activiteiten en geleidelijk aan punten te schrappen die niet voor alle activiteiten een mogelijk eindpunt zijn, kan een intersectie gevonden worden die uiteindelijk de gevoelige locatie oplevert. Dit punt is dan de gevoelige locatie.

Deze thesis onderzoekt mogelijke implementaties van dergelijke inferentieaanvallen in een situatie waarbij de afstand niet gekend is of onbruikbaar is. Als alternatieve gegevens worden de snelheid en het tempo van de activiteiten gebruikt, in combinatie met gps-punten. Deze gevuldde methode bestaat uit drie delen. In de eerste stap beschouwen we de gemiddelde snelheid en de totale duur om de totale afstand te berekenen. Ten tweede worden de gps-punten gebruikt om de afgelegde afstand buiten de EPZ te berekenen. Om dit zo accuraat mogelijk uit te kunnen voeren, bestuderen we smoothing- en map-matchingstrategieën om de best mogelijke resultaten te verkrijgen. Deze twee berekende waarden gebruiken we in de derde stap worden gebruikt om de interferentieaanval uit te voeren. De resultaten van deze aanval zullen worden vergeleken met de resultaten van eerdere implementaties van dit soort aanval.

Met de juiste afstemming van de parameters van het smoothing-algoritme kan een succespercentage tot 75% worden bereikt. Dit is lager dan eerdere implementaties van deze aanval, wat te verwachten is vanwege het type gegevens dat wordt gebruikt. Voornamelijk doordat gps-data soms fouten bevat zoals gps-drift, signaalverlies, gps-bounce, zal de afstand niet altijd even nauwkeurig berekend kunnen worden. Doordat er zoveel punten nodig zijn, resulteren kleine afwijkingen op elk punt in een grote afwijking op de berekende afstand. Maar met dit onderzoek kunnen we aantonen dat een dergelijke aanval mogelijk is en een aanzienlijke nauwkeurigheidsscore behaalt, ondanks het ontbreken van de totale afstand. Dit houdt ook in dat een selectie van de countermeasures die beschreven werden door Dhondt et al. niet meer voldoende zijn om de privacy van een gebruiker te garanderen, en moeten worden uitgebreid.

**Kernwoorden:** gps-locaties, privacy, endpoint privacy zone, inferentieaanval, snelheid

# Abstract

In a society where social media is so ubiquitous, the privacy concerns around them are more relevant than ever. While developing applications, privacy laws and concerns must be taken into account. But this does not mean all these platforms that were built with those in mind are bullet-proof. In a lot of applications it is still possible to find vulnerabilities in the system, with the possibility of rather unpleasant consequences. During this thesis, the main focus will be on the privacy policies of fitness trackers. Fitness trackers are platforms which store and display data related to sport activities. These can be shared with other users, to show your achievements, and possibly motivating others to exercise as well. This data may include heart rate, GPS locations, etc. Some pieces could potentially be more privacy-sensitive than others. The relevant data to study in this thesis are GPS locations and GPS related data (like speed, distance, ...). A great concern about sharing GPS data, is potentially sharing locations you would rather keep private, for example your home location. Sharing full GPS data of your activities could leak this location.

Most fitness tracking networks are aware of this danger and implement a series of countermeasures to prevent leaking this sensitive information. Countermeasures are coming however with a cost, namely a (slightly) worse user experience. From the perspective of the developers of the fitness trackers, a trade-off is consistently being made between privacy and user experience. On most platforms like Strava and Garmin, similar basic privacy features are implemented. These are features like hiding activities, or only sharing activities with your followers. Another commonly used countermeasure is a mechanism known as an *EPZ* (Endpoint Privacy Zone).

An *EPZ* is a circle or polygon drawn around a certain sensitive location. The circular EPZ's will be drawn using a radius chosen by the user, and a center which is a random point in the area of around the sensitive location. This center can't be further than 70% of the radius away from this sensitive location. When this zone is generated, the end and beginning of the trajectory followed which pass through this zone will be hidden for other users.

Most EPZ implementation are not perfect in assuring privacy. While hiding these parts, other useful information is not being hidden or adapted to this sort of cloaking. During this thesis, the goal is to retrieve sensitive locations. This can be achieved by using the total times and distances of the

activities. Previous research showed that it is possible to retrieve sensitive locations using the total distance combined with the street map of the area. These attacks are called *inference attacks*. The distance travelled inside the EPZ can be inferred using the total distance given by the API, and the distance travelled outside of the EPZ (this is the visible distance on the map). Using the distance travelled inside of the EPZ, a route can be constructed and mapped onto the street plan. If all the possible routes are considered, a set of locations will be found. If this is repeated for different activities, with different points where the EPZ is being entered, only one point will remain (in the best case). This would then be the sensitive location.

This thesis investigates the possibilities of such inference attacks using data other than distance as a base. In our implementation, the speed and tempo of the activities will be used, in combination with the GPS locations. This method will consist of three parts. First, the average speed and the total duration will be used to calculate the total distance. Second, the GPS points will be used to calculate the distance travelled outside of the Endpoint Privacy Zone (EPZ). In order to do this effectively, smoothing and map snapping strategies need to be tested out to get the best possible results. These two values can be used in the third step to execute the interference attack. The results of this attack will be compared with the results of the previous implementations of this sort of attack.

This attack is successful in most cases. With the correct tuning of the parameters of the smoothing algorithm, a success rate of 75% can be achieved. This is lower than previous implementations of this attack, which was as expected considering the type of data that is being used. The GPS locations in particular are not always accurate. And because there are so many GPS points needed for these calculations, small deviations on every point result in a large deviation on the calculated distance. But the main conclusion is that this attack is possible, with a reasonable success rate.

**Keywords:** fitness-trackers, privacy, GPS locations, endpoint privacy zone, inference attack

# Inhoudsopgave

<b>Voorwoord</b>	iii
<b>Samenvatting</b>	v
<b>Abstract</b>	vii
<b>Inhoud</b>	xi
<b>Figurenlijst</b>	xiii
<b>Tabellenlijst</b>	xiv
<b>Lijst met afkortingen</b>	xv
<b>1 Inleiding</b>	1
1.1 Situering . . . . .	1
1.2 Doelstelling . . . . .	3
<b>2 Achtergrond</b>	5
2.1 Fitnesstrackers . . . . .	5

2.1.1 Activiteiten . . . . .	5
2.1.2 Berekening Afstanden . . . . .	7
2.2 Mogelijke gps-fouten . . . . .	11
2.3 Privacybeleid . . . . .	12
2.3.1 Endpoint Privacy Zones . . . . .	14
2.4 Literatuur . . . . .	15
<b>3 Setting aanval</b>	<b>18</b>
3.1 Definitie aanvaller . . . . .	18
3.1.1 Assumpties . . . . .	19
3.2 Identificeren van de EPZ . . . . .	21
3.3 Identificatie Entry Gates . . . . .	22
3.4 Bepalen nodige gegevens voor predictie . . . . .	23
3.4.1 Roadgraph en Distance Matrix . . . . .	24
3.4.2 Begin- en eindnodes . . . . .	25
3.4.3 Berekeningen afstand binnenin de EPZ . . . . .	25
3.5 Voorspellen locatie . . . . .	27
3.5.1 Filteren activiteiten . . . . .	27
3.5.2 Bepalen van de locatie . . . . .	28
3.5.3 Regressie om te komen tot een eindvoorspelling . . . . .	29
<b>4 Analyse van de gebruikte data</b>	<b>30</b>

4.1	Karakteristieken van de gebruikte dataset . . . . .	31
4.2	Mogelijke afwijkingen binnenin de dataset . . . . .	32
4.3	Technieken om gps-data te verbeteren . . . . .	34
<b>5</b>	<b>Resultaten en Evaluatie</b>	<b>37</b>
5.1	Evaluatie van de aanval . . . . .	37
5.1.1	De grondwaarheid . . . . .	37
5.1.2	Manueel aanbrengen van een EPZ . . . . .	38
5.1.3	Bootstrapping . . . . .	39
5.1.4	Evaluatie metrieken . . . . .	40
5.2	Resultaten . . . . .	42
5.2.1	Model volgens Dhondt et al. . . . .	43
5.2.2	Gegeven outer distance . . . . .	43
5.2.3	Ruwe gps-data . . . . .	44
5.2.4	Smoothing . . . . .	45
<b>6</b>	<b>Conclusies en toekomstig werk</b>	<b>47</b>
6.1	Conclusies . . . . .	47
6.2	Toekomstig werk . . . . .	49
<b>A</b>	<b>Resultaten van de uitgevoerde aanvallen met verschillende smoothing windows</b>	<b>55</b>
<b>B</b>	<b>Scientific Article</b>	<b>57</b>



# Lijst van figuren

1.1 Voorbeeld Heatmap, vrijgegeven door Strava [33] . . . . .	2
1.2 Voorbeeldactiviteit Strava . . . . .	2
1.3 Voorbeeld van het EPZ mechanisme [38] . . . . .	3
2.1 Verschil snelheid en tempo . . . . .	7
2.2 Data van een activiteit . . . . .	8
2.3 Voorbeeld van de werking van <i>Map Snapping</i> [4] . . . . .	10
2.4 Voorbeeld Data smoothing met een moving average [9] . . . . .	10
2.5 Voorbeelden van gps-drift . . . . .	11
2.6 Voorbeelden van gps-bounce [31] . . . . .	12
2.7 Voorbeeld van zowel gps-drift en gps-bounce uit de gebruikte dataset . . . . .	13
2.8 Voorbeeld van signal loss uit de gebruikte dataset . . . . .	13
2.9 Voorbeeld van de werking van een EPZ . . . . .	14
2.10 Voorbeeld translatie EPZ . . . . .	15
2.11 Voorbeeld filtering van punten binnen EPZ [6] . . . . .	15

2.12 Mechanisme EPZ beschreven door Hassan et al. [38] . . . . .	16
3.1 Voorbeeld van de mogelijke scenarios bij een total distance attack scenario . . . . .	20
3.2 Voorbeeld van een inner distance attack situatie . . . . .	20
3.3 Voorbeeld werking k-means clustering [21] . . . . .	21
3.4 Example of datapoints which can identify a circle [34] . . . . .	22
3.5 Voorbeeld van entry gates gevonden door k-means clustering en de identificatie van een EPZ . . . . .	23
3.6 Voorbeeld van het genereren van een roadgraph <sup>11</sup> . . . . .	24
3.7 Haversine illustratie voor het berekenen van de afstand [39] . . . . .	27
4.1 Geografische spreiding van de activiteiten in de dataset . . . . .	31
4.2 CDF plot van het aantal activiteiten per gebruiker . . . . .	33
4.3 Verdeling van de afstanden tussen twee opeenvolgende gps-punten . . . . .	35
4.4 Verschil tussen de berekende afstand en de theoretische afstand voor één gebruiker	35
4.5 Verdeling van het verschil tussen de berekende afstand en de theoretische afstand buiten de EPZ . . . . .	36
5.1 Dhondt et al. bepaalt grafisch de trend van de afwijkingen bij het snappen van locaties op het wegennetwerk [6] . . . . .	38
5.2 Voorbeeld van een distributie van voorspellingen bepaald door het bootstrapalgoritme [37] . . . . .	40
5.3 Vergelijking van de verschillende aanvallen . . . . .	46

# Lijst van tabellen

4.1	Overzicht van gebruikers en activiteiten . . . . .	32
4.2	Verdeling van de afstanden tussen twee opeenvolgende gps-punten . . . . .	34
5.1	Aanval volgens het model van Dhondt et al. [6] . . . . .	43
5.2	Aanval op basis van gegeven <i>outer distance</i> , en snelheid . . . . .	44
5.3	Aanval op basis van ruwe gps-locaties (geen smoothing) en snelheid . . . . .	45
5.4	Aanval op basis van gesmoothed gps-data en snelheid, met een empirisch bepaald optimaal smoothing window $n = 100$ . . . . .	46
A.1	Aanval op basis van gesmoothed gps-data en snelheid . . . . .	56

# Lijst van afkortingen

**EPZ** Endpoint Privacy Zone

**gps** Global Positioning System

**E.G.** Entry Gate

**API** Application Programming Interface

**LAD** Least Absolute Deviations

**OLS** Ordinary Least Squares

**UTC** Coordinated Universal Time

**CDF** Cumulative Distribution Function

**DBSCAN** Density-Based Spatial Clustering of Applications with Noise

**GT** Ground Truth

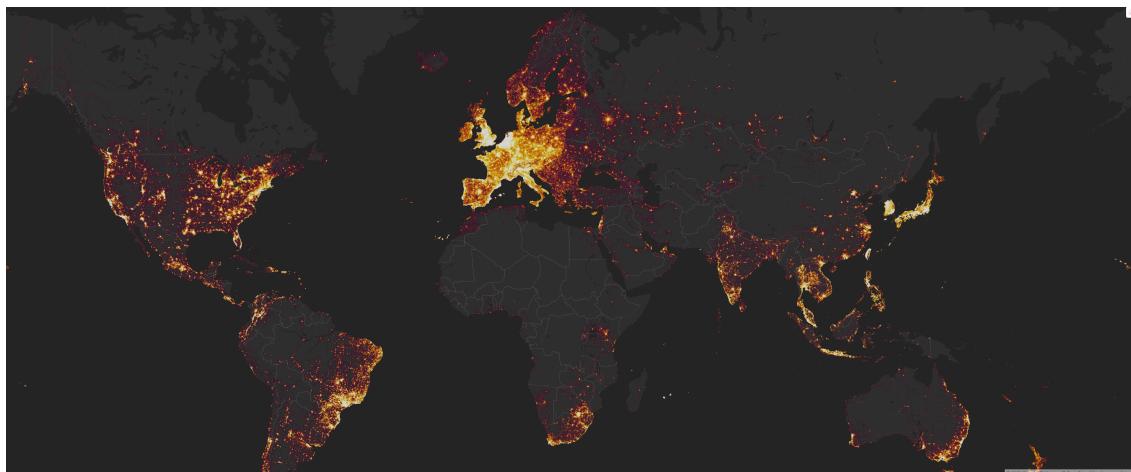
# **Hoofdstuk 1**

## **Inleiding**

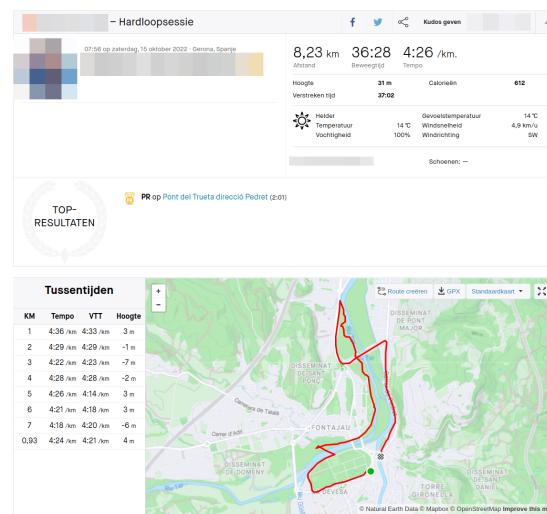
### **1.1 Situering**

Sociale media is zo goed als niet meer weg te denken uit het huidige moderne leven. Over de jaren heen zijn er verschillende definities aan het concept sociale media gegeven. Howard en Park definiëren sociale media als de infrastructuur en tools om content te maken en te verspreiden [13]. Deze definitie is erg ruim, en vertakt zich dus in heel wat facetten, waaronder sociale netwerken, media sharing networks, etc, maar ook de tak van de fitnesstrackers. Deze opkomst van nieuwe media brengt echter ook onbedoelde maar significante privacy bezorgdheden met zich mee.

De focus in deze dissertatie ligt op privacy binnen fitnesstrackers, meer specifiek platformen die Global Positioning System (gps)-locaties gebruiken, zoals Strava, Nike Run Club, etc. Dit zijn platformen waar personen sportactiviteiten zoals lopen, fietsen, wandelen,... kunnen delen met elkaar. Het algemene concept is hierbij dat wanneer je een sportactiviteit uitvoert, je deze voor je volgers en vrienden beschikbaar maakt. De sportactiviteiten zullen natuurlijk bepaalde gegevens bevatten die zichtbaar zijn voor die andere gebruikers. Figuur 1.2 geeft bijvoorbeeld weer hoe Strava de afstand, bewegingstijd, en natuurlijk de gps-locaties deelt. Vele van deze gegevens hebben direct of indirect een negatieve impact op de privacy van de user. Deze negatieve gevolgen komen dan vooral in de vorm van het onbedoeld vrijgeven van *gevoelige locaties*. Onder het concept van een gevoelige locatie vallen heel wat beschrijvingen. Een algemene beschrijving kan zijn, een locatie die geografische informatie deelt die negatieve gevolgen kan hebben, en die je dus liever niet deelt. In het kader van dit onderzoek zal dit gaan over start en eindlocaties van activiteiten. Dit kan gaan over woonplaatsen, wat kan leiden tot o.a. stalking, alsook locaties waar sportmateriaal wordt opgeborgen, wat eventueel kan leiden tot diefstal. Er zijn gevallen bekend van fietsdiefen die Strava gebruiken om fietsen te lokaliseren [36, 1]. Grootschaligere voorbeelden die zeker het vermelden waard zijn, zijn de gevallen waarbij geheime militaire basissen ontdekt worden



**Figuur 1.1:** Voorbeeld Heatmap, vrijgegeven door Strava [33]



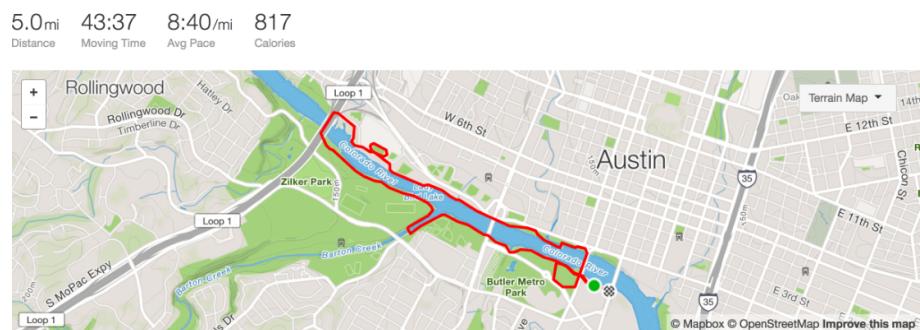
**Figuur 1.2:** Voorbeeldactiviteit Strava

door het bestuderen van de heatmap<sup>1</sup> vrijgegeven door Strava [12]. Figuur 1.1 toont een voorbeeld van deze heatmap.

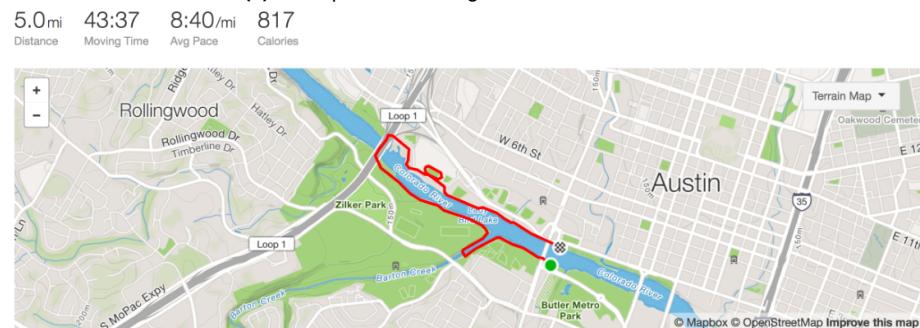
Fitnesstrackers implementeren elk manieren om de privacy van de users te verbeteren. De meest eenvoudige te bedenken manier is misschien wel de mogelijkheid om activiteiten te verbergen voor een selectie van personen (bv. iedereen die geen volger is). Zo kunnen enkel de mensen die de gebruiker explicet toelaat activiteiten bekijken. Een complexer alternatief is het gebruik van Endpoint Privacy Zones (EPZs). Hierbij wordt de weergegeven route voor de persoon die meekijkt gedeeltelijk verborgen. Er wordt als het ware een deel van de route afgekapt, wat zichtbaar is op

<sup>1</sup>Een heatmap is een visuele weergave van gegevens waarbij verschillende kleuren worden gebruikt om de intensiteit van waarden in een matrixachtige structuur weer te geven. In de context van GPS-locaties kan een heatmap worden gebruikt om de concentratie of frequentie van locaties op een kaart weer te geven [35]. Het doel is om gebieden met een hoge dichtheid of veelvuldige locaties te identificeren en visueel te markeren.

Figuur 1.3. De echte begin- en eindpunten zullen binnenzitten in het afgekapt deel liggen. Er zullen nieuwe punten worden gegenereerd, op de rand van de cirkel, die voor de externe waarnemer het begin en einde zullen voorstellen. Het begin- en eind-deel van de route wordt dus onzichtbaar voor de andere gebruikers. Door de aanwezigheid van al deze pogingen tot privacyverbeteringen valt op dat de ontwikkelaars van de platformen zich erg bewust zijn van de mogelijke gevaren. Echter is er een afweging te maken bij de implementatie tussen de bruikbaarheid van het platform, en de privacy van de eindgebruiker. Hoe meer data wordt vrijgegeven, hoe groter de kans op mogelijk gevoelige info wordt meegegeven. Aan de andere kant, bij het weglaten van informatie gaat de gebruiksvriendelijkheid en de aanwezigheid van nuttige data van het platform er sterk op achteruit.



(a) Standpunt van de eigenaar van de activiteit



(b) Standpunt van andere gebruikers, waarbij de route als het ware afgekapt wordt

**Figuur 1.3:** Voorbeeld van het EPZ mechanisme [38]

## 1.2 Doelstelling

In dit onderzoek bekijken we of er een mogelijkheid bestaat om private locaties (verborgen start- en eindlocaties) van een activiteiten te achterhalen, ondanks het gebruik van de EPZ beschreven in Sectie 2.3.1 als privacy beveiligingsmechanisme. In het verleden werden al enkele manieren beschreven om a.d.h.v. andere metadata zoals hoogtedata en afstanden de EPZ te omzeilen [37, 6, 38]. In deze thesis wordt meer in detail gegaan op het gebruik van snelheidsdata. Als basis voor

deze aanval gebruiken we de inferentieaanval beschreven door Dhondt et al.. We onderzoeken of deze aanval nog steeds mogelijk is bij het weglaten van bepaalde gegevens, en dus door het gebruik van andere gegevens. De focus ligt in deze studie voornamelijk op snelheidsdata.

Om deze doelstelling te behalen, dienen we eerst de aanval volgens Dhondt et al. onder de loep te nemen. Daarna kunnen we de verschillende mogelijke aanvalsscenario's beschrijven, en voor elk scenario de nodige berekeningen uitwerken die nodig zijn opdat dit scenario kan worden uitgevoerd. Voordat we de aanval kunnen uittesten en analyseren, moeten we de dataset beschouwen op mogelijke fouten, en deze ook in acht nemen voor mogelijke conclusies. Ook voeren we een analyse uit op het verschil tussen de berekende afstanden, en de waarden afgeleid volgens de berekeningen van Dhondt et al. Zo kunnen we de effectiviteit van de aanval a priori al voor een stuk inschatten. Dan pas voeren we de aanval uit, en kunnen we de aanvallen evalueren en besluiten trekken.

# **Hoofdstuk 2**

## **Achtergrond**

### **2.1 Fitnesstrackers**

De focus van deze scriptie ligt op mogelijke tekortkomingen of vulnerabilities betreffende het privacybeleid in fitnesstrackers. Maar vooraleer we een aanval op basis van deze kwetsbaarheden kunnen opzetten, is het noodzakelijk om een vat te krijgen op welke manier een fitnesstracker info verzamelt en weergeeft, en meer precies, hoe de mechanismen die de privacy voorzien voor de gebruikers in detail werken.

De data waarmee de aanval wordt opgezet en waarmee wordt geëxperimenteerd, is afkomstig van de populaire fitnesstracker *Strava*<sup>1</sup>. Dit is een sociaal netwerk waarbij alle soorten sporters hun activiteiten kunnen delen. Dit gaat over lopen, wandelen, fietsen, zwemmen, maar ook sporten als fitnessen, voetballen, etc. De verzamelde data wordt volgens het perspectief van een mogelijke aanval gefilterd, niet alle data blijkt nuttig te zijn. Enkel data die gevoelige informatie met betrekking tot de woonplaats zou kunnen vrijgeven wordt behouden. Dit zal er dus op neerkomen dat enkel activiteiten die relevante gps-informatie bevatten in beschouwing worden genomen, meer specifiek zal dit gaan over *runs, hikes, walks, and rides*.

#### **2.1.1 Activiteiten**

Een Strava-activiteit bevat erg veel informatie. Echter is niet alles even bruikbaar. Een correcte abstractie van de onnodige data is dus nodig. Figuur 2.2 geeft een voorbeeld van een gedetailleerde activiteit weer. Een gebruiker is in staat om de activiteit een titel te geven, en er een korte

---

<sup>1</sup><https://www.strava.com/>

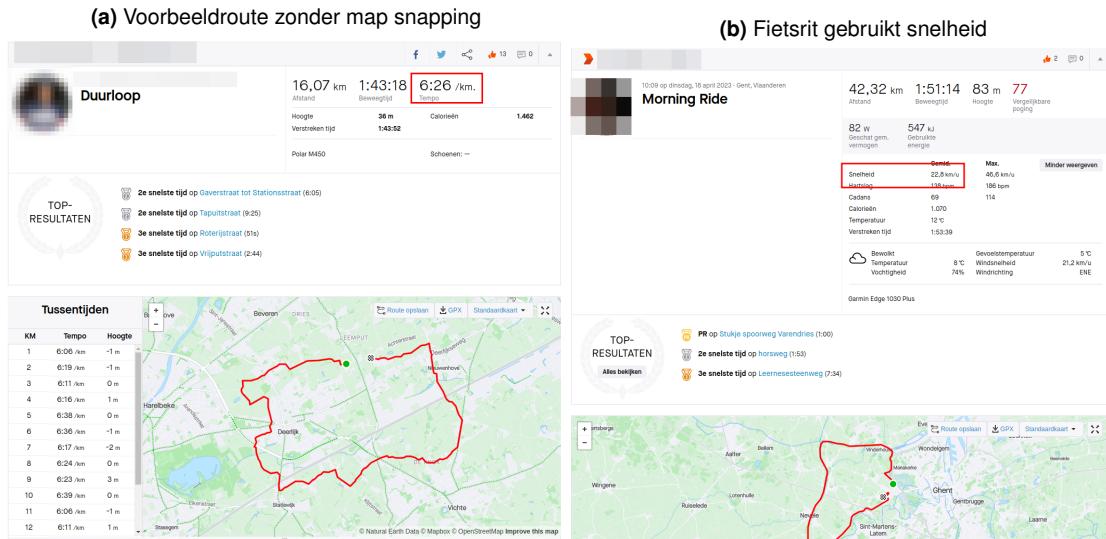
beschrijving aan toe te voegen. Ook een foto kan optioneel toegevoegd worden. De exacte datum en tijd van de start van de activiteit wordt hierbij ook weergegeven.

Rechts bovenaan zijn de algemene basisstatistieken te zien. Dit zijn de totale afgelegde afstand, de totale bewegingstijd, de gemiddelde snelheid of het gemiddelde tempo, het totale hoogteverschil, de totale verstreken tijden, en het aantal calorieën verbrand. Als extra kunnen hier enkele statistieken m.b.t. het gebruikte materiaal, zoals type fiets, loopschoenen, hartslagmeter, enzovoort worden weergegeven. Een belangrijk onderscheid in deze context is het verschil tussen de beweegtijd en de verstreken tijd. Deze twee lijken in definitie gelijk, maar dit zijn ze niet. Strava, en fitnessplatformen in het algemeen werken met twee verschillende soorten tijdsberekeningen voor het bekomen van een accuratere gemiddelde snelheid of tempo. De verstreken tijd is simpelweg het tijdsinterval tussen het vertrek van de activiteit en de aankomsttijd ervan. De bewegingstijd is de tijd waarbij de gebruiker zich effectief bewoog. Met andere woorden worden de tijden waarbij de gebruiker stilstond uit de verstreken tijd gefilterd. Dit kan gaan over bijvoorbeeld een pauze, of het wachten voor een verkeerslicht.

Er is een verschil bij fietsactiviteiten en wandelactiviteiten in hun weergave. In het geval van een fietsactiviteit wordt *snelheid* weergegeven, en in het geval van een wandel- of loopactiviteit wordt *tempo* weergegeven, zoals te zien is op Figuur 2.1. Deze worden beide berekend aan de hand van de bewegingstijd. Een kanttekening hierbij is dat dit enkel geldt voor activiteiten die niet gelaaberd zijn als *race*, indien dit toch het geval is, wordt de snelheid berekend in functie van de totaal verstreken tijd [28]. Het verschil tussen deze twee is dat de snelheid wordt berekend volgens de formule  $v = \frac{d}{t}$ . De eenheid van snelheid is dan ook  $\frac{m}{s}$  of, in het geval van fitnesstrackers,  $\frac{km}{h}$ . Het tempo wordt berekend volgens de formule  $tempo(\frac{\text{min}}{\text{km}}) = \frac{t(\text{min})}{d(\text{km})}$ . De eenheid van tempo is  $\frac{\text{min}}{\text{km}}$ . Om deze berekeningen wat te standaardiseren, werd gedurende deze thesis gekozen om altijd de omrekening te maken naar snelheid  $\frac{km}{h}$ , om zo over de volledige lijn met dezelfde standaard te werken.

Onder de basisstatistieken zijn de *Strava-segmenten* te zien. Een Strava-segment is een specifiek deel van een bepaalde route dat gebruikers van de sport-app kunnen markeren, delen en vergelijken met andere gebruikers. Het segment is een bepaalde afstand en route, bijvoorbeeld een klim of afdaling, die vaak wordt beschouwd als een uitdagende of iconische sectie van een bepaalde fiets- of hardlooproute. Gebruikers van Strava kunnen een segment maken door de begin- en eindpunten op een kaart aan te geven en een naam en beschrijving toe te voegen. Zodra het segment is gemaakt, kunnen andere gebruikers het segment vinden en deelnemen aan een leaderboard, waarop de snelste tijden worden bijgehouden en vergeleken met andere gebruikers. Segmenten worden vaak gebruikt om prestaties te meten en te vergelijken.

Centraal op de figuur is ook de kaart duidelijk zichtbaar. Daarbij horen ook de tussentijden en de grafiek van de snelheidsevolutie. Optioneel kan hierbij ook nog een visualisatie van de afgelegde hoogte en de hartslag worden weergegeven, indien de gebruiker hiervoor met de juiste meetinstru-



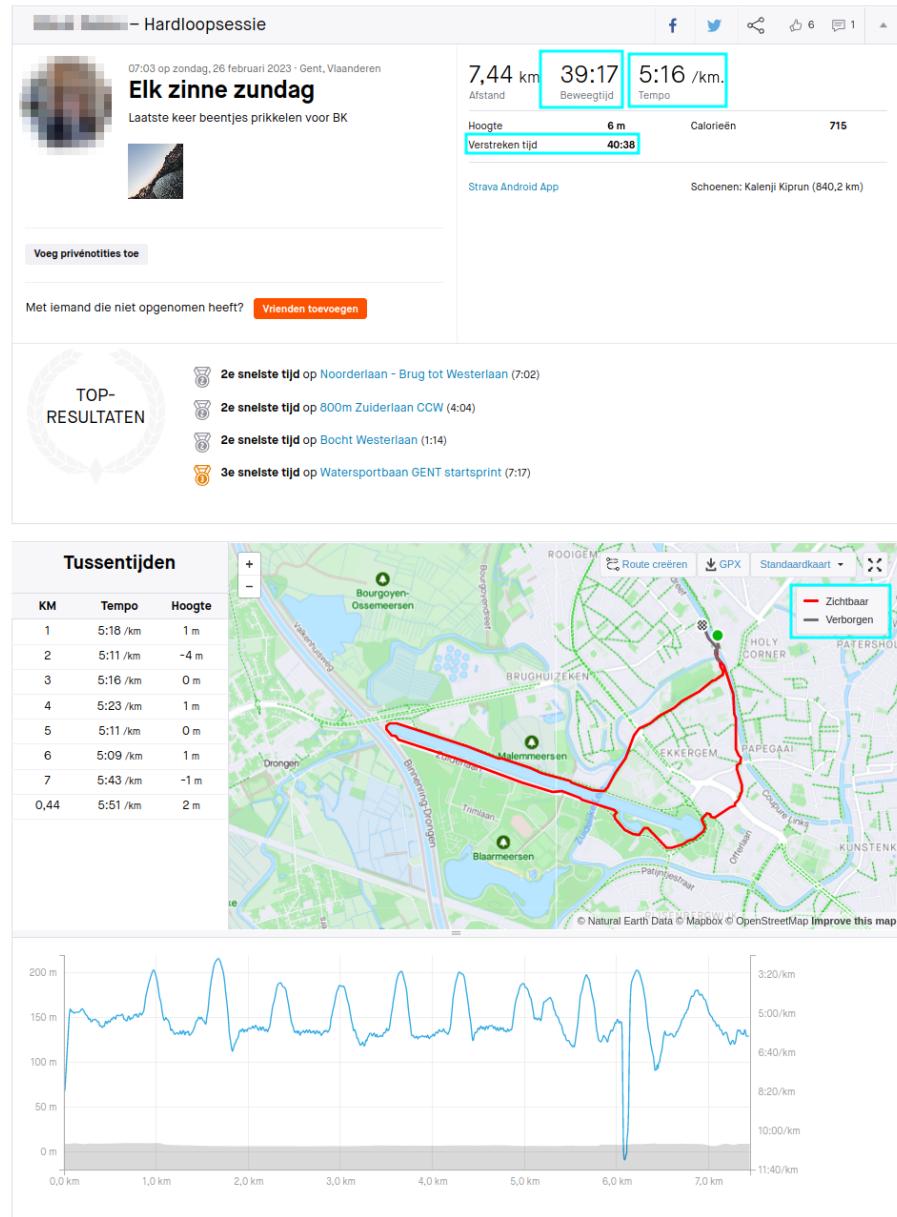
Figuur 2.1: Verschil snelheid en tempo

menten zijn sportactiviteit opneemt. De tussentijden en de grafiek van snelheid zijn qua inhoud gelijkaardig, met als verschil dat de waarden op de grafiek erg precies kunnen worden bestudeerd. Op de grafiek is voor elk afstandspunt de ogenblikkelijke snelheid zichtbaar. Bij de tussentijden wordt de gemiddelde snelheid over een kilometer weergegeven. De kaart die de route weergeeft is zeker ook belangrijk om even te bestuderen. Deze bevat namelijk alle gps-geregistreerde punten, en verbindt deze ook om zo één aaneensluitende route te vormen. Wanneer deze echter in detail bestudeerd wordt, samen met de legende die aanwezig is, is te zien dat de route uit twee delen bestaat, een zichtbaar deel en een onzichtbaar deel. Een andere gebruiker zal enkel zicht hebben op het zichtbare deel, het onzichtbare deel zal dus voor een andere gebruiker niet zichtbaar zijn. Anders geformuleerd, de activiteit zal voor deze persoon dus als het ware afgekapt zijn, en zal in de voor hem of haar zichtbare versie op een andere plek starten en eindigen. In de volgende Secties 2.3 & 2.3.1 wordt meer in detail ingegaan op de werking van deze methodiek.

Een laatste kanttekening die we hierbij maken, is dat een gebruiker keuze heeft tussen verschillende eenheden. Er is keuze mogelijk tussen de mijl en pond, en kilometer en kilogram. Gebruikers kiezen in welke eenheid ze de applicatie wensen te gebruiken. Voor de gebruiker in kwestie zal dan ook de volledige applicatie worden weergegeven in de gekozen eenheden.

## 2.1.2 Berekening Afstanden

Fitnesstrackers krijgen vanuit de buitenwereld ruwe data binnen. Deze data moet dus verwerkt worden vooraleer ze bruikbaar is voor de gebruiker. Er werd al kort ingegaan in Sectie 2.1.1 op de berekening die Strava gebruikt voor de snelheid. Echter is het ook interessant om de berekening

**Figuur 2.2:** Data van een activiteit

van Strava eens onder de loep te nemen voor de afgelegde afstand. Strava maakt gebruik van twee verschillende methodieken voor het berekenen van deze afstand. De eerste is de *GPS-calculated Distance*. Dit bestaat eruit om de afstand tussen opeenvolgende ontvangen gps-punten te berekenen, en deze op te tellen. Precisie is hier afhankelijk van de precisie van de gps-punten, aangezien de afstand wordt berekend door de punten met rechte lijnen te verbinden. Dit kan gebeuren in real time, via de gsm, smartwatch of ander toestel die gebruikt wordt om de activiteit op te nemen. Er zal dan ook mogelijkheid zijn om real time info te zien. Op elk punt zal de afstand vanaf het startpunt gekend zijn, en het is deze afstand die gedeeld zal worden op het platform. Het grote nadeel hierbij is het real-time aspect. Fouten kunnen moeilijker on the fly worden gecorrigeerd. Een tweede aanpak voor GPS-calculated distance is om gps-data pas bij het uploaden te verwerken. De gps-data wordt dan geanalyseerd, en de nodige berekeningen worden uitgevoerd.

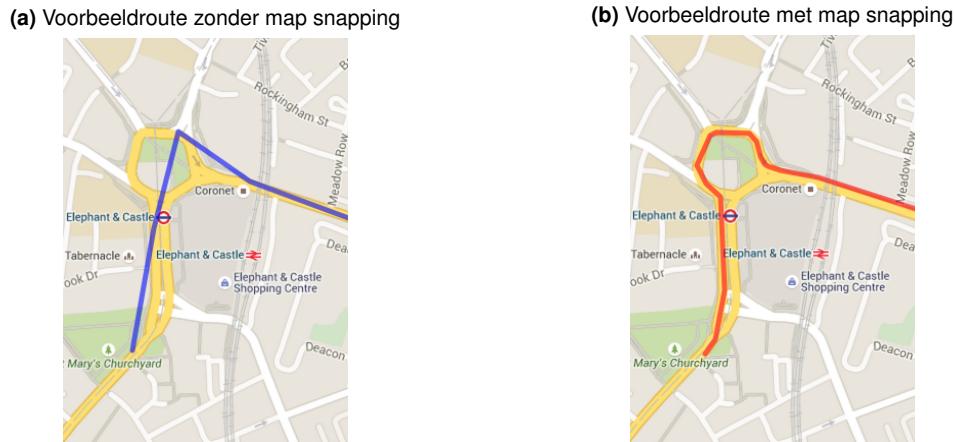
Het alternatief voor de GPS-calculated distance is de *Ground Speed Distance* methodiek. Deze afstand kan enkel worden bepaald in het geval van een fietsactiviteit met behulp van een capabele fietscomputer die omwentelingen van de wielen kan meten. De fietscomputer berekent dan de afstand door het aantal omwentelingen te vermenigvuldigen met de omtrek van het fietswiel [32].

De bovenstaande afstandsberekeningen zijn de twee technieken die de officiële supportdocumentatie van Strava beschrijft [32]. Echter blijkt wanneer we de afstand op deze manier manueel berekenen, we afwijkende resultaten bekomen worden. Dit is zeer waarschijnlijk te wijten aan de preprocessing van de data die gebeurt bij het uploaden van een activiteit. Alhoewel dit niet expliciet gedocumenteerd staat doen de resultaten dit wel sterk vermoeden. De hypothese is dat tijdens het uploaden, de afstand herberekend wordt. De gps-punten zullen worden geanalyseerd, en dat het platform hierbij technieken gebruikt om de resultaten hiervan te verbeteren. De twee meest waarschijnlijke technieken zijn *Map Snapping* en *Smoothing*.

Map Snapping (ook wel *Map Matching* genoemd) is een techniek waarbij gps-punten worden verschoven naar de dichtstbijzijnde weg. Per gps-punt wordt gezocht naar de dichtste node op de desbetreffende *roadgraph*<sup>2</sup>, op Figuur 2.3 is de werking ervan te zien [4].

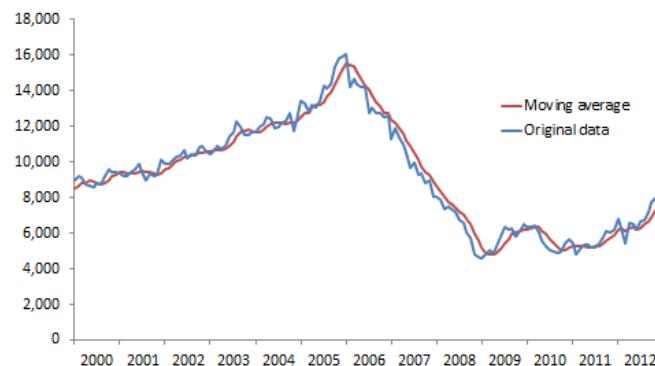
Daarnaast bestaat de kans dat er gebruik gemaakt wordt van smoothing. Smoothing is een proces dat ruwe gps-punten (of datapunten in het algemeen) op een traject probeert te optimaliseren opdat ze een vloeind 'curve' vormen. Dit wordt bekomen door ruis, schommelingen en onnauwkeurigheden te filteren uit het traject. Hiervoor bestaan verschillende implementaties. Aangezien Strava geen openbare informatie verstrekkt over het gebruik van gps-smoothing, is het niet bekend of ze deze techniek effectief toepassen. Het is dus gissen naar, indien ze deze zouden gebruiken, welke implementatie dan wel gebruikt wordt. De makkelijkste en meest modulaire methode om aan smoothing te doen, is *Smoothing met Moving Average*. Deze methode bestaat eruit om van een

<sup>2</sup>De roadgraph is een wegennetwerk, omgezet in een graaf, bestaande uit edges en nodes. Elke weg of pad, bevat één of meerdere nodes, zodat een skeletstructuur ontstaat, die een abstractie van het wegennetwerk voorstelt [23]



**Figuur 2.3:** Voorbeeld van de werking van *Map Snapping* [4]

aantal punten in een bepaalde range (ook 'window' genoemd) het gemiddelde te nemen, en vervolgens op te schuiven. Het gemiddelde wordt berekend met volgende formule:  $\bar{y}_x = \frac{y_x + y_{x+1} + \dots + y_{x+n}}{x+n}$ , voor punt x, met n als window-grootte [8, 9, 20]. Zo kan voor elk punt een evenwichtige waarde op de nieuwe grafiek bekomen worden, en krijgt de grafiek een meer vloeiende vorm. Merk wel op dat de precisie van de route afneemt op deze manier. Bij het smoothen van een traject wordt het aantal gebruikte punten namelijk verminderd volgens de grootte van de window. Afhankelijk van de grootte, worden meer (resp. minder) punten samengenomen, en zo minder of meer punten weergegeven op de grafiek. Een voorbeeld is terug te vinden op Figuur 2.4, waarbij de blauwe curve de ruwe data voorstelt, dus voorafgaand op het 'smoothen', en de rode de 'gesmoothe' curve.

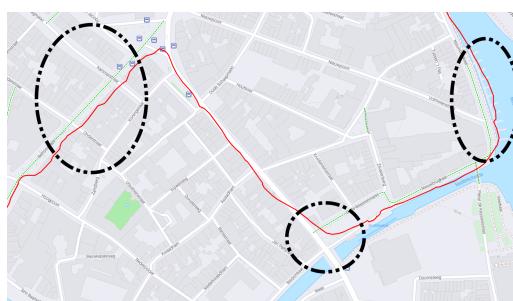


**Figuur 2.4:** Voorbeeld Data smoothing met een moving average [9]

## 2.2 Mogelijke gps-fouten

Zoals reeds aangehaald kunnen er bij het verzamelen van gps-data significante fouten optreden. Met gps-fouten wordt gedoeld op data die de gps-sensor ontvangt die niet overeenstemt met de werkelijke gps-locatie. Deze fouten kunnen verschillende oorzaken hebben. De belangrijkste zijn hierbij *gps-drift*, *gps-signal loss* en *gps-bounce*.

Gps-drift is een fenomeen waarbij de gps-locatie van een gebruiker afwijkt van de effectieve locatie. Twee voorbeeld zijn terug te vinden op Figuur 2.5, waarvan Figuur 2.5a rechtstreeks afkomstig is vanaf Strava. Hierbij is te zien dat de gebruiker een deel van de route door gebouwen heen en door het water aflegt. Dit kan worden veroorzaakt door dichtbebauwde omgevingen, en omgevingsfactoren zoals hoge bomen. Er zijn enkele redenen waarom dit gebeurd, namelijk hoge gebouwen, wolkenkrabbers of dichte boombedekking kunnen het GPS-signal verstören. GPS-satellieten zenden radiosignalen uit die gemakkelijk kunnen worden belemmerd of geblokkeerd door fysieke obstakels. Wanneer het signaal wordt verstoord, kan de GPS-ontvanger moeite hebben om nauwkeurige locatiegegevens te berekenen. Reflectie van signalen kan op zijn beurt ook een impact hebben. De eerder beschreven map snapping kan dit eventueel tegengaan.



(a) Voorbeeld van gps-drift op Strava

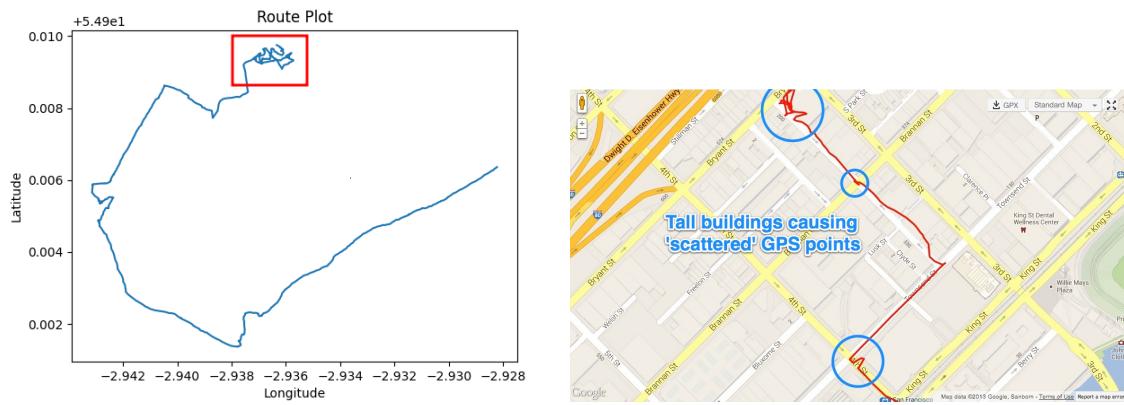


(b) Voorbeeld van gps-drift op een satellietkaart, veroorzaakt door dichte boomgroe [31]

**Figuur 2.5:** Voorbeelden van gps-drift

Gps-bouncing is een fenomeen hoofdzakelijk veroorzaakt door hoge gebouwen. Het gps-signaal zal hierbij weerkaatsen tussen de gebouwen, op weg naar het toestel vanaf de satelliet. De vertraging zorgt er voor dat het apparaat denkt een extra afstand afgelegd te hebben, terwijl dit niet zo is. De uitkomst van het traject is dan onvoorspelbaar, wat leidt tot een 'cluster' van gps-punten wanneer dit voor een paar punten in dezelfde omgeving gebeurt. Voorbeelden hiervan zijn terug te vinden op Figuur 2.6. Om dit fenomeen op zijn beurt tegen te gaan, is het best om smoothing toe te passen bij het berekenen.

Er zijn ook voorbeelden waarbij beide fenomenen voorkomen, zoals te zien is op Figuur 2.7. Indien we in dit geval op een naïve manier de totale afgelegde afstand berekenen zal zich een significant verschil voordoen tussen de afstand die de gebruiker werkelijk afgelegd heeft, en de berekende



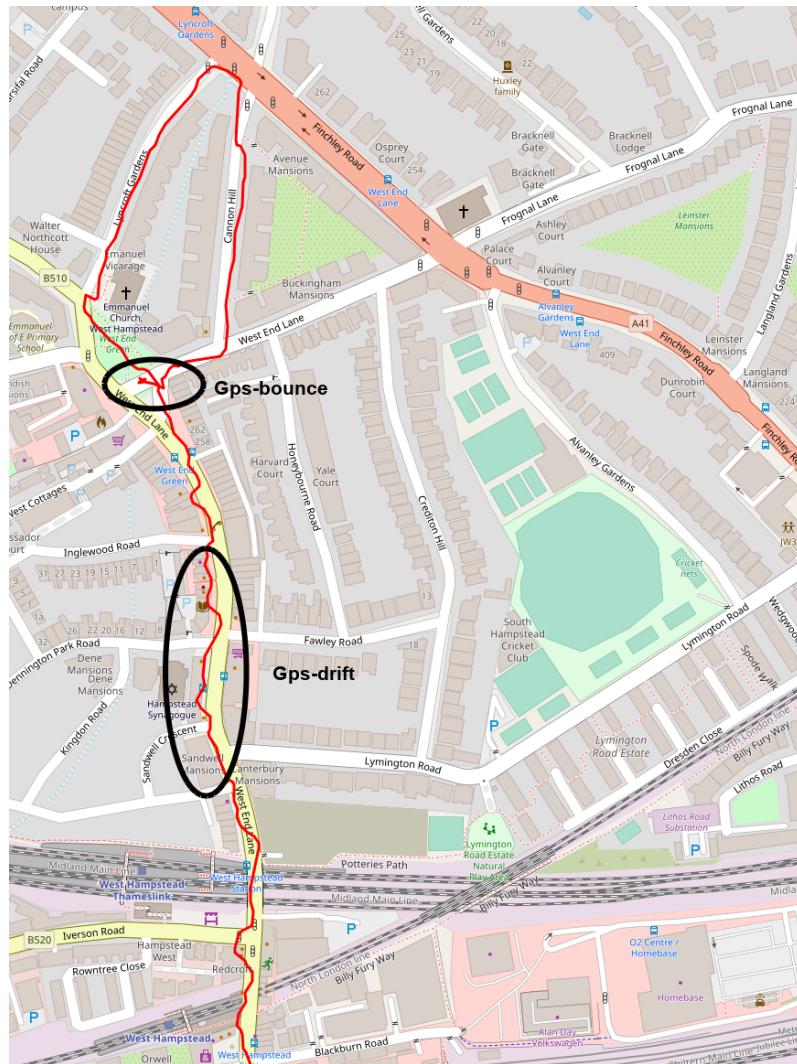
**Figuur 2.6:** Voorbeelden van gps-bounce [31]

afstand. De naïve manier van afstandsberekeningen houdt in dat we de totale som van het afstandsverschil tussen twee opeenvolgende punten nemen.

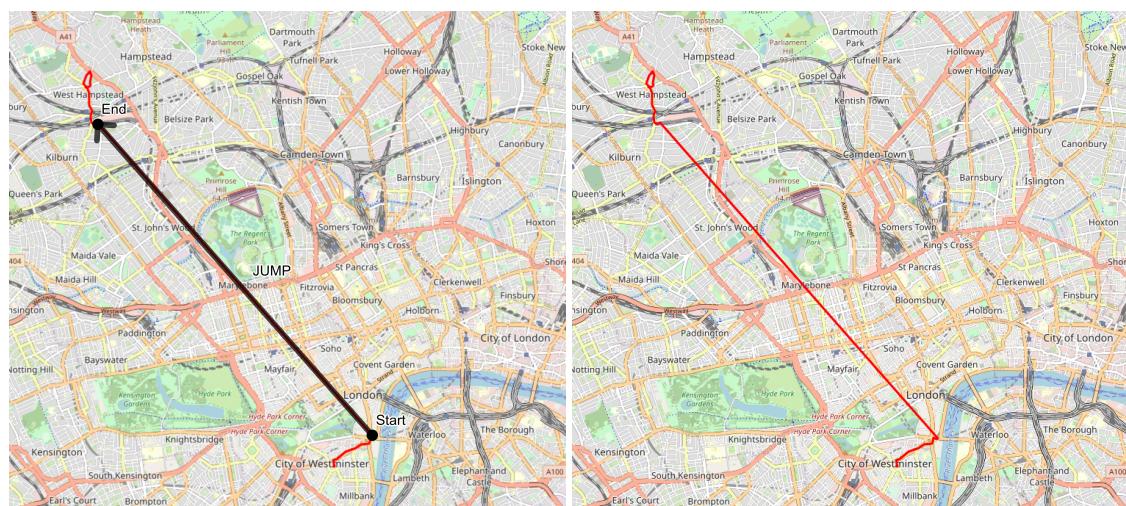
Een laatste fenomeen dat kan optreden is gps-signal loss. Hierbij gaat het signaal van de gebruiker verloren, en wordt pas op een later tijdstip terug een nieuw signaal ontvangen, waardoor een sprong werd gemaakt. In dit geval zou map matching opnieuw een goede oplossing om dit tegen te gaan. Een tweede oorzaak die kan leiden tot signal loss, die zeker van toepassing is bij fitness trackers, is de mogelijkheid tot het pauzeren van een activiteit. In dit geval wordt de activiteit gepauzeerd voor een bepaald tijdsframe, en wordt er geen data meer verzameld. Wanneer de activiteit terug wordt hervat, zal er een sprong zijn in de gps-locaties, wat kan leiden tot een verkeerde berekening van de afstand. In het geval van een pauze zal map matching geen oplossing zijn, maar zullen we deze ‘sprongafstand’ best weglaten in de berekeningen. Een voorbeeld hiervan is terug te vinden op Figuur 2.8.

## 2.3 Privacybeleid

Het is zeker niet altijd wenselijk om alle data die vervat zit in zo’n activiteit met alle andere gebruikers op het platform te delen. De ontwikkelaars kiezen er dan ook voor om gebruikers de mogelijkheid te geven om hun privacy te bewaren. In deze sectie wordt de focus gelegd op de mechanismen gebruikt door *Strava*. Er valt op dat heel wat andere sport-applicaties vergelijkbare, zo niet dezelfde methodieken gebruiken. Een eerste algemeen mechanisme bestaat eruit om de gebruiker de keuze te geven om alle activiteiten en alle gegevens over het profiel heen te laten voldoen aan bepaalde privacy regels. Deze regels kunnen ook per activiteit worden ingesteld. Onder de keuzes regels. Deze regels kunnen ook per activiteit worden ingesteld. Onder de keuzes staan meestal drie opties: *zichtbaar voor iedereen*, *zichtbaar voor volgers* en *zichtbaar voor niemand*. Er kan ook zelfs een keuze gemaakt worden om specifieke elementen van een activiteit niet te delen



**Figuur 2.7:** Voorbeeld van zowel gps-drift en gps-bounce uit de gebruikte dataset



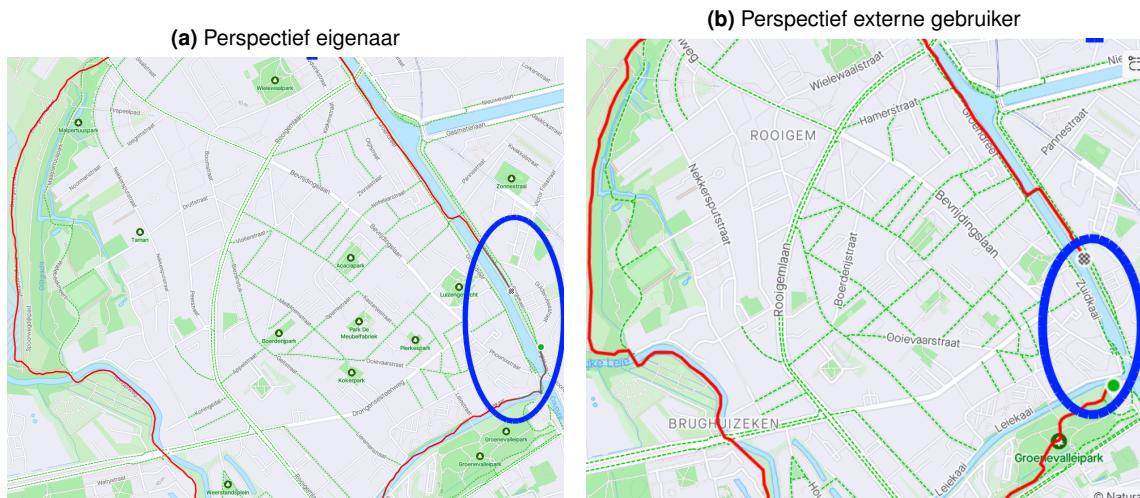
**Figuur 2.8:** Voorbeeld van signal loss uit de gebruikte dataset

met de buitenwereld, zoals bijvoorbeeld de zichtbaarheid van de route op de kaart [30].

Een tweede belangrijke maatregel is het gebruik van de EPZs.

### 2.3.1 Endpoint Privacy Zones

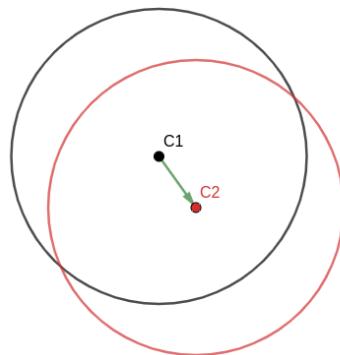
Een EPZ is een cirkelvormige zone met een bepaalde straal rond een gps-punt. Het punt in kwestie zal dus de betreffende *gevoelige locatie* zijn. De gebruiker kan de straal van deze cirkel<sup>3</sup> zelf kiezen, en in het geval van Strava hebben gebruikers keuze uit waarden van 0 tot 1600m, in stappen van 200m. Wanneer een gebruiker binnen deze zone zijn activiteit beëindigt of begint, dan zal dat deel van de route binnen de EPZ niet zichtbaar zijn voor anderen. Vanuit het perspectief van een andere gebruiker zal de activiteit dus starten en/of eindigen op de rand van deze cirkel (die natuurlijk niet zichtbaar is). Merk op dat een sporter meerdere gevoelige locaties kan verbergen op de kaart. Bijvoorbeeld een frequent bezocht café, of een huis van een partner waar regelmatig een tussenstop plaatsvindt. Een tweede opmerking is dat wanneer een gebruiker de EPZ doorkruist, maar er niet in stopt, dat deel van de route onaangepast blijft. Op Figuur 2.9 zijn de verschillende perspectieven te zien, hoe de eigenaar de activiteit ziet, en hoe het eruit ziet voor een externe gebruiker. Het traject dat de buitenstaander te zien krijgt, bestaat uit alle punten die zich buiten de EPZ bevinden. Merk ook op dat de eigenaar van de activiteit zicht heeft op de invloed van de EPZ, dus wat zal verborgen worden erdoor, en wat zichtbaar blijft. Dit onderscheid wordt gemaakt door het verschil in kleur, oranje voor de publiek zichtbare punten en grijs voor de onzichtbare.



**Figuur 2.9:** Voorbeeld van de werking van een EPZ

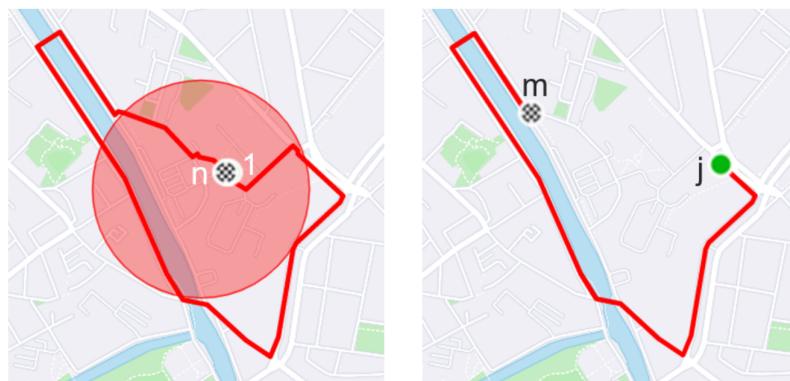
<sup>3</sup>Op Strava heeft de EPZ de vorm van een cirkel, maar op andere platformen kunnen andere vormen de norm zijn, bv. polygonen.

De methodiek die fitnesstrackers toepassen bij het opzetten van een EPZ werkt als volgt, de gevoelige locatie wordt genomen als beginlocatie. Hieruit zetten ze a.d.h.v. de op voorhand vastgelegde EPZ-straal een cirkel op. Het centrum van deze cirkel zal hierna een translatie ondervinden in een willekeurige richting. Dit kan een verschuiving zijn met een afstand die maximaal 70% van de straal van de EPZ bedraagt. Dit mechanisme is te zien op Figuur 2.10. Het translateren van deze cirkel wordt ook *spatial cloaking* genoemd.



**Figuur 2.10:** Voorbeeld translatie EPZ

Daarna worden alle punten vertrekende vanaf de gevoelige locatie tot aan de rand van de EPZ, en vanaf de rand van de EPZ tot aan de gevoelige locatie verwijderd van het zichtbare traject. Een voorbeeld van deze filtering is te zien op Figuur 2.11, waarbij duidelijk zichtbaar is dat punten die de EPZ doorkruisen, maar niet vertrekken of aankomen bij de gevoelige locatie niet worden gefilterd.

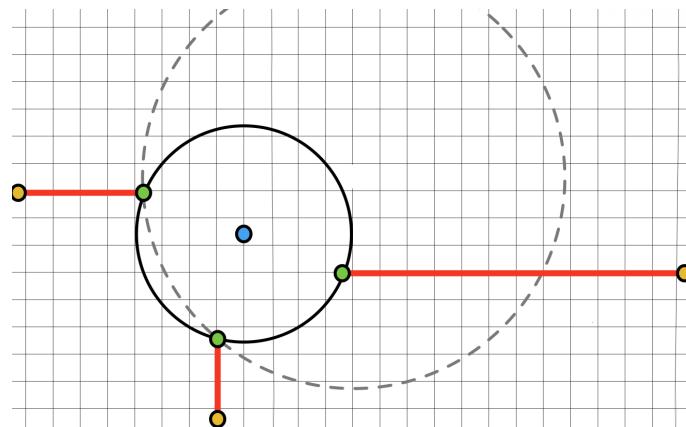


**Figuur 2.11:** Voorbeeld filtering van punten binnen EPZ [6]

## 2.4 Literatuur

In het verleden is al wat onderzoek verricht in de richting van de doeltreffendheid van EPZs bij fitnesstrackers. Hassan et al. (2018) beschreven een implementatie van EPZ waarbij het centrum

van de zone de gevoelige locatie is [38]. Met andere woorden om deze gevoelige locatie te achterhalen is het dus voldoende deze zone te identificeren. In tegenstelling tot deze thesis, wordt ervan uitgegaan dat het centrum geen translatie ondervindt, en er dus geen spatial cloaking wordt toegepast. In deze paper van Wajih Ul Hassan wordt gefocust op de reconstructie van de cirkel op basis van 3 punten op de rand, wat te zien is op Figuur 2.12. Deze 3 randpunten worden dus gekomen door begin- of eindpunten te nemen van activiteiten, volgens het perspectief van gebruiker die geen eigenaar is. Deze begin- of eindpunten zullen zich altijd op de rand van de cirkel bevinden. Door dit aanvalsmodel toe te passen bekwamen Hassan et al. een succes rate tot 95.1%. Spatial cloaking werd er aangehaald als mogelijke verdediging tegen dit soort aanvallen.



**Figuur 2.12:** Mechanisme EPZ beschreven door Hassan et al. [38]

Een onderzoek door Mink et al. (2022) toonde ook aan dat heel wat mensen in staat zijn om de gevoelige locatie te achterhalen op basis van hun intuïtie [18]. Dit gebeurde op basis van enquêtes die werden afgenoem bij gebruikers van het platform. Deelnemers aan de enquête moesten op basis van activiteiten opgenomen door een fitnesstracker, die verhuld waren gebruik makend van een EPZ, weliswaar zonder spatial cloaking, de startlocatie van een gebruiker proberen te achterhalen. Uit het onderzoek bleek dat 68% van de ondervraagden bij een EPZ-radius van 200m de beschermd locatie tot op 50m nauwkeurig konden voorspellen. Hoe meer activiteiten ter beschikking zijn, hoe effectiever de deelnemers de locatie konden schatten. Deze resultaten op zich zijn alarmerend, en tonen aan dat EPZs verre van perfect zijn, en ook te omzeilen zijn door een persoon die geen technische achtergrond heeft.

Dhondt et al. (2022) voerden tevens ook een studie naar de mogelijke lekken aanwezig in het principe van EPZs [6]. Er wordt in deze paper een nadruk gelegd op de translatie van de EPZ, en hoe deze de privacy van een gebruiker verhoogt. Ze beschrijven een inferentieaanval die gebruikmaakt van de totale afstand die terug te vinden bij de activiteit. Het principe van deze inferentieaanval wordt uitvoerig beschreven in Hoofdstuk 3. In het kort werkt de aanval als volgt: aan de hand van de totale afgelegde afstand in combinatie met het wegennetwerk in die omgeving, wordt een poging gedaan om alle mogelijke routes die de sporter binnenin de EPZ zou kunnen afgelegd hebben te

reconstrueren. Dit gebeurt voor elke activiteit. Wanneer dit gedaan wordt voor verschillende trajecten, kan een locatie voorspeld worden die het meest waarschijnlijk wordt geacht om de gevoelige locatie te zijn.

Dhondt et al. toonden aan dat de beschreven countermeasures door Hassan et al. niet feilloos zijn, en dat deze te omzeilen valt met een successrate van 85%. Aangezien activiteiten nog steeds totale afstanden van een volledige route vrijgeven, kunnen ze de afstand afgelegd binnenvinden de EPZ infereren. Deze data ligt dan ook aan de grond van de inferentieaanval volgens Dhondt et al.

Het meest recente werk in dit domein is de thesis van Verdonck (2022) [37]. Deze thesis bouwt in grote mate verder op de paper van Dhondt et al., maar maakt gebruik van alternatieve data. Verdonck onderzoekt in hoeverre hij gelijkaardige resultaten kan bekomen door het gebruik van hoogtedata om beschermd locaties te achterhalen. Via de kennis van hoogtedata van het stratenplan kan hij via de gekende hoogteverschillen een inferentieaanval opzetten, die nu geen afstanden maar hoogteverschillen infereert. Op deze manier bekomt Verdonck een succes rate van 36%. Dit lagere succesratio is terug te brengen naar het feit dat hoogtedata een stuk minder precies zijn. Ook is hoogteename in heel wat regios niet zo significant, wat de successrate niet ten goede komt.

# **Hoofdstuk 3**

## **Setting aanval**

In dit hoofdstuk beschrijven we de setting alsook de werking van de aanval. De aanval is sterk gebaseerd op de aanvallen van Dhondt et al. en Verdonck [37, 6]. Deze aanvallen worden inferentieaanvallen genoemd, vanwege het feit dat uit metadata essentiële gegevens kunnen worden geïnfereerd. In het geval van Dhondt et al. gaat dit over afgelegde afstand binnenv de EPZ. In het geval van Verdonck gaat dit dan weer over geïnduceerde hoogteverschillen binnen de privacy zone. Allereerst zullen we kort de mogelijkheden van een aanvaller in de huidige setting bespreken. Daarna wordt de inferentieaanval van Dhondt et al., die de basis vormt voor de aanval in deze thesis, besproken volgens een opdeling in drie stappen.

### **3.1 Definitie aanvaller**

Deze thesis voert een onderzoek naar de mogelijkheid om een EPZ te omzeilen. De studie wordt dus gevoerd vanuit het opzicht van een aanvaller. Vooraleer we de werking van een aanval kunnen beschreven, is het belangrijk om een zicht te hebben op het doel en de capaciteiten van een aanvaller.

Hier is een aanvaller een gebruiker van het platform die geen eigenaar is van een activiteit. Hij heeft echter wel zicht op alle metadata die publiekelijk gedeeld is. Dit is data zoals afgelegde afstand, snelheid, tempo, ... Aangezien de aanval gaat over het omzeilen van EPZs worden enkel activiteiten beschouwd die gecloaked zijn. De aanvaller heeft dus geen zicht op de reële start- en/of eindlocatie. Zijn of haar doel is dan ook om ondanks de aanwezigheid van cloaking deze gevoelige locatie te achterhalen.

Vanuit het oogpunt van de inferentieaanval beschreven door Dhondt et al. heeft de aanvaller toe-

gang tot alle data die publiek beschikbaar is. In deze aanval wordt echter wel voornamelijk gebruik gemaakt van afstandsdata. De aanvaller die we in deze thesis beschrijven, heeft echter geen toegang tot deze afstandsdata. Hij heeft wel nog toegang tot de ruwe gps-data, maar ook de snelheid, het tempo enzovoort. Het onderzoek bestaat er dus uit om te zien in hoeverre een aanval nog mogelijk is wanneer de afstandsdata onbruikbaar zou zijn. We onderzoeken dus een alternatieve aanpak om de inferentieaanval alsnog succesvol te kunnen uitvoeren.

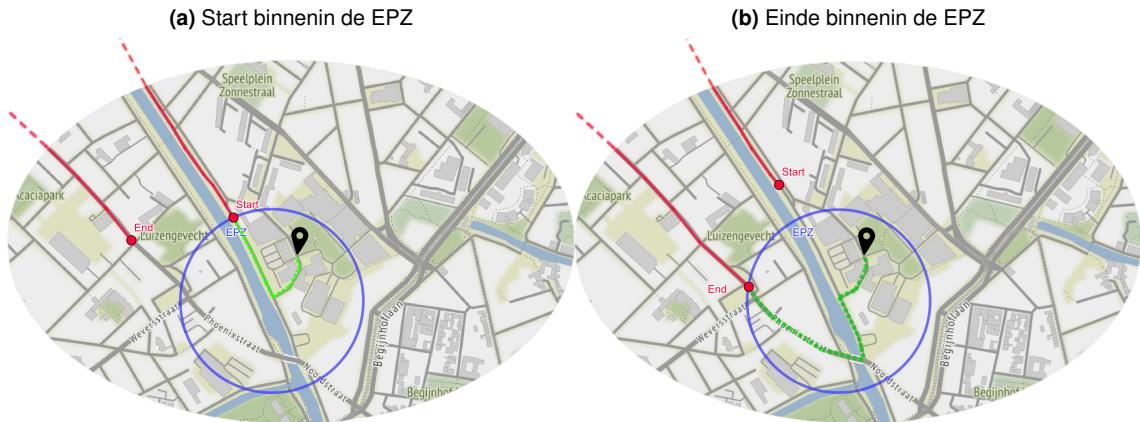
We beschrijven de aanvaller vanuit een theoretisch kader. We bestuderen onder welke omstandigheden de aanval mogelijk is, en welke maatregelen effectief blijken. Het globale scenario waar we van uitgaan is: ‘wat als fitnesstrackers de afstandsgegevens op een bepaalde manier zouden verbergen of onbruikbaar maken, door bijvoorbeeld afrondingen te maken, of onzekerheid toe te voegen’. Is de aanval dan nog mogelijk, en zo ja, hoe effectief is deze dan nog, en wat voor gevolgen heeft dit ten opzichte van eerder besproken beschermingsmaatregelen?

### 3.1.1 Assumpties

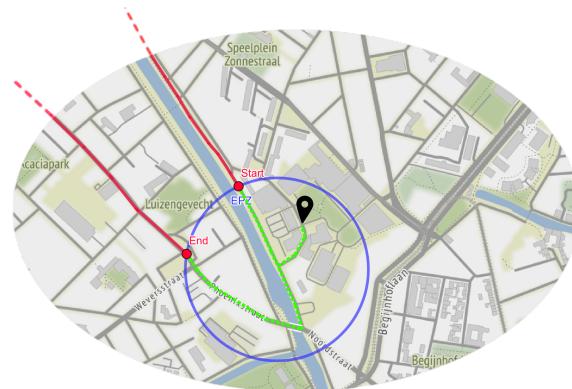
Om de aanval te kunnen uitvoeren, moeten enkele assumpties worden gemaakt. Dhondt et al. maakten al enkele assumpties om de inferentieaanval succesvol uit te voeren. Voor dit onderzoek is het dus logisch om dezelfde veronderstellingen te maken. Zo kunnen we een nuttige vergelijking maken. De eerste assumptie stelt dat de zichtbare begin -en eindpunten op de rand van de EPZ moeten liggen. Ten tweede moet de beschermde locatie op de roadgraph liggen. Hij kan niet buiten het voor ons te mappen gebied liggen, bijvoorbeeld in een bos waar geen pad in kaart is gebracht. Er wordt dieper ingegaan op de roadgraph in Sectie 3.4.1. Als laatste, maar desalniettemin belangrijk punt moet de gebruiker binnenvinden de EPZ de kortste route volgen [6].

Dhondt et al. maakte nog een laatste assumptie betreffende start- en eindpunten, meer bepaald dat deze die dezelfde locaties moeten voorstellen. Dit is echter niet van toepassing op dit onderzoek. Het onderzoek focust zich op activiteiten waar slechts één deel van het traject verborgen is. Dit wil dan ook zeggen dat de gebruiker ofwel vertrekt op de gevoelige locatie, of er eindigt, maar niet beide. Op Figuur 3.1 zijn de 2 mogelijke scenarios van een total distance attack terug te vinden, namelijk waarbij ofwel gestart als geëindigd wordt binnenvinden de zone. Dit wordt ook een *total distance attack* genoemd, omdat enkel de totale afstand en de afstand buiten de EPZ nodig is. Op deze figuur zijn de rode punten gelabeld *Start* en *End* de zichtbare start- en eindpunten. Dit scenario stelt dat één van de reële start- of eindpunten de gevoelige locatie is, aangeduid met de zwarte markering. Een andere aanval is de *inner distance attack*, hierbij zullen zowel de start als het einde van een activiteit binnenvinden het te verbergen gebied liggen, dit is te zien op Figuur 3.2. De kennis van de afzonderlijke afstand die de gebruiker aflagt van de effectieve startlocatie tot de rand van de EPZ en van de rand van de EPZ tot de effectieve eindlocatie is dan ook een vereiste. Op de Figuur zijn opnieuw de zichtbare randpunten aangeduid in het rood. Echter zal het onzichtbare

traject voor beide gevallen doorlopen en eindigen op de gevoelige locatie, wat in dit geval de reële start- en eindlocatie is. In Sectie 3.4.3 wordt dieper ingegaan op de reden waarom een *inner distance attack* niet mogelijk is. In deze thesis worden dus alle activiteiten waarvan enkel een verhulde start- of eindlocatie behouden, de rest wordt gefilterd in deze context.



**Figuur 3.1:** Voorbeeld van de mogelijke scenarios bij een total distance attack scenario



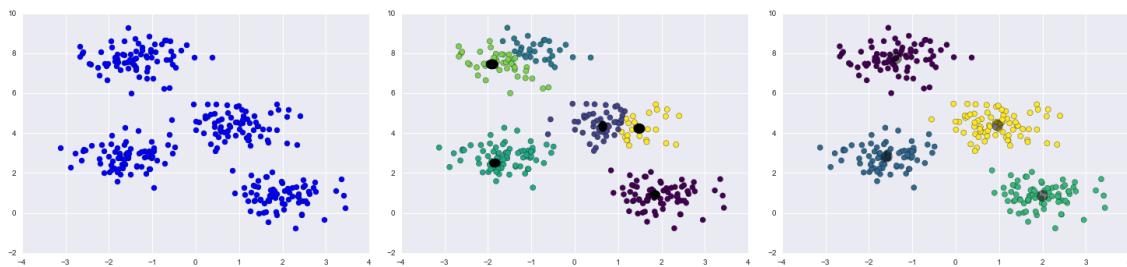
**Figuur 3.2:** Voorbeeld van een inner distance attack situatie

Deze thesis baseert zich ook voor een stuk op gemiddelde snelheden en tempo's. Hierdoor stellen we volgende bijkomende assumptie voor: een gebruiker mag niet stilstaan binnnenin de EPZ. Platformen zoals Strava hebben namelijk een ingebouwde functie die bij het uploaden van een activiteit tijdstippen waarbij een gebruiker stilstaat aan bijvoorbeeld een rood licht filtert. De gebruikers hebben op deze manier een representatieveer gemiddelde snelheid en gemiddeld tempo. Dit wil wel zeggen dat de totale bewegingstijd waarop de gemiddelde snelheid en tempo gebaseerd zijn, niet overeenkomt met de totale tijd van de activiteit. Bij een berekening gebaseerd op totale verstreken tijd zou een significante fout kunnen optreden.

## 3.2 Identificeren van de EPZ

De eerste stap is het identificeren van de EPZ. Alhoewel deze stap niet noodzakelijk is, vernauwt deze de zoekruimte drastisch. Hierbij nemen we van alle activiteiten die van een gebruiker ter beschikking zijn gesteld, de zichtbare begin- en eindpunten genomen. Deze zullen dan via het k-means algoritme worden gegroepeerd en een cirkel vormen.

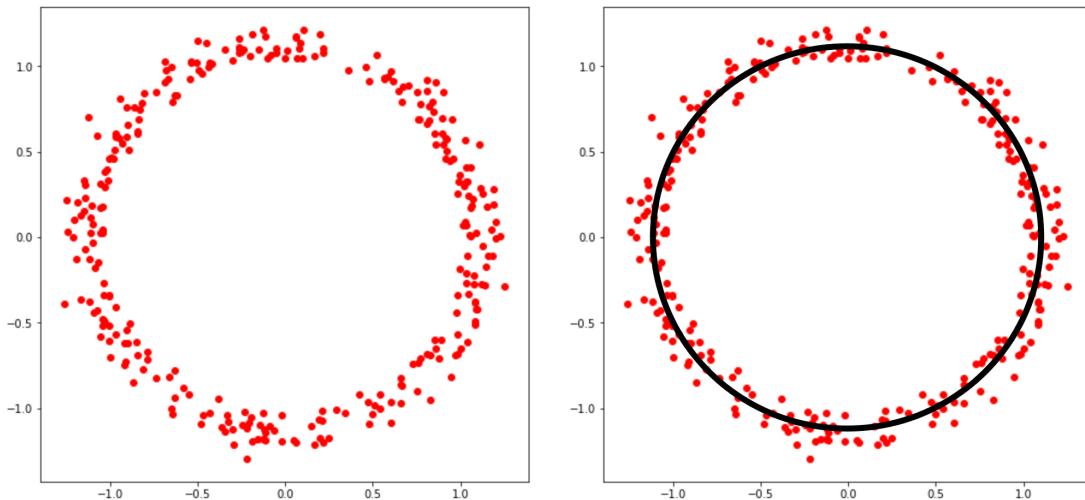
K-means clustering is een unsupervised machine learning techniek die veel wordt gebruikt bij het clusteren van data. Het is een iteratief proces waarbij het algoritme  $k$  clusters tracht te creëren waarbij de datapunten in elke cluster zo dicht mogelijk bij het gemiddelde van die cluster liggen [17]. Dit algoritme kiest willekeurig initiële middelpunten voor de verschillende clusters. Daarna kent het alle punten in de data toe aan de cluster met de laagste Euclidische afstand tot het centrum van deze cluster. Daarna herberekent het algoritme de gemiddeldes van deze clusters, en worden deze gemiddeldes gezien als nieuwe centrums. Opnieuw zal het alle punten aan de correcte cluster toekennen, en het proces herhaald zich verschillende iteraties op deze manier. Op Figuur 3.3 is te zien hoe de clustering bij elke iteratie beter wordt.



**Figuur 3.3:** Voorbeeld werking k-means clustering [21]

In de context van het identificeren van de EPZ gebruiken we k-means om gps-punten te groeperen op basis van hun locaties. In plaats van te werken met een centraal punt, werkt onze implementatie met een cirkelvormige zone, wat zichtbaar is op Figuur 3.4. De euclidische afstand zullen we in dit geval berekenen ten opzichte van de rand van deze cirkel. Bij elke iteratie beschouwen we een nieuwe cirkel, en berekenen we de afstanden opnieuw. We herhalen het proces totdat een stabiele cirkel bekomen wordt. Een cirkel is stabiel wanneer het verschil in afstand tussen twee opeenvolgende gevonden cirkels kleiner is dan een drempelwaarde, in dit geval 10 meter [6, 37].

Het algoritme zal na de identificatie van een EPZ ook nog nakijken of er niet meer dan één EPZ te vinden is. Het onderzoekt of punten die meegenomen zijn in de beschouwing van de huidige EPZ, toch niet horen bij een mogelijks andere EPZ van de user. Als controle berekent het van elk eind- of beginpunt de Euclidische afstand tot de rand van de bijhorende gevonden EPZ. Indien deze kleiner is dan de grootst mogelijke radius, dan veronderstellen we dat het punt bij deze zone hoort. Indien dit voor alle punten geldt, dan stopt het algoritme hier. In het andere geval waarbij



**Figuur 3.4:** Example of datapoints which can identify a circle [34]

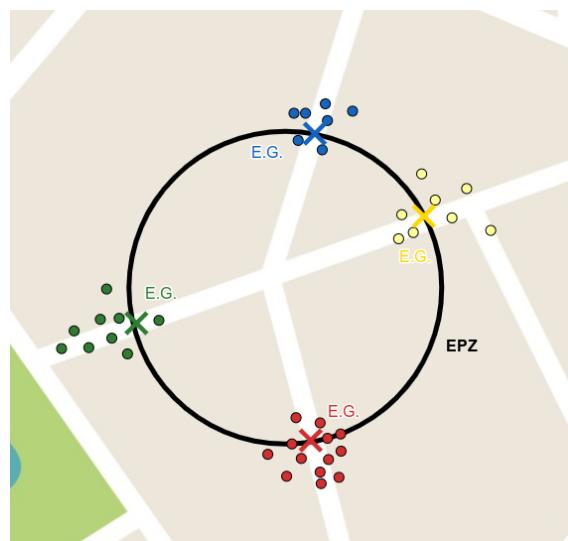
de berekende afstand groter is, worden meer clusters toegevoegd aan het algoritme van k-means clustering. Dit zal dus een nieuwe privacy zone aanwijzen.

Deze stap is niet noodzakelijk in het globale verhaal van de thesis, maar is wel een stap die de zoekruimte drastisch kan verkleinen. Indien het algoritme één of meerdere EPZs vindt, dan zullen er enkel en alleen voorspellingen gebeuren in de regio binnenin de geïdentificeerde zone. Indien dit niet het geval is en er geen EPZ gevonden is, bestaat de kans dat voorspellingen van locaties gebeuren buiten de verhullende zone. Ook is in dit geval een groter stuk van het stratenennetwerk nodig om de locatie te achterhalen.

### 3.3 Identificatie Entry Gates

Entry gates zijn zoals de naam al doet vermoeden de ‘toegangspoorten’ tot de EPZ. Dit zijn de ‘zones’ waar de gebruiker de EPZ kan betreden en/of verlaten. Deze vormen zich dan ook rond wegen die de EPZ betreden. Op Figuur 3.5 is te zien dat meerdere eindpunten van activiteiten geclusterd worden en een Entry Gate (E.G.) vormen. Dit is van belang belangrijk bij het filteren van afwijkende activiteiten. De detectie ervan gebeurt via het Density-Based Spatial Clustering of Applications with Noise (DBSCAN)-algoritme.

DBSCAN is een algoritme dat clusters kan vinden in een dataset. Het vindt clusters op basis van hun dichtheid [11]. In tegenstelling tot het eerder besproken k-means clustering algoritme dat gebaseerd is op het vinden van de geometrische centra van clusters, richt DBSCAN zich op het vinden van gebieden met een hoge dichtheid van punten. DBSCAN heeft als voordeel dat het goed overweg kan met uitschieters, want in deze context erg nuttig blijkt. Het werkt als volgt:



**Figuur 3.5:** Voorbeeld van entry gates gevonden door k-means clustering en de identificatie van een EPZ

1. Selecteer een willekeurig niet-bezocht punt uit de set van zichtbare begin- en eindpunten.
2. Bepaal of het punt een kernpunt is door te controleren of er binnen een bepaalde afstand een minimum aantal punten aanwezig is. Als dit het geval is, wordt het punt als een kernpunt beschouwd en wordt een nieuwe cluster gestart.
3. Breid de cluster uit door alle punten binnen deze afstand van het kernpunt toe te voegen aan de cluster. Herhaal dit proces voor alle punten in de buurt totdat er geen nieuwe punten meer kunnen worden toegevoegd.
4. Ga door naar het volgende niet-bezochte punt en herhaal de stappen 2 en 3 totdat alle punten in de dataset zijn bezocht.
5. Punten die niet tot een cluster behoren en niet voldoen aan de criteria voor kernpunten, worden beschouwd als ruispunten.

Het algoritme maakt gebruik van twee op voorhand vast te leggen parameters: de maximale afstand tussen twee punten in eenzelfde cluster, wat in deze context vastgelegd is op 22.9 meter en een minimaal aantal punten dat de cluster moet bevatten, in dit geval één punt.

### 3.4 Bepalen nodige gegevens voor predictie

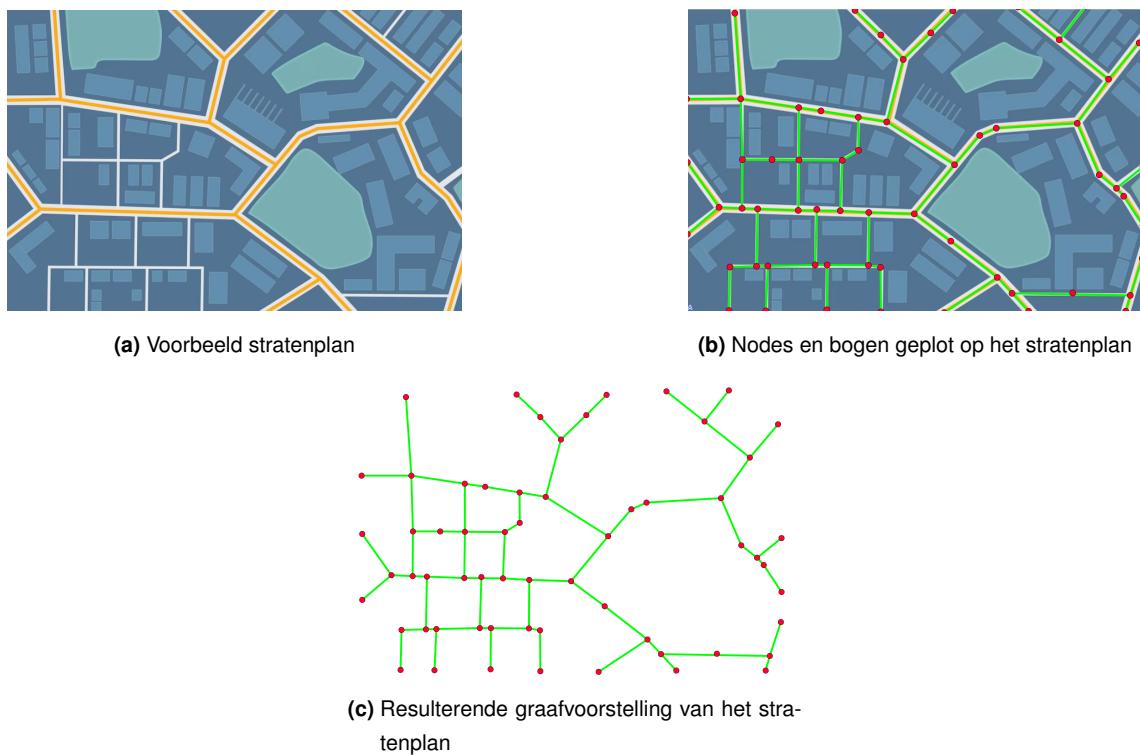
Na het bepalen van de EPZs van de gebruiker gaan we over tot het berekenen en achterhalen van de bijhorende gegevens die nodig zijn om de gevoelige locatie te voorspellen. Hiervoor wordt

verder ingegaan op de methodiek beschreven inferentieaanval door Dhondt et al., maar er worden enkele gegevens op een andere manier benaderd volgens de huidige definitie van de aanvaller.

### 3.4.1 Roadgraph en Distance Matrix

Voor elke gevonden EPZ is het noodzakelijk om een graafvoorstelling van de omgeving op te stellen. Op Figuur 3.6 is een voorbeeld terug te vinden van hoe een graaf kan worden geëxtraheerd. Er worden punten geplaatst op de straten op een vaste afstand van elkaar<sup>1</sup>, en deze kunnen dan worden verbonden. Indien geen EPZs geïdentificeerd zijn, nemen we een ruime omgeving die we omzetten naar een graaf. De graafvoorstelling bestaat uit een serie van nodes, die zich allemaal op een gekende straat bevinden. De bogen waarmee de nodes verbonden zijn volgen het straatplan, opdat een boog een mogelijk te volgen weg is [19]. Aan de hand van de ‘Chaining Distance’ wordt bepaald hoeveel afstand tussen de nodes zal zitten, en zo dus impliciet ook welke densiteit het netwerk zal hebben. Hoe lager de chaining distance, hoe meer nodes, en dus ook hoe preciezer de graafvoorstelling zal zijn. Om voorspellingen te maken is wel een bepaalde precisie vereist, dus mag deze waarde niet te hoog zijn. Empirisch koos Dhondt et al. voor een waarde van 3.0 meter.

**Figuur 3.6:** Voorbeeld van het genereren van een roadgraph<sup>1</sup>



<sup>1</sup>Op de Figuur 3.6 zijn de afstanden niet altijd gelijk. De figuur is enkel nuttig ter illustratie, en is geen perfecte representatie de werkelijkheid.

Op basis van de nodes in deze graaf, kan de *Distance Matrix* worden opgesteld. Dit is een matrix die voor alle startnodes (op de rand van de EPZ) de theoretische afstand bevat tot alle nodes aanwezig in de graaf. Gebruikmakend van het Dijkstra algoritme<sup>2</sup>, die het in staat stelt om voor elk punt de kortste theoretische afstand te bepalen tot alle punten in de graaf. Deze afstanden worden opgeslagen, en zijn belangrijk in een verder stadium van de aanval.

### 3.4.2 Begin- en eindnodes

Voor elke activiteit is het volledige traject buiten de EPZ gegeven. Dit omvat alle gps-punten die niet verborgen zijn. De begin- en eindnodes van het traject zijn hier van belang. Voor de duidelijkheid en de vlotheid van de tekst zullen we naar deze punten refereren als het zichtbare beginpunt en het zichtbare eindpunt. Volgens één van de voorafgaand gemaakt assumptie vertrekt de sporter in de EPZ of eindigt hij erbinnen, maar beide is geen mogelijkheid. Dit betekent dat ofwel het reële eindpunt, ofwel het reële beginpunt zal overeenstemmen met de gevoelige locatie. In geval dat een gebruiker aankomt binnenin de EPZ, en dus ook vertrekt erbuiten, starten de berekeningen vanaf het zichtbare eindpunt. Omgekeerd geldt indien de gebruiker vertrekt binnenin de EPZ, worden de berekeningen gestart vanaf het zichtbare beginpunt. Deze punten zullen in het vervolg *randpunten* genoemd worden, refererend naar de rand van de EPZ. Deze randpunten zullen de basis vormen voor de volgende berekeningen.

Bijhorend zijn bij de randpunten ook bepaalde extra gegevens beschikbaar. De belangrijkste zijn de cumulatieve afstand tot dit punt<sup>3</sup>, en de cumulatieve tijd tot dit punt<sup>4</sup>. Bij de aanval van Dhondt et al. wordt de afstand gebruikt om predicties te doen. Dit wil dus zeggen dat deze afstand dus aan de basis zal liggen. In deze thesis wordt ervan uitgegaan dat afstanden verborgen worden. Onder het verbergen van afstanden wordt een onderscheid gemaakt tussen 2 scenarios: het eerste gaat ervan uit dat de totale afstand verborgen wordt, maar de cumulatieve afstand gegeven is. Het tweede scenario gaat ervan uit dat alle afstandsgegevens verborgen worden. Het alternatieve type data waar dus mee zal moeten gewerkt worden is dus gps-data.

### 3.4.3 Berekeningen afstand binnenin de EPZ

Om voorspellingen te kunnen doen zullen volgens de inferentieaanval die we hier bespreken, moeten twee belangrijke gegevens ter beschikking zijn. Met name het straatnetwerk met de mogelijks

<sup>2</sup>Het Dijkstra-algoritme is een algoritme in de grafentheorie dat wordt gebruikt om de kortste weg te vinden tussen twee knooppunten in een gewogen grafiek [15]. Het algoritme werkt door iteratief knooppunten toe te voegen aan een 'bezochte' set en de kortste afstand te berekenen vanaf het beginpunt naar elk aangrenzend knooppunt dat nog niet is bezocht.

<sup>3</sup>De totale afstand afgelegd vanaf het begin van de activiteit tot en met het punt in kwestie.

<sup>4</sup>De totale tijd afgelegd vanaf het begin van de activiteit tot en met het punt in kwestie.

gevolgde routes, wat werd besproken in Sectie 3.4.1, en de afstand die wordt afgelegd binnenvin de EPZ. Deze afstand benoemen we ook als de *inner distance* (dit is niet te verwarren met de *inner ditstance attack*).

In de implementatie van Dhondt et al. kan de *inner distance* simpelweg berekend worden door het verschil te nemen tussen de afgelegde afstand buiten de verhulde zone (deze noemen we de *outer distance*), en de totale afstand:

$$\text{inner distance} = \text{total distance} - \text{outer distance}$$

In deze thesis is er echter een tussenstap noodzakelijk. In het eerste scenario waarbij de cumulatieve afstand gegeven is, maar de totale afstand niet, moet de totale afstand berekend worden. Maar door de aanwezigheid van snelheids- en tijdsgegevens kan dit via basisformules gebeuren. Gebruik makend van het gemiddelde tempo kan de voorgaande formule worden omgevormd tot:

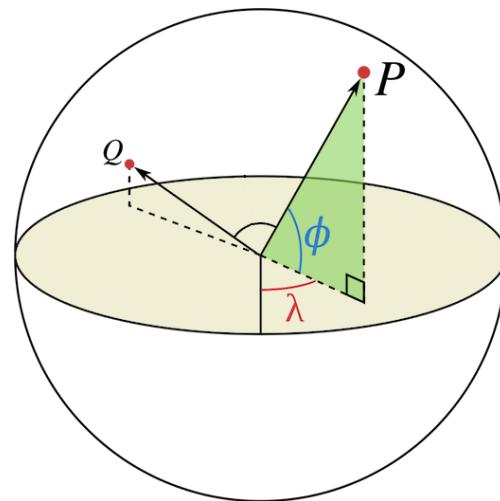
$$\text{inner distance} = \text{total time} \times \text{average speed} - \text{outer distance}$$

In opzicht van het tweede scenario, waarbij alle afstandsgegevens verborgen zijn, ontbreekt nu ook de *outer distance*. We bepalen deze via de gps-coördinaten. Dit gebeurd door de som te nemen van de afstanden van tussen alle opeenvolgende punten. Let wel dat we de afstand tussen twee gps-punten berekenen door gebruik te maken van de *haversine* formule. Vergelijking 3.1 is een uitwerking van deze formule [24]. Deze berekent de afstand tussen twee punten op een bolvormig oppervlak, in dit geval de aarde. De breedte- en lengtegraden van elk punt moeten omgezet worden naar radialen. Vervolgens worden deze waarden ingevoerd in de formule, samen met de straal van de aarde ( $r$ ), meestal genomen als  $6.371\text{km}$ . De formule berekent dan de haversine ( $\text{haversine}(\theta) = \sin^2\left(\frac{\theta}{2}\right)$ ) van de helft van het verschil tussen de breedtegraden en de haversine van de helft van het verschil tussen de lengtegraden ( $\lambda$ ), evenals de cosinus van de breedtegraden ( $\phi$ ) van beide punten. Deze waarden worden vervolgens gebruikt om de afstand tussen de twee punten (P & Q op Figuur 3.7) te berekenen.

Merk op dat ook dit een benadering is van de werkelijke afstand. De aarde is niet perfect sferisch, wat de nauwkeurigheid kan beïnvloeden. Maar voor de doeleinden binnen deze thesis is dit voldoende nauwkeurig, zeker omdat de afstanden in deze context relatief klein zijn, waardoor over het algemeen slechts een minimale buiging is.

$$d = 2r \arcsin \left( \sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cos(\phi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)} \right) \quad (3.1)$$

Uit de voorgaande paragrafen kunnen we dus besluiten dat de inner distance af te leiden valt uit gegeven outer distance, total time en de gemiddelde snelheid. Om een *outer distance attack* uit



**Figuur 3.7:** Haversine illustratie voor het berekenen van de afstand [39]

te voeren is de berekening van de totale inner distance voldoende. Maar bij het uitvoeren van een *inner distance attack* zijn de twee afzonderlijke inner distances nodig (degene van start tot de EPZ en degene van de EPZ tot de finish). Wanneer de cumulatieve afstand gegeven is, zouden we een deze aanval kunnen uitvoeren doordat in dit geval de twee afstanden te achterhalen zijn.  $d_{start} = d_{eerste\ node}$  en  $d_{finish} = d_{totaal} - d_{laatste\ node}$ . Maar wanneer deze niet beschikbaar zijn, is dit niet mogelijk, in dit geval zijn deze afstanden niet individueel te achterhalen.

## 3.5 Voorspellen locatie

Alle nodige gegevens zijn nu beschikbaar om de gevoelige locatie te achterhalen. Hier wordt besproken hoe voor elke bruikbare activiteit een locatie zal worden voorspeld. Doordat voor elke activiteit één of meerdere locaties worden voorspeld, zullen we deze moeten bundelen tot één locatie voor alle activiteiten.

### 3.5.1 Filteren activiteiten

Voorafgaand aan het voorspellen is het van essentieel belang om uitsluitend voorspellingen te maken op basis van activiteiten die een waardevolle voorspelling kunnen genereren. Andere activiteiten zouden enkel de accuraatheid van de voorspelling naar beneden halen. Dit gaat dan bijvoorbeeld over activiteiten waarbij niet de kortste route binnendoor de EPZ wordt gevuld. Al deze activiteiten proberen we dus in de mate van het mogelijke eruit te filteren.

Het geval waarbij een gebruiker niet de kortste route volgt vanaf de rand van de EPZ tot de gevoelige locatie kunnen we in zekere mate opvangen door te stellen dat we een activiteit enkel nog gebruiken wanneer de nog af te leggen afstand binnenin de EPZ kleiner is dan de maximaal mogelijk af te leggen weg. In de andere gevallen zullen we de activiteit weggooien. De maximale afstand die hiervoor nodig is wordt bepaald gebruik makend van de *Distance Matrix*, die beschreven staat in Sectie 3.4.1. De maximale afstand is gelijk aan de maximale afstand terug te vinden in de matrix, voor de bijhorende startnode. Dit is de afstand die een gebruiker maximaal kan afleggen naar eender welke node op de graaf, vertrekend van de startnode, door het volgen van de theoretisch kortste route. Dit zal ook gevallen in rekening brengen waarbij het traject voor een stuk verborgen wordt, maar niet zal eindigen op de gevoelige locatie.

Op een gelijkaardige manier kan een filtering gebeuren voor afgelegde afstanden die lager zijn dan de minimale mogelijke afstand. Dit zou opnieuw activiteiten kunnen filteren die niet eindigen op de gevoelige afstand. De minimale afstand is dan ook degene tot de node met de laagste minimale afstand tot deze node vanaf het zichtbare startpunt of eindpunt van de activiteit, gelegen op de rand van de EPZ.

Om de zichtbare eind- en beginpunten van de activiteiten te controleren op compatibiliteit met de road graph, voeren we een actieve verificatie uit. Hierbij worden alle eind- en beginpunten op de graaf ‘gesnapt’, ofwel vervangen door de dichtstbijzijnde knooppunt op de graaf. We berekenen de afstand tussen de oorspronkelijke locatie en de gesnapte locatie. Indien het verschil in afstand te groot is, filteren we de betreffende activiteit. Een aanzienlijk verschil in afstand kan duiden op een afwijking tussen de gevolgde route en de road graph, of op onnauwkeurige gps-gegevens.

Als laatste wordt nog gekeken naar afwijkingen bij de E.G.’s. Indien bij een activiteit een afwijking wordt vastgesteld tussen de zichtbare begin- en eindpunten en de E.G. die groter is dan drie maal de standaardafwijking, wordt de activiteit gefilterd. Dit wijst dan op een te groot verschil ten opzichte van de E.G., waar het grotendeels van de andere activiteiten vertrekt of aankomt, en dus op een grote kans op inaccurate voorspellingen.

### 3.5.2 Bepalen van de locatie

Om een predictie te maken per activiteit wordt de inner distance die berekend werd in Sectie 3.4.3 gebruikt. Deze wordt dan gematched met het stratenennetwerk. Het idee hierachter is om alle mogelijke routes (die de kortste route vormt naar alle nodes op zijn pad) binnenin de EPZ af te leggen, en te stoppen wanneer de afgelegde afstand gelijk is aan de berekende inner distance. Het resultaat van deze gevolgde route is dan een node, die mogelijks de gevoelige locatie kan voorstellen. In de praktijk kunnen we dit mechanisme op een simpele manier toepassen door gebruik te maken van de vooraf berekende distance matrix. De berekende inner distance zal worden vergeleken met de

afstanden in de distance matrix.

Deze methodiek herhalen we voor elke activiteit die niet werd gefilterd. Al deze voorspellingen worden uiteindelijk gebundeld tot één voorspelling in de volgende stap.

### 3.5.3 Regressie om te komen tot een eindvoorspelling

Om de routes die we in vorige sectie bepaalden om te vormen tot één eindvoorspelling, voeren we een regressie-analyse uit, aan de hand van de Least Absolute Deviations (LAD) estimator. Het resultaat van deze regressie-analyse zal een gps-locatie zijn, die onze *eindvoorspelling* zal vormen.

De LAD estimator wordt gebruikt om een lineaire regressie uit te voeren op een set punten, door de som van de absolute waarden van de absolute verschillen te minimaliseren. LAD staat gekend als een robuuste methode die erg nuttig blijkt te zijn voor datasets met grote uitschieters. In Vergelijking 3.2 is te zien dat, door het werken met absolute waarden, extremen in mindere mate doorwegen in de berekeningen. Dit is een groot voordeel ten opzichte van andere regressietechnieken, zoals bijvoorbeeld Ordinary Least Squares (OLS) [14]. OLS werkt gelijkaardig, maar zoals te zien is in Vergelijking 3.3 probeert deze de som van de kwadratische afwijkingen te minimaliseren. Uitschieters zullen dus meer doorwegen. Een nadeel van LAD is dat het berekenen van de LAD-schattingen meer rekentijd en computerbronnen vereist dan OLS, wat het minder geschikt maakt voor grote datasets.

$$LAD : \min \sum_{i=1}^n |d_i - d_{inner}| \quad (3.2)$$

$$OLS : \min \sum_{i=1}^n (d_i - d_{inner})^2 \quad (3.3)$$

In deze context gebruiken we LAD regressie door de aard van de data en voorspellingen, meer bepaald door de grote kans op uitschieters. Gps-data kan erg onnauwkeurig zijn, en grote of kleine afwijkingen kunnen dus voorkomen. Ook is het mogelijk dat door acties van een sporter zoals bijvoorbeeld eenmalig de kortste route niet volgen, uitschieters voorvallen. Rekentijd is in deze thesis geen probleem, aangezien de dataset beperkt blijft tot 100 activiteiten.

In de context van deze thesis kan Vergelijking 3.2 worden toegepast voor elke node in de graaf. Elke node in graaf zullen we individueel beschouwen als een mogelijke eindvoorspelling. Het verschil tussen de nog af te leggen afstand voor deze activiteit, of dus de inner distance en de theoretische afstand tot de node vanaf het zichtbare begin- of eindpunt, wat af te lezen valt uit de distance matrix, stelt dan de afwijking voor van deze activiteit. Indien we dit sommeren over alle activiteiten voor dezelfde node, bekomen we een waarde die de totale afwijking voorstelt indien we voor deze node als eindvoorspelling zouden kiezen. We overlopen alle nodes in de graaf, en selecteren deze met de laagste totale afwijking. Deze node zal dan de **eindvoorspelling** voor deze activiteit voorstellen.

## **Hoofdstuk 4**

# **Analyse van de gebruikte data**

We testen de aanval beschreven in Hoofdstuk 3 op een dataset met activiteiten van een aantal gebruikers, om zo de kwetsbaarheid van gebruikers van fitnessplatformen te evalueren. Maar het is dan ook essentieel dat een representatieve dataset wordt gebruikt. We onderzoeken eigenschappen en mogelijke afwijkingen of onregelmatigheden opdat we een gefundeerde conclusie kunnen vormen, die eventueel bepaalde eigenschappen van de aard van de data mee in rekening brengt.

Aangezien deze thesis voor een stuk verder bouwt op het onderzoek van Dhondt et al., is het handig om verder te werken met deze dataset [6]. Dit maakt een directe vergelijking mogelijk. Deze dataset werd volledig gescraped door Dhondt et al. vanaf de officiële Application Programming Interface (API) ([strava.com/stream/ID](https://strava.com/stream/ID)) van het platform Strava<sup>1</sup>. De scope van deze dataset is een periode van één week, startend op 11 juli 2021 00:00 Coordinated Universal Time (UTC). De site werd chronologisch afgelopen voor alle activiteiten beschikbaar op het platform, met sprongen van 9000 activiteiten. Let wel dat door mogelijke vertragingen door bijvoorbeeld het uploaden van een activiteit, de activiteiten niet exact chronologisch kunnen worden opgehaald. Indien een activiteit privaat is, of reeds verwijderd is, dan zal deze worden overgeslagen en de volgende worden genomen. Voor elke gevonden activiteit beschouwen we daarna de gebruiker. Van alle bekomen gebruikers worden dan de rest van de activiteiten afgehaald en bijgehouden in één grote dataset. De gegevens werden ook geanonimiseerd opgeslagen, zodat de gebruikers niet meer kunnen worden geïdentificeerd. De dataset bevat dus geen namen of andere persoonlijke gegevens, enkel willekeurig toegekende ID's.

Deze thesis experimenteert echter slechts met een subset van 131 users, waarvan er 101 gebruikt worden voor analyses en conclusies, en 30 voor het testen van de aanval. Let wel, de fractie van de dataset die wij ter beschikking kregen is met 101 gebruikers wel relatief klein, wat een vertekend beeld kan geven over de werkelijkheid.

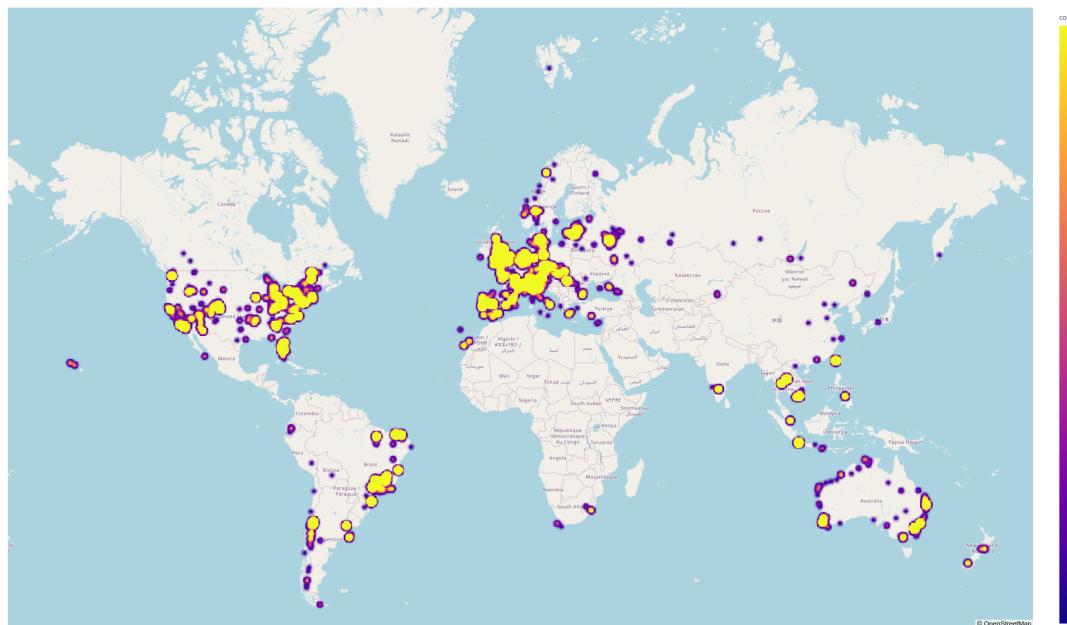
---

<sup>1</sup><https://www.strava.com/>

## 4.1 Karakteristieken van de gebruikte dataset

Op Figuur 4.1 is de geografische spreiding van de activiteiten gevisualiseerd aan de hand van een heatmap. Hierop is duidelijk te zien dat de meeste activiteiten zich in Centraal-Europa bevinden. Daarnaast is ook een duidelijke concentratie te zien in de Verenigde Staten. In mindere mate zijn ook activiteiten in Australië en Zuid-Amerika. De dataset bevat dus een relatief grote spreiding van activiteiten over de hele wereld, wat een goede basis is voor het testen van de aanval.

Op Tabel 4.1 zijn enkele globale statistieken met betrekking tot gebruikers en de bijhorende activiteiten van de dataset weergegeven. Er valt op dat de dataset per gebruiker toch meestal een groot aantal activiteiten ter beschikking zijn. De gemiddelde gebruiker bevat 411 activiteiten, de mediaan is 296. Volgens de inferentieaanval beschreven in Hoofdstuk 3 resulteert een gebruiker met meer activiteiten over het algemeen in een aanval met een hogere kans op slagen. Op Figuur 4.2 is de Cumulative Distribution Function (CDF) plot<sup>2</sup> te zien die het aantal activiteiten per gebruiker weer geeft. Hierop worden voorgaande besluiten enkel maar bevestigd. Het plot duidt ook aan dat meer dan 20% van de gebruikers een aantal activiteiten heeft dat groter is dan 100.



**Figuur 4.1:** Geografische spreiding van de activiteiten in de dataset

<sup>2</sup>Een Cumulative Distribution Function (CDF) plot is een grafiek die de cumulatieve verdeling van de waarden van een continue variabele weergeeft [3]. De x-as van de grafiek bevat de verschillende waarden die de continue variabele kan aannemen, terwijl de y-as de kans aangeeft dat de variabele een waarde kleiner dan of gelijk aan die op de x-as aanneemt. De curve van de CDF laat zien hoe waarschijnlijk het is dat een willekeurige waarde van de continue variabele kleiner is dan een bepaalde drempelwaarde.

	Aantal
Totaal # gebruikers	101
Totaal aantal activiteiten	41 554
Gemiddeld # activiteiten per gebruiker	411
Mediaan van het # activiteiten per gebruiker	296
Maximaal # activities voor een enkele gebruiker	2946
Minimaal # activities voor een enkele gebruiker	31

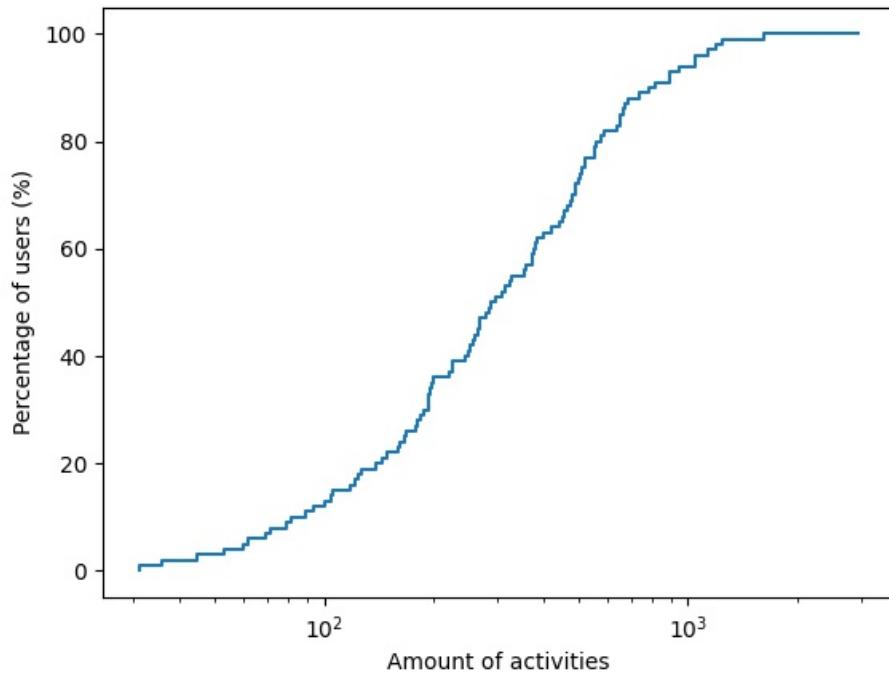
**Tabel 4.1** Overzicht van gebruikers en activiteiten

## 4.2 Mogelijke afwijkingen binnenin de dataset

Doordat de dataset niet expliciet werd gecheckt op onnauwkeurigheden en een willekeurige sample is, is er een grote kans op activiteiten die afwijkingen of fouten vertonen. Zeker door het belang van gps-data in deze studie, die een grote kans heeft op fouten, is het belangrijk om de dataset te analyseren op mogelijke afwijkingen. Gps-data is een signaal die a.d.h.v. gekende locaties van satellieten, gecombineerd met de tijd die het signaal nodig heeft om vanaf de satelliet het toestel te bereiken, de locatie van een gebruiker kan bepalen [31]. Door de hoge snelheid van het signaal, kunnen kleine vertragingen in het signaal al een grote invloed hebben op de accuraatheid van de data. Andere factoren zoals hoge bomen of gebouwen, maar ook de aanwezigheid van wolken kunnen een impact hebben op het signaal. Ook de het interval waartussen opeenvolgende locaties worden bepaald, wat afhankelijk is van het gebruikte toestel, kan meespelen. De soorten gps-fouten die kunnen optreden zijn reeds besproken in Sectie 2.2.

Allereerst bestuderen we de aanwezigheid van gps-fouten in de vorm van signal losses of pauzes. Dit gebeurt door de afstand tussen twee opeenvolgende gps-punten te bestuderen. Tabel 4.2 geeft een globaal overzicht van deze verdeling, en de volledige verdeling is terug te vinden op Figuur 4.3. De gemiddelde afstand tussen twee opeenvolgende locaties is 6.41 meter, met een standaardafwijking van 42.53 meter. Het gemiddelde is relatief laag, wat kan wijzen op accurate gegevens, maar de hoge standaardafwijking wijst op grote schommelingen. Op de grafiek en in de tabel is te zien dat de meeste afstanden onder de 20 meter liggen, wat opnieuw een indicatie kan zijn van een degelijke precisie. Er is echter wel een klein deel van de gps-punten die een grote onderlinge afstanden vertoond. Door de omvang van het aantal gps-punten, en een gemiddeld aantal punten per activiteit van 2574.90, valt dit zeker niet te verwaarlozen. Als we empirisch stellen dat een significant verschil 150 meter bedraagt, wat ongeveer drie maal de standaardafwijking voorstelt, dan ligt 0.2% van de data boven deze drempel. Per activiteit zou dit dan resulteren op gemiddeld 5.1 afwijkende punten, wat zeker al kan zorgen voor een significante afwijking op de resulterende afstand.

Om het aantal gps-afwijkingen in de dataset te bepalen, wordt ook het verschil onderzocht tus-



**Figuur 4.2:** CDF plot van het aantal activiteiten per gebruiker

sen de berekende afstand afgelegd binnen de EPZ (het zichtbare traject afgetrokken van de totale afstand) en de theoretisch afgelegde afstand binnen de EPZ, die af te lezen valt uit de dataset via de cumulatieve afstand<sup>3</sup>. Een eerste visualisatie is te zien op Figuur 4.4. De figuur illustreert de schommelingen tussen de handmatig berekende afstand en de theoretische afstand van één gebruiker. De pieken duiden op sterk afwijkende berekende afstanden, en dus ook op grote gps-fouten. Maar ook de schommelingen die iets minder opvallend zijn duiden op grote inaccuraaties tussen de berekende en theoretische afstanden. De verschillen in de berekeningen voor de volledige dataset worden weergegeven op Figuur 4.5. De figuur bevat de CDF-verdeling van de verschillen voor alle activiteiten, gebruik makend van een logaritmische schaal. Op de grafiek valt op dat heel wat significante verschillen aanwezig zijn. Dit duidt op het relatief zwaar doorwegen van de gps-fouten in de dataset. Aangezien het gaat over bepalen van woonplaatsen of andere gevoelige locaties, kunnen afwijkingen vanaf 50 à 100 meter al sterk doorwegen zijn. Daarnaast gaat het vaak over kleine afwijkingen op een heel wat punten, wat kan resulteren in een grote afwijking. De grafieken tonen aan dat er bij de ruwe ontvangen data heel wat gps-fouten aanwezig zijn. Smoothing zal dus zeker nodig zijn om deze te beperken.

<sup>3</sup>Er wordt gesproken van een theoretische waarde, maar deze is eigenlijk de berekende waarde volgens het platform. We beschouwen deze dus als referentie.

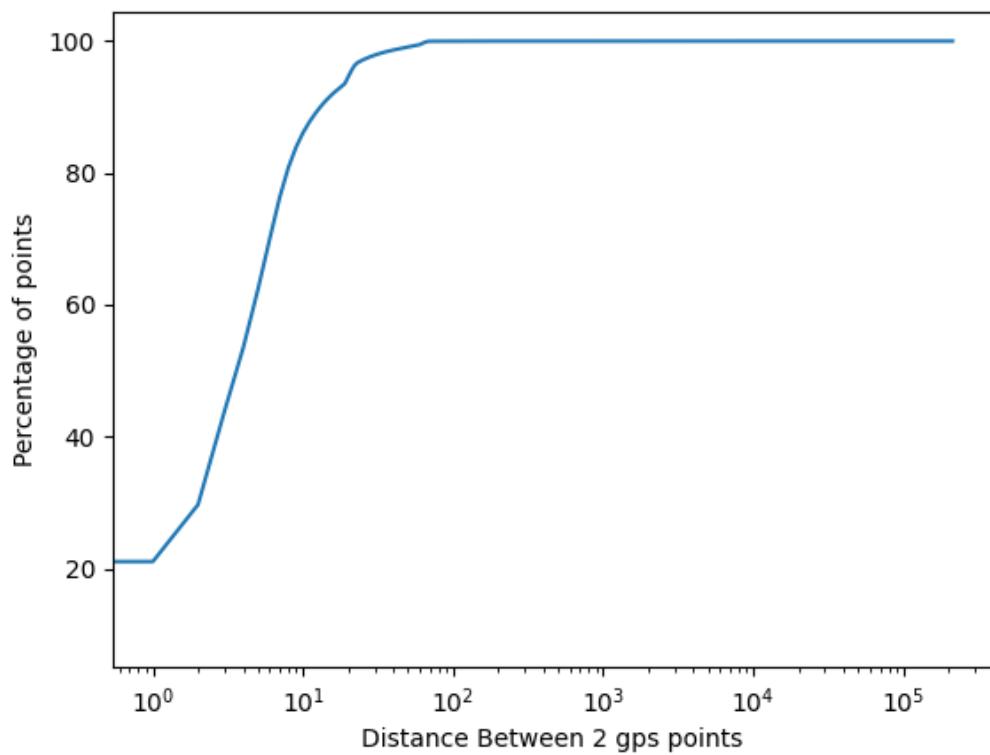
Total number of gps-points	$1.070 \cdot 10^8$
distance between 2 gps points 20m > 10m	13.91%
distance between 2 gps points 75m > 50m	$8.80 \cdot 10^{-1}\%$
distance between 2 gps points 100m > 75m	$9.11 \cdot 10^{-3}\%$
distance between 2 gps points 150m > 100m	$4.99 \cdot 10^{-3}\%$
distance between 2 gps points 200m > 150m	$1.97 \cdot 10^{-3}\%$
distance between 2 gps points 500m > 200m	$3.47 \cdot 10^{-3}\%$
distance between 2 gps points 1000m > 500m	$1.26 \cdot 10^{-3}\%$
distance between 2 gps points 2000m > 1000m	$9.39 \cdot 10^{-4}\%$
distance between 2 gps points > 2000m	$4.056 \cdot 10^{-4}\%$

**Tabel 4.2** Verdeling van de afstanden tussen twee opeenvolgende gps-punten

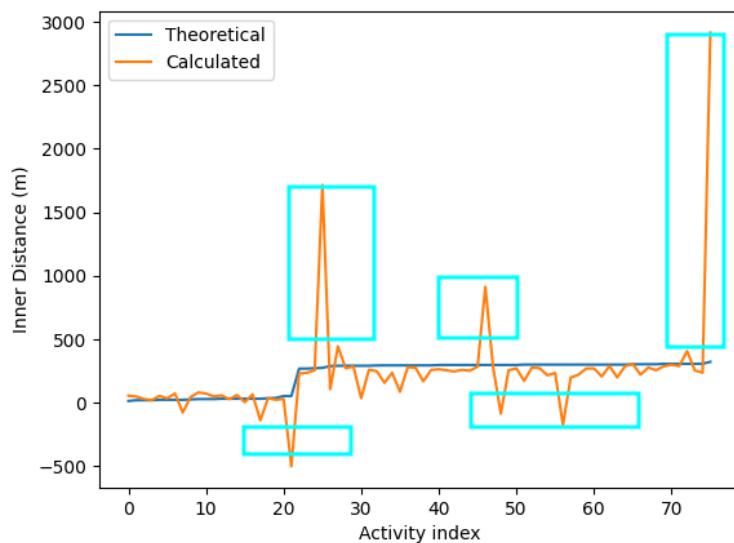
### 4.3 Technieken om gps-data te verbeteren

Om de accuraatheid van de gps-data te verbeteren, en zo een betere *outer distance* te kunnen berekenen en uiteindelijk een accuratere aanval te bekomen, worden enkele technieken toegepast. Zoals besproken in Sectie 2.1.2 is de hypothese dat de fitnessplatformen ook gebruik maken van gelijkaardige technieken om de gps-data te verbeteren. De besproken technieken zijn *map matching* en *gps-smoothing*. Bij de uitvoering van de aanvallen wordt smoothing toegepast. Er wordt dan ook geëxperimenteerd met verschillende smoothing windows, op zoek naar het window met het beste effect op de aanval.

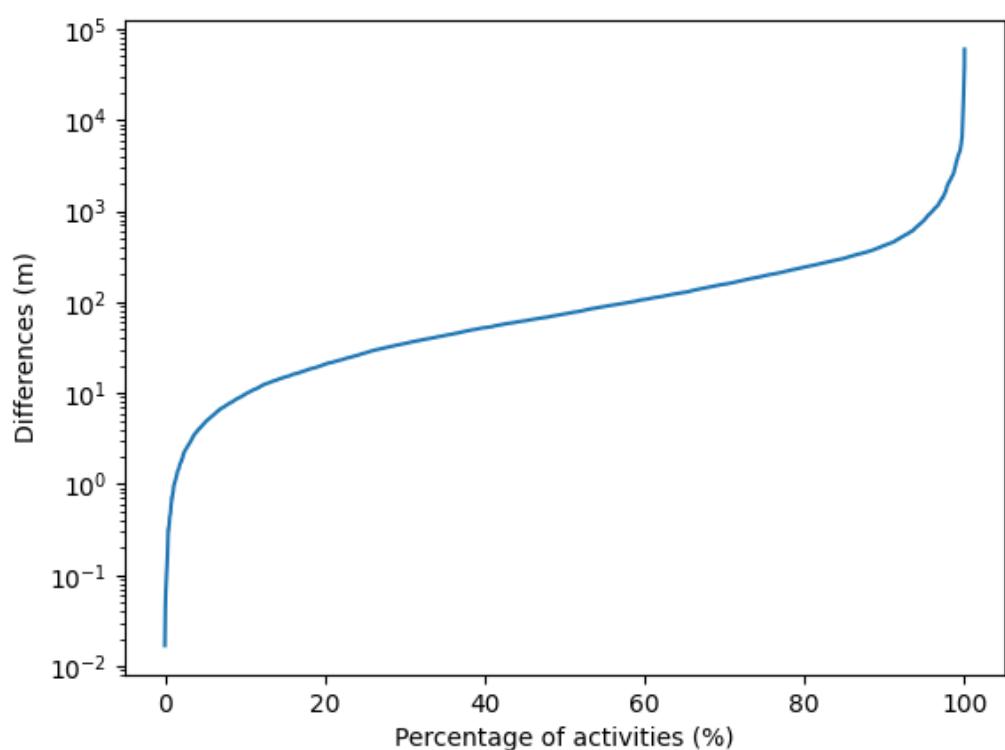
Daarnaast passen we ook een extra filtering toe die rekening houdt met gps-sprongen. Het idee is te stellen dat wanneer gps-sprongen gebeuren, en er dus een te grote afstand tussen twee opeenvolgende punten is, deze afstand niet te laten meetellen. Dit is echter niet vanzelfsprekend, aangezien dit waarschijnlijk wel op een andere manier in rekening gebracht wordt door de platformen, bijvoorbeeld door map matching. Stel dat wij een sprong van 150 meter laten vallen, maar de platformen laten deze wel meetellen, dan zullen we de afwijking op het eindresultaat enkel maar verhogen. De drempel voor de filtering moet dus hoog genoeg zijn om deze voorvalen te vermijden. In dit onderzoek kozen we voor een drempelwaarde van 200 meter.



**Figuur 4.3:** Verdeling van de afstanden tussen twee opeenvolgende gps-punten



**Figuur 4.4:** Verschil tussen de berekende afstand en de theoretische afstand voor één gebruiker



**Figuur 4.5:** Verdeling van het verschil tussen de berekende afstand en de theoretische afstand buiten de EPZ

## **Hoofdstuk 5**

# **Resultaten en Evaluatie**

Het aanvalsmodel werd tot hiertoe al volledig beschreven en toegelicht, net als de dataset die gebruikt wordt om de aanval uit te testen. Dit hoofdstuk bespreekt hoe de evaluatie van het aanvalsmodel wordt aangepakt, wat de bekomen resultaten zijn en wat ze betekenen.

### **5.1 Evaluatie van de aanval**

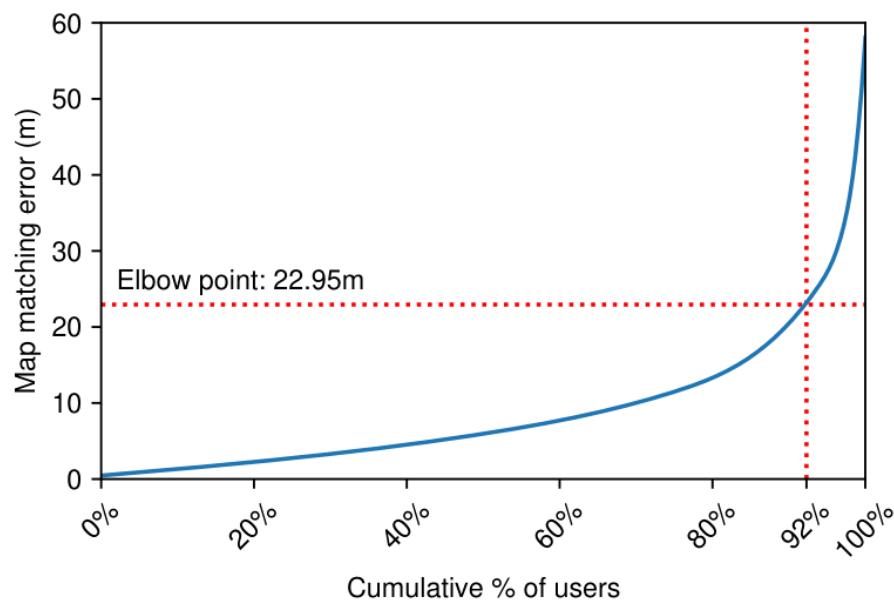
Het doel van de aanval is om de locatie te voorspellen waar een gebruiker effectief vertrekt of aankomt bij het uitvoeren van een sportactiviteit, ondanks het feit dat deze locatie wordt verborgen door het gebruik van een EPZ. Maar indien we dit zouden uittesten op activiteiten die al cloaking ondergingen door het platform zouden we de bekomen resultaten niet kunnen valideren. Daarom zullen we de aanval uittesten en evalueren op publieke activiteiten die geen EPZ bevatten, en deze manueel voorzien van een EPZ. Zo kunnen we de bekomen resultaten vergelijken met een referentie, namelijk de *grondwaarheid* of de Ground Truth (GT).

#### **5.1.1 De grondwaarheid**

De grondwaarheid van een gebruiker is de effectieve woonplaats, of de plaats waar deze persoon meestal vanuit vertrekt of aankomt. Dit is de locatie die we beschouwen als degene waarrond de EPZ wordt aangebracht, en die wij ultiem trachten te achterhalen. We bepalen deze locatie door alle activiteiten van een gebruiker te overlopen, en de begin- of eindpunten die binnen een straal van 50 meter uit elkaar liggen aan dezelfde cluster toe te voegen volgens het DBSCAN-algoritme. Hassan et al. stelden dat 50 meter vergelijkbaar is met een breedte van een gemiddeld perceel,

en dat dit dus een goede benadering vormt voor de maximale afwijking [38, 37]. Indien een cluster bestaat uit 15 punten of meer, dan berekenen we hiervan het gemiddelde en wordt dit gemiddelde gemapt op de roadgraph. Dit gesnakte punt stelt dan een grondwaarde voor van deze gebruiker. Let wel, het kan dat één gebruiker meerdere grondwaarden bevat.

Een tweede kanttekening die we hierbij moeten maken is dat de effectieve begin- en eindlocaties niet altijd perfect op het wegennetwerk zullen liggen. Een gebruiker kan bijvoorbeeld starten op een parking of een opritlaan, wat niet vertaald zit in het netwerk. Wij mappen deze dan achteraf op het straatnetwerk, maar gedurende de upload berekent het platform in kwestie wel de totale afgelegde afstand tot het effectieve startpunt. Dit kan dus voor een afwijking bij de predicties zorgen. Dhondt et al. onderzochten deze afwijking door een CDF te plotten die alle afwijkingen visualiseert, en op zoek te gaan naar het elleboogpunt [6]. Het elleboogpunt is een visueel punt in de curve waar zich een knik voordoet [5]. Dit duidt in theorie de optimale afweging tussen lage afwijking en hoge precisie aan. Zo bekomen Dhondt et al. een drempelwaarde van 22.95 meter om van een succesvolle aanval te kunnen spreken.



**Figuur 5.1:** Dhondt et al. bepaalt grafisch de trend van de afwijkingen bij het snappen van locaties op het wegennetwerk [6]

### 5.1.2 Manueel aanbrengen van een EPZ

We werken zoals al eerder vermeld met publieke activiteiten die geen EPZ bevatten, om zo de evaluatieproces mogelijk te maken. Maar om de aanval te kunnen uitvoeren moet we dus

nog manueel een EPZ aanbrengen. Zo kunnen we een situatie creëren die de werkelijke situatie benadert, en kunnen we ons aanvalsmodel uitvoeren. Het is dus wel belangrijk dat het zelf geïmplementeerde EPZ-mechanisme op een realistische manier wordt aangebracht zodat deze de werkelijkheid weerspiegelt.

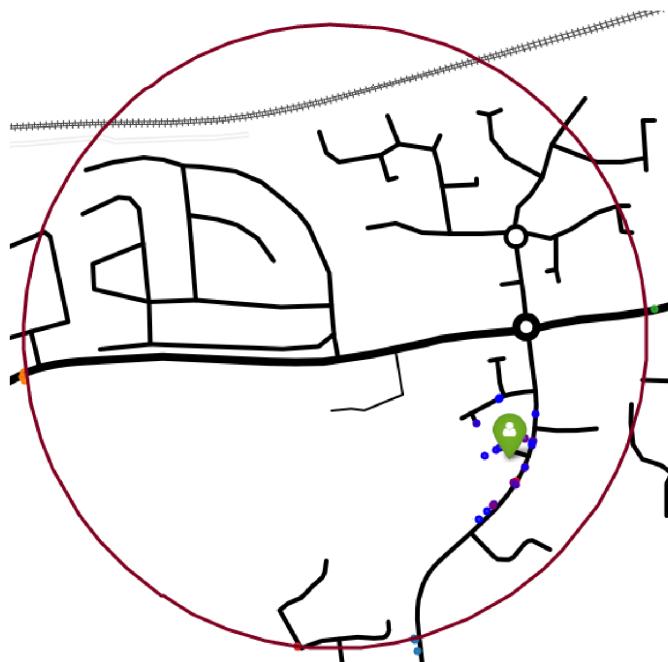
Sectie 2.3.1 bespreekt al uitvoerig het mechanisme van een EPZ, en hoe deze wordt bepaald. Om het even kort te recapituleren, een EPZ wordt bepaald door een centraal punt (de gevoelige locatie), wat een willekeurige translatie zal ondervinden<sup>1</sup>, en een gekozen straal. Vanaf het getransleerde punt wordt een cirkel opgezet met de desbetreffende straal. Strava als fitnessplatform heeft de grootste keuze uit mogelijke stralen, namelijk van 200 meter tot 1600 meter in sprongen van 200 meter. Het doel is om de effectiviteit van de aanval vast te leggen voor verschillende radiussen, dus zullen we per gebruiker en per aanvalsmodel een EPZ opzetten met alle verschillende radiussen. Zo kunnen we het effect van de radius zien, maar ook de types aanval onderling onafhankelijk van de straal vergelijken. We starten dus vanaf de GT. Deze locatie ondervindt dan een willekeurige translatie. De verschuiving van het punt kan gebeuren in alle richtingen, en kiezen we dus lukraak. De afstand van de translatie kan in principe ook willekeurig worden gekozen, maar moet wel binnen bepaalde grenzen liggen, namelijk tussen 0 en 70% van de straal van de EPZ. De cirkel kan dan worden opgesteld, met als middelpunt het getransleerde punt, en de bijhorende straal. Alle punten die zich binnen deze zone bevinden, zullen worden verwijderd uit de activiteit.

### 5.1.3 Bootstrapping

Bij het uittesten van de aanval wordt de aanval niet zomaar een enkele keer uitgevoerd voor alle activiteiten per gebruiker. Dit zou vertekende resultaten kunnen geven, door bijvoorbeeld één erg afwijkende activiteit. Per gebruiker wordt een betrouwbaarheidsinterval berekend via bootstrapping [6, 37]. De set met manueel verhulde activiteiten wordt beschouwd. Het bootstrapalgoritme kiest hieruit willekeurig één voor één activiteiten, en plaatst deze in een nieuwe groep totdat deze nieuwe groep activiteiten even groot is als de originele set van activiteiten. Let wel, het algoritme kan meerdere malen dezelfde activiteit kiezen, dus met andere woorden kan de nieuw gemaakte groep duplicaten bevatten en andere activiteiten helemaal niet bevatten. Dit gebeurt 1000 keer, en er worden dus 1000 verschillende sets gemaakt. Voor elke set zal een voorspelling worden uitgevoerd. Zo bekomen we een aantal voorspelde locaties, waarvan er een kans is dat enkele locaties meerdere malen voorspeld worden. Op Figuur 5.2 is te zien hoe de distributie eruit ziet op een kaart [37]. Op de figuur is de individuele voorspelling aangeduid met een blauwe tot paarse kleur, afhankelijk van hoe frequent ze voorspeld werden (hoe meer de kleure naar de paarse of rode kleur neigt, hoe frequenter de node voorspeld is). Ook zichtbaar zijn de start- en eindpunten, met een verschillend kleur per E.G. De locatie van de grondwaarde is aangeduid door de groene

<sup>1</sup>In de context van de beschrijving van Hassan et al. gebeurd er geen translatie, maar deze thesis gaat wel degelijk uit van een model waarbij spatial cloaking op toegepast is [38].

marker.



**Figuur 5.2:** Voorbeeld van een distributie van voorspellingen bepaald door het bootstrapalgoritme [37]

#### 5.1.4 Evaluatie metrieken

Om iets zinnigs te kunnen vertellen over de effectiviteit van de aanval definiëren we enkele metrieken die we kunnen gebruiken om de aanval te evalueren. We gebruiken hiervoor dezelfde metrieken die de studies van Dhondt et al. en Verdonck beschrijven, om zo onze resultaten er eenduidig mee te kunnen vergelijken. In totaal gebruiken we acht verschillende evaluatiemetrieken.

De eerste evaluatie metriek is de *Success Rate* [6]. Dit is het percentage van de uitgevoerde aanvallen waar de gevoelige locatie succesvol is achterhaald. Rekening houdend met de overshoots die komen met het snappen van locaties op het wegennetwerk, is een correcte locatie een locatie die zich binnen een straal van 22.95 meter van de GT bevindt. Hoe hoger het percentage, hoe succesvoller de aanval.

De *Correctness* van een aanval is de som van de Euclidische afstanden tussen de GT en de voorspelde locatie gedeeld door het aantal keer deze locatie werd voorspeld [6, 37]. Dit geeft een indicatie van de gemiddelde afwijking in afstand van de voorspelde locaties ten opzichte van de GT. Hoe lager deze waarde, hoe preciezer de aanval. Let wel, een success rate kan hoog zijn, met een correctheid die ook hoog is. Dit duidt op een aanval die veel overshoots heeft, maar waar de correcte locatie zich wel binnen de straal van 22.95 meter bevindt. De probabiliteitsdistributie wordt

gegeven door  $\widehat{\Pr}(v | a)$ , waarbij  $v$  de beschermde locaties voor activiteit  $a$  zijn in Vergelijking 5.1.

$$\sum_{v \in V} \widehat{\Pr}(v | a) \text{dist}(v, v_{GT}) \quad (5.1)$$

De *Accuracy* definiëren we als de breedte van het betrouwbaarheidsinterval [6, 37]. Met deze breedte doelen we op het aantal unieke voorspellingen, het aantal nodes dat precies eenmalig worden voorspeld. Hoe meer unieke nodes, hoe hoger de accuracy en ook hoe minder ‘zeker’ onze voorspelling is.

De *Reduction of the k-anonymity set* kwantificeert de afname in de set van alle mogelijke eindlocaties voor en na de effectieve predicties van een aanval [6, 37]. De mogelijke eindlocaties voor de aanval zijn simpelweg alle nodes in de graafvoorstelling, eventueel begrensd door de EPZ. Degene na de aanval zijn alle nodes die effectief voorspelt worden. De reductie is dus een percentage die het verschil tussen de twee sets aangeeft. Hoe hoger de reductie, hoe meer nodes verdwijnen uit de set van mogelijke eindlocaties na de aanval. Dit zegt dus iets over de hoeveelheid kandidaten het algoritme voorspelt, ten opzichte van hoeveel kandidaten er mogelijk zijn.

$$\frac{k - |V_{\text{pred, ext}}|}{k} \quad (5.2)$$

De *Uncertainty Region* ( $m^2$ ) is de som van oppervlaktes van de unie<sup>2</sup> van de onzekerheidsregio's rond de voorspelde nodes [6, 37]. De chaining distance, die in ons geval drie meter aanneemt, veroorzaakt deze onzekerheidsregio's. Aangezien pas om de drie meter een node bestaat, zal voor elke voorspelde node een kans bestaan dat deze eigenlijk een punt representeert die ergens in de zone van drie meter rond deze node ligt. Hoe groter deze waarde, hoe groter de onzekerheid van de aanval, aangezien dit wijst op weinig overlap. Een grotere waarde wijst dus ook op predicties die verder uit elkaar liggen.  $C_{v_p}$  stelt de cirkel rond een node voor, met straal  $d_{\text{chain}}$ .

$$\text{Area} \left( \bigcup_{v_p \in V_{\text{pred}}} C_{v_p}, d_{\text{chain}} \right) \quad (5.3)$$

De *Certainty* definieert de concentratie van de probabiliteitsdistributie [6]. Hoe hoger deze waarde, hoe verder van elkaar de nodes in de probabiliteitsdistributie liggen.

$$-\sum_{v \in V} \widehat{\Pr}(v | a) \log(\widehat{\Pr}(v | a)) \quad (5.4)$$

De *Spatial Certainty* is de certainty, maar in plaats van de probabiliteit van elke node te beschouwen, gebruiken we de neighborhood probabiliteit om zo de densiteit in de buurt van elke node te

---

<sup>2</sup>De unie van de oppervlakte van twee cirkels is de som van de twee oppervlakten, min één maal het overlappende deel.

beschouwen [6].

$$-\sum_{v \in V} \widehat{\Pr}(v | a) \log \left( \widehat{\Pr}_n(v | a) \right) \quad (5.5)$$

De laatste evaluatiefactor is de *Degree of Anonymity* [6, 37]. Dit is de genormaliseerde entropie van de verwachte distributie. Deze wordt genormaliseerd op basis van de maximale mogelijke entropie, en wordt bepaald op basis van percentage  $p_v$ . Dit is een percentage die aangeeft bij hoeveel percent van de voorspellingen node  $p$  voorspeld wordt. Dit zal voor vele nodes gelijk zijn aan 0. De maximale entropie komt voor indien elke node exact evenveel voorspelt wordt, en is dus gelijk aan  $\frac{1}{\#nodes}$ .

$$\frac{-\sum_{v \in V} \widehat{\Pr}(v | a) \log_2(\widehat{\Pr}(v | a))}{H_0(V)} \quad (5.6)$$

## 5.2 Resultaten

Nu we alle evaluatiemetrieken besproken hebben, kunnen we overgaan naar de evaluatie van de geteste scenarios. Elk getest scenario zal afzonderlijk worden besproken alsook onderling vergeleken worden. De resultaten van de aanvallen worden weergegeven in tabellen, die telkens voor elke metriek een score weergeven. Ook draaien we voor (zo goed als) alle beschreven scenario's een aanval voor enkele radiussen.

Daarnaast worden alle resultaten ook grafisch weergegeven op Figuur 5.3. Dit geeft een mooi globaal overzicht van alle modellen ten opzichte van elkaar, en maakt de onderlinge verschillen zichtbaar. Bij de bespreking van de resultaten zullen we dan ook zowel refereren naar de tabellen als naar de grafieken.

Over het algemeen is een gelijklopende trend merkbaar over de modellen heen bij het veranderen van de EPZs-radius. Bij een toenemende radius zakt de success rate doordat een grotere radius meer nodes met zich meebrengt. Meer nodes zorgt voor meer mogelijke verwarring in de LAD regressie [37]. Dit brengt ook een grotere degree of anonymity en uncertainty region met zich mee. Het aantal voorspellingen neemt niet evenredig toe met het aantal nodes in de graaf naarmate de omvang toeneemt, wat resulteert in een verhoogde reduction. Als laatste valt ook op dat de correctness ook stijgt bij een grotere radius. Dit komt door een grotere kans op schending van één van de gestelde assumenties uit Sectie 3.1.1. Deze sectie stelt onder andere dat een gebruiker het kortste pad moet volgen binnenin de EPZ. Echter hoe groter de EPZ van omvang is, hoe groter de kans dat hieraan niet voldaan wordt, wat zal resulteren in een hogere correctness.

### 5.2.1 Model volgens Dhondt et al.

Het eerste model dat we testen is het model van Dhondt et al. [6], Tabel 5.1 toont de betreffende resultaten. We gebruiken deze resultaten als referentie voor de rest van de resultaten. Het model van Dhondt et al. heeft geen restricties betreffende de beschikbare data, het kan dus alle data gebruiken. Het is dan ook logisch dat dit resulteert in goede scores. We zien hier dan ook een hoge succes rate, die relatief weinig afneemt bij hogere radiussen. De rest van de statistieken wijzen ook op een goede aanval, wat in lijn met de verwachtingen ligt. Het doel van de andere aanvallen was dan ook deze waarden te benaderen, maar met alternatieve data. Het aanvalsmodel is geïmplementeerd en uitgevoerd op ons eigen systeem op de beschikbare fractie van de dataset, wat de lichte verschillen ten opzichte van de resultaten beschreven in de desbetreffende paper verklaart.

Radius (m)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	89.86	28.61	17	86.06	352.08	2.06	0.58	30.69
400	79.1	56.82	21	93.21	469.76	2.26	1.01	28.31
600	70.37	79.94	24	96.52	502.42	2.32	1.12	27.15
800	73.33	113.68	27	97.05	670.53	2.47	1.31	26.86
1000	68.64	166.74	33	97.84	777.57	2.62	1.54	27.25
1200	58.33	180.97	27	98.15	684.71	2.55	1.57	26.25
1400	57.14	235.76	33	98.61	782.30	2.54	1.82	25.31

**Tabel 5.1** Aanval volgens het model van Dhondt et al. [6]

### 5.2.2 Gegeven outer distance

Het eerste scenario dat we testen is het model waarbij de outer distance rechtstreeks af te lezen valt uit de data. We bespraken dit geval al kortstondig in Sectie 3.4.3. Dit model komt voor wanneer de cumulatieve afstanden ter beschikking zijn. Het voordeel dat dit model heeft ten opzichte van de volgende modellen is dat er geen gps-data nodig is.

Hierbij zien we gelijkaardige trend als bij het model van Dhondt et al., wel met bij zo goed als alle metrieken een kleine achteruitgang. Er is slechts één tussenstap nodig ten opzichte van het model van Dhondt et al., namelijk het omrekenen van snelheid en de tijd tot de totale afstand. Dit verklaart dan ook meteen de kleine afnames en toenames in de resultaten. Deze omrekening zal een kleine afwijking met zich meebrengen, waarschijnlijk door afrondingen en mogelijke additionele berekeningen van Strava bij het berekenen van de snelheid.

Over het algemeen zijn de verschillen met het model van Dhondt et al. klein. De success rate zakt met ongeveer 5 à 10%, wat een logisch gevolg is van de extra stap die nodig is. De andere statistieken zoals de accuracy, reduction en uncertainty region liggen erg dicht bij de voorgaande resultaten. Al deze factoren wijzen op een goede aanval, die slechts een kleine afwijking vertoont ten opzichte van het optimale scenario, hoogstwaarschijnlijk veroorzaakt door kleine omrekenings-

fouten.

Radius (m)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	81.43	35.96	15	86.01	322.32	1.91	0.68	28.33
400	79.71	51.38	21	93.78	445.30	2.26	0.92	27.80
600	70.77	96.94	23	95.78	542.48	2.33	1.18	27.34
800	65.83	113.18	30	97.28	703.00	2.48	1.41	27.38
1000	62.39	191.47	31	97.60	698.69	2.62	1.62	27.31
1200	57.98	212.06	36	97.86	850.01	2.62	1.76	27.13
1400	49.15	270.35	29	98.54	648.70	2.51	1.72	24.90

**Tabel 5.2** Aanval op basis van gegeven *outer distance*, en snelheid

### 5.2.3 Ruwe gps-data

De volgende aanval is deze zonder gegeven cumulatieve afstand, maar ook zonder smoothing. Dit zorgt ervoor dat de aanvaller de ruwe gps-data rechtstreeks gebruikt voor het berekenen van de outer distance. In dit geval zit zowel de afwijking die afkomstig is van de snelheidsomrekening, die besproken werd in het vorige model (waarbij de outer distance gegeven is) alsook de afwijkingen die afkomstig zijn van de gps-data zelf vervat in de resultaten.

Deze afwijkingen wegen zoals verwacht relatief sterk door. Zeker bij grotere radiussen heeft dit een grote impact op de resultaten. Vanaf een radius van 1000 meter is de success rate zelfs 0%. Maar ook bij de rest van de metrieken zien we dat de score een sterke achteruitgang vertoond bij een hogere EPZ-radius. Dit is te wijten aan de grote afwijkingen die de gps-data in zijn geheel met zich mee brengt, zeker bij grotere radiussen weegt dit sterk door. Hoe groter de af te leggen afstand, in dit geval de inner distance, hoe groter de fout.

Een bijkomende hypothese is dat in het eerste geval, bij een radius van 200 meter, de filtering op een manier gebeurd waardoor de afwijkingen van de gps-data erg goed opgevangen worden. Wat leidt tot hoge success rates. Bij grotere EPZs is de filtering echter wat moeilijker uit te voeren en kunnen afwijkende activiteiten moeilijker gefilterd worden, wat ervoor kan zorgen dat grote afwijkingen toch nog de resultaten beïnvloeden. Als voorbeeld nemen we de filtering die de maximaal af te leggen afstand binnenin de EPZ in rekening brengt. Bij een grotere radius, is deze hoger dan bij lagere radiussen. Dit wil zeggen dat veel meer marge is waarop niet gefilterd zal worden.

Bijkomend zien we bij de slechte success rate ook een erg hoge correctness. Dit is te wijten aan de inaccuraatheid van de voorspellingen. De reduction is dan samen met de uncertainty region relatief gelijklopend met de waarden uit vorige modellen. Dit wijst op voorspellingen die goed bij elkaar liggen, maar de lage success rate en hoge correctness duiden erop dat deze op de verkeerde plek liggen. Als laatste merken we ook een zekere onvoorspelbaarheid op in de resultaten. Bijvoorbeeld op Figuur 5.3 zien we de accuracy serieuze spronzen maken. Dit is te wijten aan de onvoorspelbaarheid van de grote van de mogelijke fouten, en het effect ervan. Bij bepaalde activiteiten of gebruikers kunnen deze fouten een grotere waarde aannemen, en bij

andere zal deze dan weer minder zijn. En afhankelijk van de orde kan dit ook meespelen in de filtering, bij bepaalde radiussen kan het zijn dat deze afwijkingen zich zodanig manifesteren dat ze de filtering niet overleven, maar bij andere radiussen dan weer wel.

Radius (m)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	72.06	59.92	21	81.89	473.05	2.22	1.01	33.43
400	2.08	351.85	17	90.71	446.35	2.15	1.67	27.80
600	4.55	473.15	27	92.46	734.62	2.57	2.17	30.67
800	2.13	651.38	42	95.06	1161.95	2.87	2.32	30.84
1000	0.00	737.93	37	96.84	994.80	2.76	2.22	29.69
1200	0.00	955.79	22	97.63	592.09	2.54	2.28	25.16
1400	0.00	986.46	25	98.08	697.50	2.44	2.21	23.70

Tabel 5.3 Aanval op basis van ruwe gps-locaties (geen smoothing) en snelheid

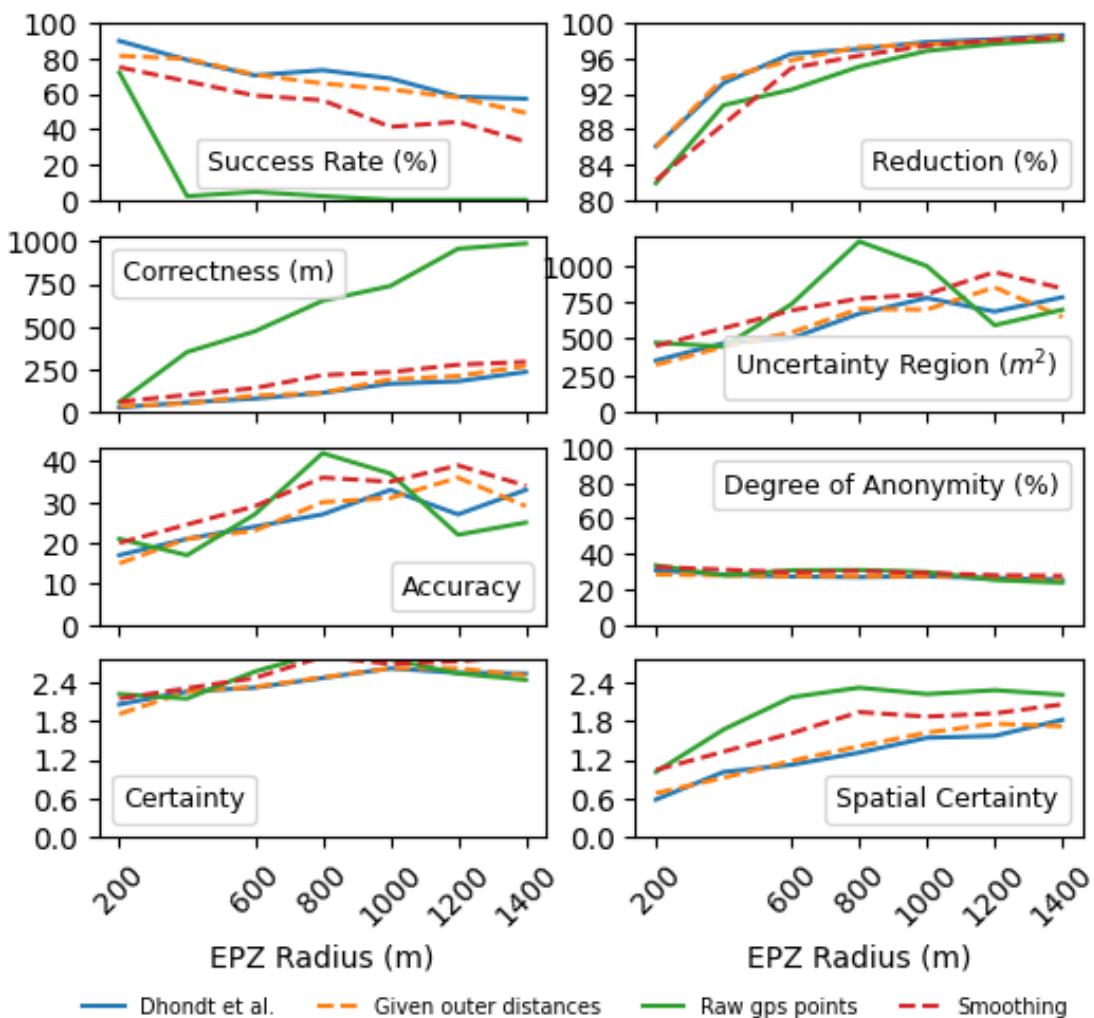
#### 5.2.4 Smoothing

Het laatste model is gelijkaardig aan het voorgaande, met als verschil dat nu wel smoothing wordt toegepast op de trajecten in een poging om de afwijkingen van de gps-data te verminderen. In Sectie 3.4.3 beschreven we al het smoothing mechanisme. Zeker de extremen zullen worden afgevlakt, wat de fouten tot een lagere orde zou moeten reduceren. Dit toont zich ook in de resultaten. Het gebruikte smoothing-algoritme heeft één parameter die wij kunnen aanpassen, namelijk de smoothing window. Deze parameter bepaalt hoeveel punten er per window worden gecombineerd. Het optimale smoothing window wordt hier bepaald door het uitvoeren van een aantal experimenten, en hiervan de resultaten met elkaar vergelijken. Deze resultaten zijn terug te vinden in Bijlage A op Tabel A.1. Deze zijn ietwat wisselvallig, en er is geen duidelijke trend in terug te vinden. Let wel dit smoothing window werd empirisch bepaald en optimaal gekozen voor deze dataset. Voor een andere dataset kan het optimale window een verschillende waarde aannemen dan voor degene die we in deze thesis gebruiken. Maar wanneer we zoeken naar het best scorende smoothing window, komen we uit op een smoothing window van 100. Dit hoge smoothing window wijst wel nog maar eens op de vele fouten die de gps-data bevat.

De success rate toont een relatief kleine verbetering ten opzichte van het model met de ruwe gps-data bij kleine EPZs, maar bij grotere EPZs tonen de metrieken veel minder verval. Ook andere metrieken dan success rate vertonen een gelijkaardig patroon, waarbij de afzwakking veel minder sterk is dan bij het gebruik van ruwe gps-punten. Dit toont dat het gebruik van smoothing zeker een significante toegevoegde waarde heeft. We zien echter wel dat over het algemeen de accuracy iets hoger ligt, wat wel nog duidt op onzekere voorspellingen. De fouten in de gps-locaties spelen dus wel nog zeker een rol in de voorspelling.

Radius (m)	Smoothing Window (n)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	100	75.0	61.37	20	82.22	450.15	2.15	1.04	32.57
600	100	58.97	141.04	29	94.89	692.52	2.47	1.61	29.51
800	100	56.34	217.13	36	96.30	773.61	2.80	1.94	30.30
1000	100	41.27	234.27	35	97.43	802.93	2.69	1.87	29.13
1200	100	44.12	278.00	39	98.06	953.93	2.73	1.92	27.86
1400	100	32.81	294.24	34	98.28	841.94	2.82	2.06	27.51

**Tabel 5.4** Aanval op basis van gesmoothe gps-data en snelheid, met een empirisch bepaald optimaal smoothing window  $n = 100$



**Figuur 5.3:** Vergelijking van de verschillende aanvallen

# **Hoofdstuk 6**

## **Conclusies en toekomstig werk**

### **6.1 Conclusies**

Deze thesis toont aan dat de inferentie-aanvallen mogelijk zijn op basis van snelheden en gps-data, weliswaar met een lagere success rate en grotere onzekerheid. Indien de cumulatieve afstand ter beschikking is, kan via een eenvoudige omrekening ( $\text{inner distance} = \text{total time} \times \text{average pace} - \text{outer distance}$ ) de aanval nog steeds worden uitgevoerd met een success rate van 81.43%, 79.71%, 70.77%, 65.83%, 62.39%, 57.98%, 49.15% voor de desbetreffende radii: 200m, 400m, 600m, 800m, 1000m, 1200m en 1400m, wat absoluut aanvaardbaar is. Indien de cumulatieve afstandsdata op zijn beurt niet voor handen is, kan via gps-locaties een additionele omrekening gebeuren om de aanval alsnog te doen slagen. Door het gebruik van smoothing algoritmes kan de aanval ook in precisie winnen (ten opzichte van het gebruik van ruwe gps-data), en komen we tot een success rate van 75.0%, 58.97%, 56.34%, 41.27%, 44.12%, 32.81% voor de desbetreffende radii en een smoothing window van 100. Dit is op zijn beurt een acceptabele success rate, maar is wel een stuk lager dan de success rate bekomen door Dhondt et al. [6].

Bepaalde afstandsdata, meer bepaald de cumulatieve afstand en de totale afstand zijn dus niet van cruciaal belang voor het succesvol uitvoeren van dit aanvalsmodel. Let wel, de dataset die ter beschikking werd gesteld voor dit onderzoek is aan de kleine kant. De getrokken conclusies in deze thesis mogen dus niet zomaar worden geëxtrapoleerd naar de volledige groep van Stravagebruikers.

Dhondt et al. beschreven enkele maatregelen om de privacy van gebruikers te verzekeren, zoals het afronden van afstandsdata of het toevoegen van ruis aan deze data om deze zo onbruikbaar te maken [6]. Dit is dus niet meer van toepassing, bijna alle ‘*Distance-Focused Countermeasures*’ beschreven door Dhondt et al. kunnen worden omzeild door ons aanvalsmodel. De aanvaller

kan de afstandsdata in theorie herberekenen, en ultiem nog steeds de aanval tot een goed einde brengen. Alle distance-based maatregelen die Dhondt et al. beschreven zijn dus in principe niet meer bruikbaar. We zouden deze countermeasures wel kunnen uitbreiden door hierbij de snelheid en/of tijd in rekening te brengen. Bijvoorbeeld bij generalization, wat neerkomt op afronding van een afstand die zichtbaar is naar de buitenwereld, kan de snelheid op dezelfde manier een manipulatie ondergaan. Voor alle countermeasures in deze categorie, behalve *Shifting Distances* geldt dit principe, namelijk dat deze teniet worden gedaan ten opzichte van ons aanvalsmodel, maar indien deze uitgebreid worden naar de tijd en/of snelheid zouden deze alsnog nuttig kunnen blijken. Alhoewel niet expliciet getest, is de hypothese dat alle countermeasures, indien uitgebreid naar de snelheid en/of tijd, een gelijkaardig effect zullen hebben. De besproken countermeasures zijn:

- *Generalization* houdt in dat de afstand wordt afgerond tot een bepaalde precisie. Dhondt et al. stelt een precisie van 500 meter voor [6].
- *Noisy Distances* is een techniek waarbij een willekeurige waarde wordt toegevoegd of afgetrokken van de totale afstand [6]. Let wel dat, indien dit ruis volledig willekeurig zou worden toegevoegd, deze maatregel indien er een voldoende aantal activiteiten ter beschikking niet bruikbaar is. Het ruis zou op deze manier zichzelf opheffen. Wij stellen daarom voor dat deze ruis niet volledig willekeurig mag zijn. Indien we deze echter combineren met de countermeasure *Shifting Distances*, kan deze volgens ons wel nuttig blijken.
- *Shifting Distances* verschuift de zichtbare begin- of eindpunten van een activiteit met een willekeurige afstand, in een willekeurige richting, zodat het vertrekpunt onzeker wordt [6]. Dit zorgt voor problemen bij het reconstrueren van de route.
- *Truncation* stelt dat het verborgen traject niet mee in rekening wordt gebracht in de totale afstand [6]. Dit verwijdert eigenlijk het verborgen traject volledig uit de activiteit. Dit is dus een zeer drastische maatregel.

De countermeasure *Shifting Distances* is de enige die niet kan worden omzeild door onze implementatie, doordat voor ons model ook de zichtbare begin- en eindpunten worden genomen, en een vitaal onderdeel van de aanval zijn, net zoals bij het model door Dhondt et al..

Het is en blijft natuurlijk wel een afweging tussen privacy en gebruiksvriendelijkheid. Indien waarden van de gebruiker worden afgerond of gemanipuleerd, verliest deze informatie in waarde. In fitnessplatformen is er meestal een functie ingebouwd met betrekking tot leaderboards en segmenten, of records op bepaalde afstanden van een gebruiker. Indien we knoeien met de precisie van deze waarden, verliezen deze functies in nut. Indien bijvoorbeeld een segment aansluit bij het begin van een EPZ, dan zijn door de invloeden van de tegenmaatregelen deze waarden niet meer bruikbaar. Wij stellen daarom voor om deze segmenten, in het geval dat deze countermeasures van kracht zouden zijn, te laten meetellen enkel en alleen indien de segmenten geen invloed ondervinden van een EPZ. Bijkomend kunnen statistieken over snelste tijden op een bepaalde afstand

of langste activiteiten niet gelinkt worden aan een activiteit (wat nu wel het geval is, momenteel kan men vlot terug vinden welke activiteit zorgde voor welke tijd, snelheid en afstand, indien dit op de een of andere manier een record voorstelt), maar enkel en alleen aan een gebruiker zelf. Zo kan een gebruiker bijvoorbeeld de statistieken van zijn snelste tijd op de marathon in detail weergeven, zonder dat hier routes en locaties worden vrijgegeven. Daarnaast zouden we de gebruiker de keuze kunnen geven om deze statistieken volledig privaat te houden, want ondanks de loskoppeling van locaties zou deze toch in vele gevallen via vergelijkingen van statistieken kunnen teruggevonden worden. Daarnaast zijn we ook van mening dat de gebruiker hier dan zelf bijkomstig nog een keuze moet hebben in de betrekking tot leaderboards. De gebruiker moet zelf kunnen kiezen of hij/zij deze extra privacy wenst door zichzelf hiervan te onttrekken, of dat deze functionaliteit voor hem/haar belangrijker is.

In het geval van de andere categorie van countermeasures, namelijk de ‘EPZ-Focused Countermeasures’, is de verwachting dat deze wel nog steeds gelden als nuttige countermeasures. Deze zijn:

- *Increasing EPZ radii* wat, zoals de naam doet vermoeden inhoudt dat de EPZ wordt ver-groot [6]. We merken op dat de aanval aanzienlijk minder presteert bij grote EPZ radiussen.
- *Complex EPZ shapes* houdt in dat de EPZ niet langer een cirkel is, maar een complexere vorm aanneemt [6]. Dit kan bijvoorbeeld een veelhoek zijn.

Als laatste merken we wel op dat onze aanval enkel mogelijk is bij activiteiten waarvan slechts het begin- of eindpunt wordt gecloaked, nooit beide. Indien beide punten worden gecloaked, is het onmogelijk om de route te reconstrueren volgens ons model. Dit is een beperking van ons model, maar resulteert wel in een mogelijke countermeasure. Indien de snelheid en tijd van een activiteit onaangestast blijven, maar een *Distance-Focused Countermeasure* wordt toegepast zoals beschreven door Dhondt et al., zou het volstaan om beide punten te cloaken om de aanval die we beschrijven te voorkomen, zoals de dag van vandaag gebeurt.

## 6.2 Toekomstig werk

Naast de volledige uiteenzetting van dit onderzoek, zijn er nog enkele zaken die misschien niet in detail genoeg werden onderzocht, en die ook interessante resultaten zouden kunnen naar voor brengen. Ook zijn er bepaalde onderwerpen die hierop verder bouwen en ook het onderzoeken waard kunnen zijn. Daaronder valt het implementeren van het beschreven map matching, wat eventueel zou kunnen leiden tot een hogere success rate bij het uitvoeren van de aanval. Zeker de combinatie met het smoothing mechanisme kan een interessant onderwerp zijn. Hierbij zou ook

nog kunnen geëxperimenteerd met een dynamisch smoothing window [22]. Dit is een smoothing window dat zich aanpast aan de eigenschappen van het signaal, zoals de ruis of veranderingsnelheid van de grafiek. Bijvoorbeeld dat de breedte van het venster verhoogt bij grote hoeveelheden ruis is een mogelijke implementatie.

Ook zou er een additionele analyse kunnen plaatsvinden om te zien onder welke externe omstandigheden de aanval het meest succesvol is. Een belangrijk eigenschap die hierbij zeker ook aan bod moet komen is de hoeveelheid activiteiten die zorgen voor een succesvolle aanval. Daarnaast zou in een bijkomstig onderzoek dat een wat bredere invloed heeft op het onderzoeks domein kunnen worden onderzocht in hoeverre omgevingsfactoren zoals bebouwingsdichtheid een invloed hebben op de success rate en dergelijke en of de success rate niet dynamisch kan worden gesteld op basis van deze dichtheid. Een straat die slechts één huis bevat is immers al te onderscheiden met een veel lagere precisie dan een met meer inwoners in een stad. De 22.5 meter die nu empirisch werd bepaald zouden we kunnen aanpassen op basis van de bebouwingsdichtheid.

Deze thesis bespreekt uitvoerig het model van Dhondt et al., maar het model van Verdonck komt relatief beperkt aan bod. Verdonck werkt op basis van een hoogteverschillen als basis voor zijn aanval. In de toekomst zouden we echter een onderzoek kunnen voeren in hoeverre deze twee modellen kunnen worden gecombineerd. Wij werken net als Dhondt et al. op basis van afstandsverschillen, maar indien we deze verschillen die we nu beschouwen, twee-dimensionaal beschouwen, waarvan hoogte en afstand elk een dimensie voorstellen zou de precisie mogelijks verhoogd kunnen worden, indien de verschillen we nu via euclidische afstanden zouden berekenen.

Als laatste is er een idee om een alternatieve implementatie van de LAD-regressie te gebruiken. In de huidige implementatie wordt voor elke activiteit de afwijking berekend ten opzichte van alle nodes in de graaf, en wordt de node gekozen die over de volledige lijn de laagste afwijking bevat, wat beschreven staat in Sectie 3.5.3.

Het alternatief dat wij voorstellen is om voor elke activiteit één of meerdere nodes te voorspellen die de laagste afwijkingen hebben ten opzichte van die activiteit. En uiteindelijk de node te verkiezen die in totaal de laagste totale som van alle onderlinge afstanden heeft. Met andere woorden, degene die het meest gecentreerd ligt. De LAD-methode die beschreven werd in deze thesis gaat op zoek naar een zo laag mogelijke afwijking voor elke activiteit, uitgaande van het ideale scenario waarbij elke activiteit stopt of begint op dezelfde locatie. In de hier alternatief beschreven LAD-methode vertrekken we eerder vanuit het feit dat activiteiten alternatieve locaties aanwijzen volgens hun afstand, door het feit dat ze effectief niet op dezelfde locatie eindigen. Gebruikers stoppen eens wat verder, eens wat vroeger, enzovoort. De alternatieve implementatie van de LAD-methode zou dit dus voor een stuk kunnen opvangen.

# Bibliografie

- [1] Bowden, A. (2018). Cyclist who had five bikes stolen says thieves are looking for quick times on strava to try and find high-end bikes – warns other users to check their privacy settings — road.cc. <https://road.cc/content/news/248798-cyclist-who-had-five-bikes-stolen-says-thieves-are-looking-quick-times-strava>. (Geraadpleegd op 02/20/2023).
- [2] Carr, C. T. and Hayes, R. A. (2015). Social media: Defining, developing, and divining. *Atlantic Journal of Communication*, 23(1):46–65.
- [3] Clement, L. (2019). Cursus statistiek 2019-2020. <https://statomics.github.io/statistiekCursusNotas/index.html>. (Geraadpleegd op 05/14/2023).
- [4] Croft, J. (2015). Snapping gps tracks to roads. <https://www.jamesrcroft.com/2015/06/snapping-gps-tracks-to-roads/>. (Geraadpleegd op 04/07/2023).
- [5] Cui, M. (2020). Introduction to the k-means clustering algorithm based on the elbow method. <https://www.claudiuspress.com/article/592.html>. (Geraadpleegd op 05/17/2023).
- [6] Dhondt, K., Le Pochat, V., Voulimeneas, A., Joosen, W., and Volckaert, S. (2022). A run a day won't keep the hacker away: Inference attacks on endpoint privacy zones in fitness tracking social networks. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 801–814, New York, NY, USA. Association for Computing Machinery.
- [7] Driesen-Joanknecht, H. (2020). Tempo (min/km) vs snelheid (km/h) bij hardlopen – sport sneller massage, preventie, nijmegen. <https://sportsneller.nl/2020/11/25/tempo-min-km-vs-snelheid-km-h-bij-hardlopen/#:~:text=Waarom%20gebruiken%20hardlopers%20geen%20km,de%20snelheid%20uit%20te%20drukken>. (Geraadpleegd op 04/18/2023).
- [8] Early, J. (2020). Smoothing and interpolating noisy gps data. <https://jeffreyearly.com/smoothing-and-interpolating-noisy-gps-data/>. (Geraadpleegd op 04/07/2023).
- [9] Early, J. J. and Sykulski, A. M. (2020). Smoothing and interpolating noisy gps data with smoothing splines. *Journal of Atmospheric and Oceanic Technology*, 37(3):449 – 465.

- [10] EduPristine (2017). What is standard deviation and how is it important? <https://www.edupristine.com/blog/what-is-standard-deviation>. (Geraadpleegd op 05/12/2023).
- [11] Hashemifar, S. (2018). Kmeans vs. dbSCAN. in data science and machine learning... — by soroush hashemifar — medium. <https://soroushhashemifar.medium.com/kmeans-vs-dbscan-d9d5f9dbe8b>. (Geraadpleegd op 05/17/2023).
- [12] Hern, A. (2018). Fitness tracking app strava gives away location of secret us army bases — gps — the guardian. <https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases>. (Geraadpleegd op 02/20/2023).
- [13] Howard, P. and Parks, M. (2012). Social media and political change: Capacity, constraint, and consequence. *Journal of Communication*, 62.
- [14] Iqbal, M. (2021). Application of regression techniques with their advantages and disadvantages. *Elektron. Mag*, 4:11–17.
- [15] Javaid, A. (2013). Understanding dijkstra algorithm. *SSRN Electronic Journal*.
- [16] Ladetto, Q., Gabaglio, V., and Merminod, B. (2001). Combining gyroscopes, magnetic compass and gps for pedestrian navigation. *Proceedings of the International Symposium on Kinematic Systems in Geodesy, Geomatics, and Navigation*.
- [17] (LEDU), E. E. (2018). Understanding k-means clustering in machine learning — by education ecosystem (edu) — towards data science. <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>. (Geraadpleegd op 04/25/2023).
- [18] Mink, J., Yuile, A. R., Pal, U., Aviv, A. J., and Bates, A. (2022). Users can deduce sensitive locations protected by privacy zones on fitness tracking apps. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.
- [19] Neira, M. and Murcio, R. (2022). Graph representation learning for street networks.
- [20] of Dallas, F. R. B. (n.d.). Smoothing data with moving averages - dallas-fed.org. <https://www.dallasfed.org/research/basics/moving#:~:text=A%20moving%20average%20smoothes%20a,the%20variable's%20timeliness%20is%20lost>. (Geraadpleegd op 04/13/2023).
- [21] Plas;, J. V. (2016). In depth: k-means clustering — python data science handbook. <https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html>. (Geraadpleegd op 05/01/2023).

- [22] Schrader, F., Durner, W., Fank, J., Gebler, S., Pütz, T., Hannes, M., and Wollschläger, U. (2013). Estimating precipitation and actual evapotranspiration from precision lysimeter measurements. *Procedia Environmental Sciences*, 19:543–552.
- [23] Seiler, K. M. (2022). Haul road mapping from gps traces.
- [24] Sheppard, W. and Soule, C. (1922). *Practical Navigation*. World Technical Institute.
- [25] Strava, I. (2020). Strava milestones: 50 million athletes and 3 billion activity uploads. <https://blog.strava.com/press/strava-milestones-50-million-athletes-and-3-billion-activity-uploads/>. (Geraadpleegd op 05/14/2023).
- [26] Strava, I. (2021a). Strava-privacybeleid. <https://www.strava.com/legal/privacy>. (Geraadpleegd op 02/20/2023).
- [27] Strava, I. (2021b). Strava's year in sport 2021 charts trajectory of ongoing sports boom. <https://blog.strava.com/nl/press/yis2021/>. (Geraadpleegd op 02/26/2023).
- [28] Strava, I. (2022a). Moving time, speed, and pace calculations – strava support. <https://support.strava.com/hc/en-us/articles/115001188684-Moving-Time-Speed-and-Pace-Calculations>. (Geraadpleegd op 02/26/2023).
- [29] Strava, I. (2022b). Why is gps data sometimes inaccurate? – strava support. <https://support.strava.com/hc/en-us/articles/216917917-Why-is-GPS-data-sometimes-inaccurate->. (Geraadpleegd op 05/14/2023).
- [30] Strava, I. (2023a). Activity privacy controls – strava support. <https://support.strava.com/hc/en-us/articles/216919377-Activity-Privacy-Controls>. (Geraadpleegd op 02/27/2023).
- [31] Strava, I. (2023b). Bad gps data – strava support. <https://support.strava.com/hc/en-us/articles/216917707-Bad-GPS-Data>. (Geraadpleegd op 03/01/2023).
- [32] Strava, I. (2023c). How distance is calculated – strava support. <https://support.strava.com/hc/en-us/articles/216919487-How-Distance-is-Calculated>. (Geraadpleegd op 03/01/2023).
- [33] Strava, I. (2023d). Strava global heatmap. <https://www.strava.com/heatmap#7.00/-120.90000/38.36000/hot/all>. (Geraadpleegd op 05/17/2023).
- [34] Sumudrala, A. (2019). Density estimation: Mle, map, mom, kde, ecdf, q-q plot, gan — by ajit samudrala — analytics vidhya — medium. <https://medium.com/analytics-vidhya/density-estimation-mle-map-mom-kde-ecdf-q-q-plot-gan-5161f84d28d7>. (Geraadpleegd op 05/23/2023).
- [35] TechTarget (2011). What is heat map (heatmap)? — definition from techtarget. <https://www.techtarget.com/searchbusinessanalytics/definition/heat-map>. (Geraadpleegd op 05/17/2023).

- [36] Vanmeldert, D. (2022). Sportapp strava laat fietsdieven of stalkers nog altijd mee kijken — vrt nws: nieuws. <https://www.vrt.be/vrtnws/nl/2022/10/28/strava-kul/>. (Geraadpleegd op 02/20/2023).
- [37] Verdonck, T. (2022). Inferentie-aanvallen met hoogteprofielen tegen (endpoint) privacy zones in fitness tracking sociale netwerken. Master's thesis, KU Leuven. Faculteit Industriële Ingenieurswetenschappen, Leuven. Book Title: Inferentie-aanvallen met hoogteprofielen tegen (endpoint) privacy zones in fitness tracking sociale netwerken.
- [38] Wajih UI Hassan, Saad Hussain, A. B. (2018). Analysis of privacy protections in fitness tracking social networks -or- you can run, but can you hide?
- [39] Zheng, J. (2019). Distance application in data science - jingwen zheng. <https://jingwen-z.github.io/distance-application-in-data-science/>. (Geraadpleegd op 05/07/2023).

## **Bijlage A**

# **Resultaten van de uitgevoerde aanvallen met verschillende smoothing windows**

**Tabel A.1** Aanval op basis van gesmoothde gps-data en snelheid

Radius (m)	Smoothing Window (n)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	5	73.98	60.35	22	82.41	450.52	2.21	1.03	32.92
600	5	62.50	124.71	32	94.64	720.98	2.59	1.58	29.95
800	5	58.93	186.66	38	96.31	806.26	2.87	1.82	31.83
1000	5	40.20	243.65	34	97.43	812.48	2.66	1.85	28.42
1200	5	50.00	249.91	42	97.55	1007.75	2.92	1.91	29.35
1400	5	40.38	248.87	41	97.98	977.24	2.99	2.11	29.81
200	10	70.59	69.52	22	81.14	480.38	2.2	1.06	33.34
200	15	71.67	61.49	22	82.75	480.13	2.17	1.01	32.96
200	20	70.94	61.21	21	82.76	458.57	2.21	1.03	33.35
200	25	72.17	60.44	22	83.07	464.94	2.17	1.02	32.89
200	50	72.12	60.67	20	82.4	451.9	2.15	1.03	32.28
600	50	54.44	150.15	35	94.50	793.84	2.59	1.63	29.96
400	50	70.53	96.87	31	90.66	685.01	2.52	1.38	32.7
800	50	50.60	190.30	37	96.61	834.45	2.72	1.83	29.26
1000	50	52.38	224.32	36	97.08	906.66	2.71	1.88	27.93
1200	50	38.46	275.84	48	97.77	1127.09	2.96	1.94	30.14
1400	50	40.24	335.89	41	97.96	1037.62	2.83	2.11	27.80
200	100	75.0	61.37	20	82.22	450.15	2.15	1.04	32.57
600	100	58.97	141.04	29	94.89	692.52	2.47	1.61	29.51
800	100	56.34	217.13	36	96.30	773.61	2.80	1.94	30.30
1000	100	41.27	234.27	35	97.43	802.93	2.69	1.87	29.13
1200	100	44.12	278.00	39	98.06	953.93	2.73	1.92	27.86
1400	100	32.81	294.24	34	98.28	841.94	2.82	2.06	27.51
200	110	72.62	62.94	19	82.46	432.95	2.15	1.04	32.23
200	125	72.15	66.86	20	82.01	461.84	2.16	1.04	32.2
200	150	72.73	67.81	20	82.25	475.05	2.14	1.05	31.7
400	150	63.01	91.54	31	90.41	681.20	2.51	1.44	32.77
600	150	62.69	145.28	33	94.05	831.22	2.63	1.75	30.28
800	150	52.54	179.38	41	96.96	990.78	2.70	1.68	29.72
1000	150	40.68	255.28	30	97.57	776.27	2.54	1.82	27.57
1200	150	42.19	309.25	35	97.61	888.10	2.73	2.05	26.69
1400	150	37.50	328.00	38	98.13	904.30	2.75	2.08	27.24

## **Bijlage B**

## **Scientific Article**

# Time Is Running Out

## Assessing Temporal Privacy of Privacy Zones in Fitness Tracking Social Networks

Deleu, Wout

KU Leuven, Campus Rabot

Ghent, Belgium

**Abstract**—In a society where social media is so ubiquitous, the privacy concerns around them are more relevant than ever. During this article, the main focus will be on the privacy policies of fitness trackers. Fitness trackers are platforms which store and display data related to sport activities. These can be shared with other users. This data may include heart rate, GPS-locations, etc. This type of data sharing can however cause unintentionally sharing of sensitive information, like home addresses.

Most fitness tracking networks are aware of this danger and implement a series of countermeasures to prevent this. One of these countermeasures is the use of Endpoint Privacy Zones (EPZs) which is a zone around a sensitive location, which hides the part of the trajectory which ends or begins in this zone. Previous research has shown that it is possible to retrieve the sensitive location using the available data from the activity. Dhondt et al. showed that based on the total distance travelled, the sensitive location can be retrieved using an ‘inference attack’ [1]. This study will investigate the possibilities of such inference attacks using other data than the distance. We want to recreate the results as good as possible using the speed and tempo of the activity, together with GPS-locations. This can result in an attack model with a success rate up to 75%. This is lower than the previous implementation of Dhondt et al., but this shows that the attack is still possible under circumstances where the distance is rendered unusable. This also includes some countermeasure described by Dhondt et al. But countermeasures like enlarging the EPZ or shifting endpoints still have effect.

**Keywords**—fitness-trackers, privacy, GPS-locations, endpoint privacy zone, inference attack

### I. INTRODUCTION

Social media has become virtually indispensable in today’s modern life and branches out into a lot of facets, including social networking, media sharing networks, etc, but also the branch of fitness trackers. This rise of these new media, however, also brings unintended but significant privacy concerns.

The focus of this thesis is on privacy within these fitness trackers, more specific platforms that use GPS locations, such as Strava<sup>1</sup>, Nike Run Club, etc. These are platforms where individuals can share sports activities such as running, cycling, hiking, ... with each other. The general concept is here that when you perform a sports activity, you make it available to your followers and friends. The sports activities will naturally release certain data to those other users, which could possibly have negative effects on the users privacy. For example leaking the home address of the user, which can lead to stalking and

burglary [2, 3]. There are even some reports about military bases being discovered using the Strava heatmap [4].

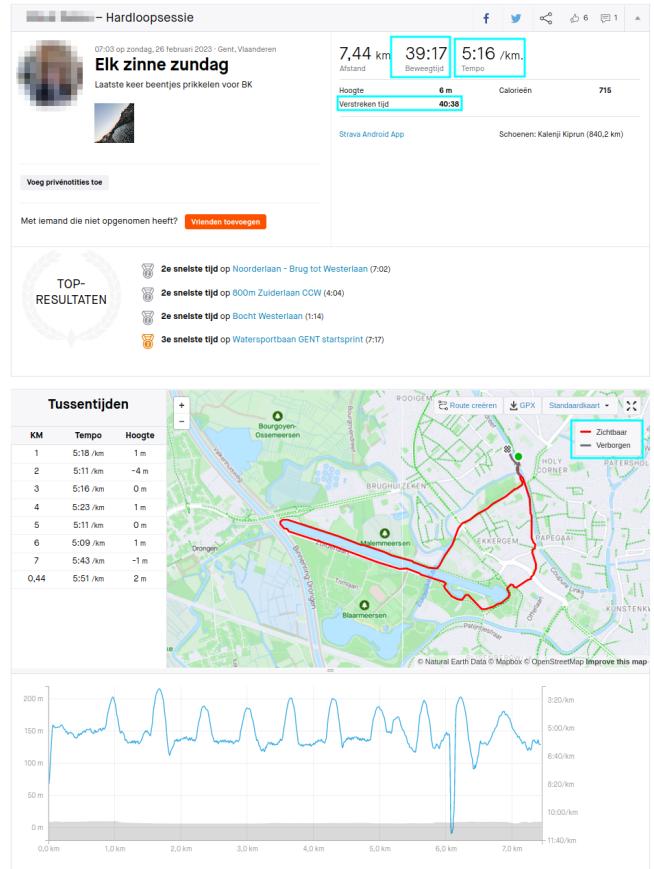


Fig. 1: Example of a Strava activity

Fitness trackers each implement ways to improve user privacy. Perhaps the most easy way to think of is perhaps the ability to hide activities from a selection of people (e.g., anyone who is not a follower). Thus, only the people whom the user explicitly allows to view activities. A more complex alternative is to use EPZs. Hereby a portion of the displayed route will be hidden for external viewers. Part of the route is cut off, so to speak. The real starting and ending points will lie inside the truncated part. New points will be generated, on the edge of the circle, which for the external observer will represent the beginning and end. The beginning and end part

<sup>1</sup><https://www.strava.com/>

of the route will thus become invisible to the other users. Due to the presence of all these attempts at privacy enhancements, it is noticeable that the developers of the platforms are very aware of the potential dangers. However, there is a trade-off to be made when implementing between the usability of the platform, and the privacy of the end user. The more data is released, the greater the chance of potentially sensitive info being passed along. On the other hand, when omitting information, the usability and presence of useful data of the platform is greatly diminished.

In this study, we consider whether there is a possibility to retrieve hidden locations of an activity, despite the use of an EPZ as privacy security mechanism. Some ways to bypass the EPZ using other metadata such as elevation data and distances have been described in the past [1, 5, 6]. This thesis goes into more detail on the use of velocity data. As a base for this attack, we use the inference attack, which was described by Dhondt et al. We investigate whether this attack is still possible when omitting certain data, and thus by the use of alternative data. The focus of this study is mainly on velocity-based data.

To achieve this objective, we must first examine the attack model according to Dhondt et al. Then we can describe different possible alternative attack scenarios, and for each work out the calculations necessary in order for this scenario to be possible. Before we can test and analyze the attack, we must consider the possible errors in the used dataset. We also perform an analysis on the difference between the calculated distances, and the values derived according to the calculations of Dhondt et al. So can we estimate the effectiveness of the attack a priori to some extent. Only then do we execute the attack, evaluate the attacks and form meaningful conclusions.

## II. BACKGROUND

### A. Fitness Trackers

The data used to test the effectiveness the attack and perform experiments comes from the popular fitness tracker Strava<sup>1</sup>. This is a social network where all types of athletes can share their activities. This includes running, walking, cycling, swimming, ... The collected data is filtered according to the perspective of a possible attacker. Not all data turns out to be useful. Only data that could reveal sensitive information regarding residence is retained. This will therefore mean that only activities that contain relevant GPS information will be considered, so only *runs, hikes, walks, and bike rides*.



Fig. 2: Strava logo [7]

### B. GPS faults

Some visible data of an activity, which can also be found on Figure 1, are distance, duration, average speed, etc. The average speed forms the core to the attack which we will describe in this article. This will be calculated as the total distance

divided by the moving time of the athlete. Fitness trackers receive raw data from the devices. This data must therefore be processed before it is useful to the users. Especially GPS-data, which can be exposed to a lot of faults and noise. There are three types of GPS faults, namely *GPS drift*, *GPS signal loss* and *GPS bounce*. GPS-drift is a phenomenon where a user's GPS location deviates from the effective location. This may be caused by densely built environments, and natural factors such as tall trees. GPS bouncing is an anomaly caused mainly by tall buildings. In this situation, the GPS signal will bounce in between buildings on its way to the device from the satellite. The extra delay that the bounces bring along causes the device to mistakenly think it has traveled some additional distance. The outcome of the trajectory is then unpredictable, leading to a 'cluster' of GPS points. A last incident that can occur is GPS signal loss. This occurs when the user's signal is lost, and a new signal is only received at a later time stamp, causing a jump. A second cause that can lead to signal loss, which especially applies to fitness trackers, is the ability to pause an activity. When the activity is resumed again, there will be a jump in GPS locations, which may lead to miscalculation of distance.

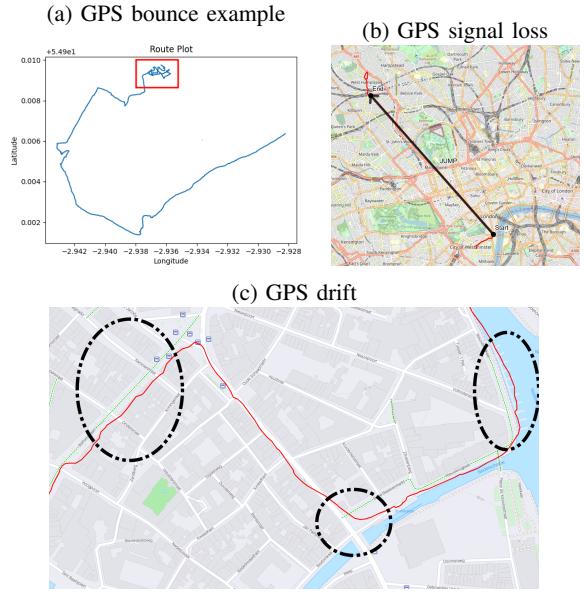


Fig. 3: Possible GPS-faults

Some techniques can be used to improve the overall accuracy of the GPS data, and so improve the effectiveness of the attack. The main technique used in this thesis is *smoothing*, and the hypothesis is that fitness platforms use this technique as well. This is a technique to enhance the accuracy and reliability of GPS data. There are different implementations of GPS smoothing, but the one used in this context is moving average filtering. This method calculates an average position by considering a sliding window of the most recent GPS measurements. By averaging multiple measurements over a certain time period, the effects of noise and temporary inaccuracies

can be mitigated, resulting in a smoother and more reliable trajectory. The size of the window can be chosen. The larger the window, the less accurate the trajectory will be, but the more noise will be countered. This technique is especially useful to counter GPS bouncing and GPS drift.

### C. Endpoint Privacy Zones

An important privacy-enhancing mechanism is the use of an *Endpoint Privacy Zone* (EPZs). An EPZ is a circular zone with a certain radius around a GPS point, which represents a sensitive location. The radius of this circle can be chosen by the user. In the case of Strava, users have the option to select values ranging from 0 to 1600m, in increments of 200m. When a user starts or finishes their activity within this zone, that specific part of the route within the EPZ will not be visible to others. From another user's perspective, the activity will appear to start and/or end at the edge of this circle (which, of course, is not visible). It's important to note that if a user passes through the EPZ without stopping within it, that segment of the route remains unmodified.

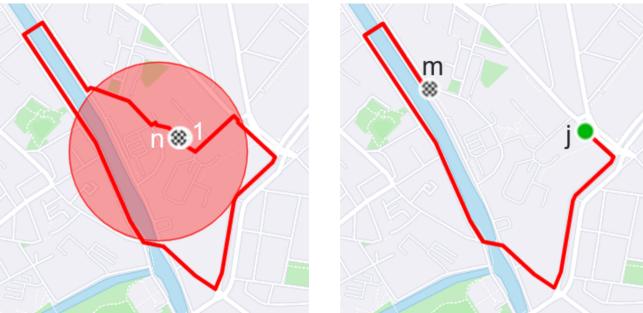


Fig. 4: Example EPZ filtering mechanism [1]

### D. Related work

There has been previous research conducted on the effectiveness of EPZs in fitness trackers. Hassan et al. (2018) presented an implementation of EPZs where the sensitive location serves as the center of the zone [6]. This means that in this scenario by identifying this zone, one can determine the sensitive location. However, in contrast to most real world implementations of EPZs, it is assumed that the center does not undergo any translation, and therefore no spatial cloaking is applied. Dhondt et al. (2022) also conducted a study on potential vulnerabilities in the concept of EPZs [1]. The paper places particular emphasis on the translation of the EPZ and its impact on user privacy. It introduces an inference attack that exploits the total distance covered in an activity. In brief, the attack operates as follows: by leveraging the total distance traveled and combining it with the road network in the given environment, an attempt is made to reconstruct all possible routes that the athlete could have taken. This analysis is performed for each activity. By comparing these reconstructed routes, it becomes possible to predict a location that is deemed most likely to be the sensitive location.

## III. SETTING OF THE ATTACK

### A. Threat model

This thesis focuses the feasibility of bypassing EPZs from the perspective of an attacker, who is a user of the platform without ownership of the activity data. Given that activities are cloaked using EPZs, the attacker lacks visibility into the actual start and/or end locations. Consequently, their objective is to determine the sensitive location despite the presence of cloaking. It is important to note that the attacker in this thesis does not have access to assistance data or state data. However, they do have access to raw GPS data, as well as additional information such as speed and pace.

The research aims to explore the extent to which an attack is still possible when the distance data is rendered unusable. Therefore, an alternative approach to the inference attack is investigated to determine its potential success. The thesis presents a theoretical framework to describe the attacker and examines the circumstances under which the attack remains feasible, as well as effective countermeasures. The overall scenario assumed is: what if fitness trackers were to obfuscate or make the distance data unusable through techniques like rounding or adding uncertainty? In such a case, would the attack still be possible? If so, what would be its effectiveness and how would it impact the previously discussed protective measures?

To enable the attack, certain assumptions need to be made. The first assumption is that the visible start and end points must lie on the edge of the EPZ circle [1]. Secondly, the protected location, which is the sensitive location, must be located on the road graph. It cannot be outside the mapped area, such as in a forest where there are no paths. The user is expected to follow the shortest route within the EPZ. Additionally, this thesis relies on average speeds and paces, leading to the proposal of an additional assumption: the user should not remain stationary within the EPZ. Lastly, it is assumed that activities that involve two concealed locations, such as both start and endpoints being hidden, are not usable for the purposes of this thesis.

### B. Inference Attack

The actual attack can be broken down in seven steps.

- 1) The first step is to identify the EPZ, which, although not mandatory, significantly narrows down the search space. In this process, all the activities made available by a user are considered. The visible start and end points of these activities are extracted and then grouped together using the k-means algorithm. This grouping helps to form a circle that represents the EPZ.
- 2) Then we can move on to the identification of Entry Gates (E.G.). Entry gates refer to the zones where users can enter or exit the EPZ. These gates are typically located around roads that lead into the EPZ. Identifying these entry gates is crucial for filtering out anomalous activities. The detection of entry gates is accomplished using the

- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm.
- 3) For each identified EPZ, it is necessary to create a graph representation of the surrounding area. The graph representation consists of a series of nodes, all located on a known street. The edges connecting the nodes follow the street layout, representing possible routes [8]. Based on the nodes in this graph, the Distance Matrix can be constructed. This matrix contains the theoretical distances from all starting nodes (on the boundary of the EPZ) to all nodes present in the graph. By utilizing the Dijkstra algorithm, it becomes possible to determine the shortest theoretical distance from each point to all other points in the graph. These distances are stored and are crucial in the later stages of the attack.
  - 4) To make an effective prediction, it is crucial to know the distance traveled within the EPZ. This is referred to as the inner distance. Two possible scenarios apply in this thesis: one where the cumulative distance is known, from which we can infer the distance traveled outside the EPZ, and another where this distance is unknown. In the first scenario, the conversion is performed using the following equation:  $\text{inner distance} = \text{total time} \times \text{average speed} - \text{outer distance}$ . However, if the distance is unknown, the outer distance needs to be calculated externally using visible GPS locations. The outer distance utilizes the Haversine formula to calculate distances between two points on a spherical surface [9].
  - 5) Prior to predicting the location, it is important to filter out activities that cannot yield useful predictions. We aim to exclude all other activities as much as possible. In cases where a user does not follow the shortest route from the EPZ boundary to the sensitive location, we can partially address this by considering an activity only if the remaining distance within the EPZ is smaller than the maximal possible distance to be covered. Similarly, filtering can be applied for distances traveled that are lower than the minimum possible distance. Furthermore, the visible start and end points of activities are checked for compatibility with the road graph. If the difference in distance between the original location and the snapped location is too large, the activity is filtered out. Lastly, deviations in the E.G. are examined. If there is a deviation between the visible start and end points and the E.G. that exceeds three times the standard deviation, the activity is filtered out.
  - 6) The next step is to predict the sensitive location. To make a prediction for each activity, the calculated inner distance is used. This inner distance is then matched with the street network. The idea behind this is to traverse all possible routes (forming the shortest path to the nodes on the path) within the EPZ and stop when the traveled distance matches the calculated inner distance.
  - 7) To transform the routes determined in the previous step into a final prediction, regression analysis is applied using the Least Absolute Deviations (LAD) method.

The outcome of this regression analysis will be a GPS location, which will form our final prediction.

Note that this attack is very similar to the attack proposed by Dhondt et al. and Verdonck T. [1, 5].

#### IV. USED DATA

It is crucial to use a representative dataset in order to draw meaningful conclusions and identify potential deviations or irregularities. By examining the characteristics of the data, we can form well-grounded conclusions that take into account certain properties of the data. Since this thesis builds upon the research of Dhondt et al., it is convenient to continue working with their dataset. In total, a dataset of 4000 users was collected. However, this thesis only experiments with a subset of 131 users, with 101 users used for analyses and conclusions, and 30 users reserved for testing the attack.

##### A. Geographical distribution

A geographical distribution is visible in Figure 5. It clearly shows that most activities are located in Central Europe. Additionally, there is a noticeable concentration in the United States. To a lesser extent, there are also activities in Australia and South America. The dataset exhibits a relatively broad spread of activities worldwide, providing a solid foundation for testing the attack. However, it is important to note that the fraction of the dataset we have access to, with 101 users, is relatively small, which may result in a distorted representation of reality.

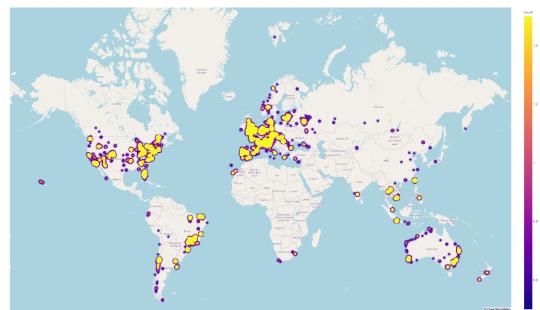


Fig. 5: Geo-heatmap of the users in the dataset

##### B. Users and activities

Table I displays several global statistics regarding users and their associated activities in the dataset. Figure 6 shows the CDF plot, illustrating the number of activities per user. It is notable that there is a significant number of activities available per user in the dataset. However, it's important to note that the dataset, with an average of 411 activities per user, is not entirely representative of reality. When comparing these figures with data from a study conducted by Strava itself in 2020, there seems to be a mismatch.

	Amount
Total # users	101
Total # activities	41 554
Average # activities per user	411
Median of the # activities per user	296
Maximal # activities for a user	2946
Minimaal # activities for a user	31

TABLE I: Overview of users and activities

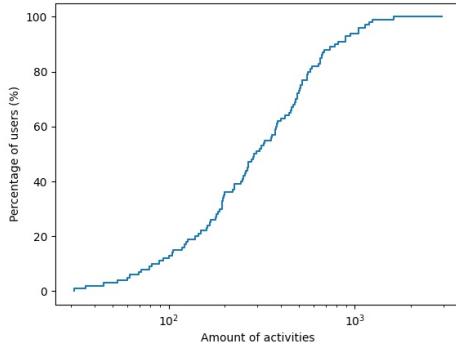


Fig. 6: CDF plot of the amount of activities per user

### C. GPS anomalies

Given the importance of GPS data in this study, which has a high chance of containing errors, it is crucial to analyze the dataset for potential deviations. First, we examine the presence of GPS errors in the form of signal losses or pauses. This is done by studying the distance between consecutive GPS points. Figure 7 depicts the distribution of these distances. The average distance between consecutive locations is 6.41 meters, with a standard deviation of 42.53 meters. The average value is relatively low, indicating potentially accurate data. However, the high standard deviation suggests significant fluctuations. On the graph and in the table, it can be observed that most distances fall below 20 meters, which again indicates decent precision. However, there is a small portion of GPS points that exhibit large inter-point distances. Given the magnitude of the number of GPS points and an average number of points per activity of 2574.90, this cannot be overlooked.

To determine the number of GPS deviations in the dataset, we also examine the difference between the calculated distance traveled within the EPZ (obtained by subtracting the visible trajectory from the total distance) and the theoretically traveled distance within the EPZ, which can be read from the dataset through the cumulative distance. An initial visualization is shown in Figure 8. The figure illustrates the fluctuations between the manually calculated distance and the theoretical distance for one user. The peaks indicate significant deviating calculated distances, indicating large GPS errors. Additionally, the less noticeable fluctuations also indicate significant inaccuracies between the calculated and theoretical distances. The differences in calculations for the entire dataset are presented in Figure 9. The graphs reveal that there are indeed many

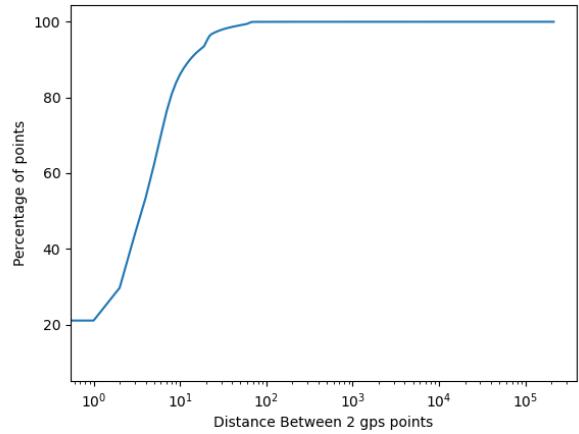


Fig. 7: Distribution of distances between two consecutive GPS points

significant differences present.

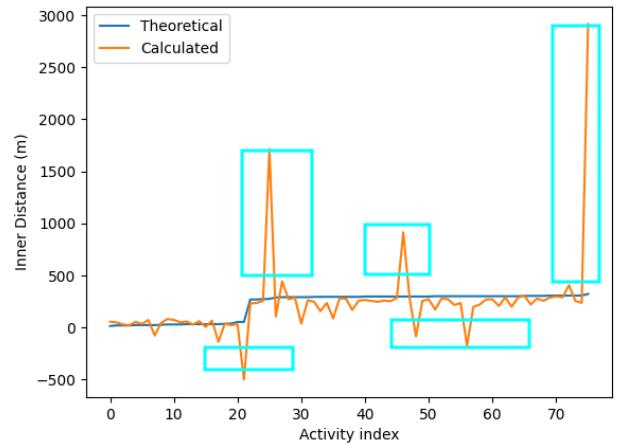


Fig. 8: Difference between the calculated distance and the theoretical distance for a single user

## V. EVALUATION MECHANISM

We will test and evaluate the attack on public activities that do not contain an EPZ, but will manually provide them with an EPZ. This way, we can compare the obtained results with a reference, namely the ground truth (GT).

### A. ground truth

The ground truth of a user is their actual place of residence or the location they typically depart from or arrive at. This is the location we consider as the center around which the EPZ is applied and the one we ultimately aim to determine. We determine this location by examining all activities of a user and adding the starting or ending points that are within a radius of 50 meters to the same cluster using the DBSCAN

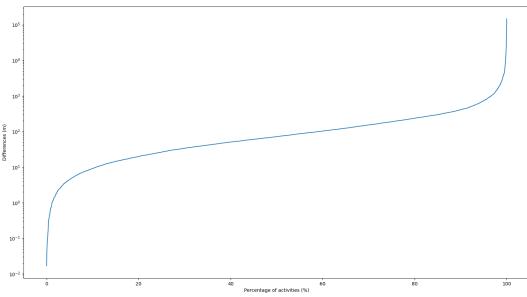


Fig. 9: Distribution of the difference between the calculated distance and the theoretical distance outside the EPZ

algorithm. Please note that it is possible for a user to have multiple ground truths.

A second caveat we need to mention is that the actual starting and ending locations may not always align perfectly with the road network. We subsequently map them to the street network, but during the upload process, the platform in question calculates the total distance traveled to the actual starting point. This can introduce deviations in the predictions. Dhondt et al. conducted a study to determine the average deviation and obtained a threshold of 22.95 meters to classify a successful attack [1].

#### B. Manually adding EPZs

As mentioned earlier, we are working with public activities that do not contain an EPZ in order to simplify the evaluation process. However, in order to perform the attack, we still need to manually introduce an EPZ. An EPZ is determined by a central point (the sensitive location), which will undergo a random translation, and a chosen radius. From the translated point, a circle is drawn with the corresponding radius. We start from the ground truth location, which then undergoes a random translation. The shift of the point can occur in any direction and is chosen randomly. The distance of the translation can, in principle, be chosen arbitrarily, but it must fall within certain limits, namely between 0 and 70% of the radius of the EPZ. The circle is then constructed with the translated point as the center and the associated radius. All points within this zone will be removed from the activity.

#### C. Bootstrapping

During the testing procedure of the attack, it is not simply performed once for all activities per user. For each user, a confidence interval is calculated using bootstrapping. The set of manually obfuscated activities is considered. The bootstrap algorithm randomly selects one activity at a time from this set and places it in a new group until this new group of activities is the same size as the original set of activities. It's important to note that the algorithm may choose the same activity multiple times, so the newly created group may contain duplicates and may not include other activities at all. This process is repeated

1000 times, resulting in 1000 different sets. For each set, a prediction is made, yielding a number of predicted locations, some of which may be predicted multiple times.

#### D. Evaluation metrics

To provide meaningful insights into the effectiveness of the attack, eight metrics are defined to evaluate the attack. These metrics are chosen to align with the metrics used in the studies by Dhondt et al. and Verdonck, enabling clear comparisons with their results [1, 5].

The *Success Rate* is defined as the percentage of performed attacks in which the sensitive location is successfully determined. Taking into account the overshoots that may occur when snapping locations to the road network, a correct location is considered to be within a radius of 22.95 meters from the ground truth. A higher percentage indicates a more successful attack.

The *Correctness* of an attack is calculated as the sum of the Euclidean distances between the ground truth and the predicted location, divided by the number of times that location was predicted. This metric provides an indication of the average deviation in distance of the predicted locations from the ground truth. A lower value indicates a more precise attack.

The *Accuracy* in this context is defined as the width of the confidence interval. It refers to the number of unique predictions, which represents the number of nodes that are predicted exactly once. A higher number of unique nodes indicates higher accuracy, but this also implies less certainty in our predictions.

The *Reduction of the k-anonymity set* quantifies the decrease in the set of all possible end locations before and after the actual predictions of an attack. The possible end locations before the attack are simply all nodes in the graph representation, potentially limited by the EPZ. The ones after the attack are the nodes that are effectively predicted. The reduction is therefore a percentage that indicates the difference between the two sets. A higher reduction percentage implies a more significant reduction in the set of possible end locations, indicating a higher precision of the attack.

The *Uncertainty Region ( $m^2$ )* is the sum of the areas of the union of the uncertainty regions around the predicted nodes. In this context, the uncertainty regions are caused by the chaining distance, which is assumed to be three meters. Since nodes exist only at intervals of three meters, each predicted node has a chance of representing a point that lies somewhere within a three-meter zone around that node. The Uncertainty Region metric quantifies the combined uncertainty associated with these predicted nodes by calculating the total area of their uncertainty regions.

The *Certainty metric* quantifies the concentration of the probability distribution. A higher Certainty value indicates that the nodes in the probability distribution are spread further apart, indicating a greater level of certainty or confidence in the predictions. Essentially, it measures the degree of dispersion or concentration of the predicted locations in the probability distribution.

*Spatial Certainty* is a metric that considers the density or concentration of neighboring nodes in the vicinity of each predicted node, instead of focusing solely on the probability of each individual node. It takes into account the density or clustering of predicted locations in the neighbourhood of each node. A higher Spatial Certainty value indicates a higher density or concentration of predicted nodes in the local vicinity, suggesting a higher level of certainty in the predicted locations within their respective neighbourhoods.

The *Degree of Anonymity* is a metric that quantifies the level of anonymity achieved by the attack. It is calculated as the normalized entropy of the expected distribution, where the entropy represents the amount of uncertainty or randomness in the distribution of predicted locations. By normalizing it based on the maximum possible entropy, the Degree of Anonymity provides a measure of how much information is leaked or preserved by the attack.

## VI. RESULTS

Each of the four described scenarios will be discussed separately and compared to each other. The difference in results is visible on Figure 10. The individual results are given by the tables in Appendix A. In general, a similar trend is observed across the models when changing the EPZs. As the radius increases, the success rate decreases due to the inclusion of more nodes. More nodes lead to increased potential confusion in the LAD regression. This also results in a greater degree of anonymity and uncertainty region. The number of predictions does not increase proportionally with the number of nodes in the graph as the size expands, leading to an increased reduction. Lastly, it is noticeable that the correctness also improves with a larger radius. This is attributed to a higher probability of violating one of the specified assumptions.

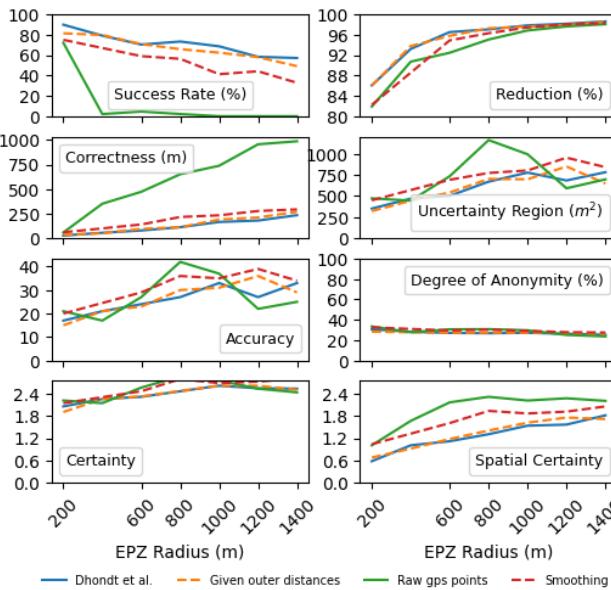


Fig. 10: Comparison of different attack models

### A. Model according to Dhondt et al.

The first model we test is the model by Dhondt et al. [1]. We use these results as a reference for the rest of the findings. The model by Dhondt et al. does not have any restrictions regarding the available data, so it can utilize all the data. It is therefore expected that this leads to good scores.

### B. Given outer distance

This model is based on the availability of cumulative distances. One advantage of this model is that it does not require GPS data. We observe a similar trend as in the model by Dhondt et al., with slight declines in most scores compared to their results. There is only one additional step necessary compared to the model by Dhondt et al., which involves converting speed and time to total distance. This conversion accounts for the minor decreases and increases in the results. The conversion process may introduce small deviations, likely due to rounding errors and additional calculations performed by Strava in determining the speed.

### C. Raw GPS data

The next attack scenario is performed without using cumulative distance and without smoothing. This means that the attacker directly utilizes the raw GPS data for calculating the outer distance. In this case, both the deviation resulting from the speed conversion discussed in the previous model (where the outer distance is given) and the deviations originating from the GPS data itself are incorporated in the results.

As expected, these deviations have a relatively strong impact. Especially for larger radii, they significantly affect the results. From a radius of 1000 meters onwards, the success rate drops to 0%. We also observe a substantial decline in the scores of other metrics for higher EPZ radii. This can be attributed to the significant deviations introduced by the raw GPS data, which have a particularly noticeable effect at larger radii. The longer the distance to be traveled, in this case the inner distance, the greater the error.

### D. Smoothing

The final model is similar to the previous one, with the difference that smoothing is now applied to the routes in an attempt to reduce the deviations caused by GPS data. Smoothing helps to flatten out the extremes and should reduce errors to a lower order. This is reflected in the results. The success rate shows a relatively small improvement compared to the model using raw GPS data for smaller EPZs, but for larger EPZs, the metrics exhibit much less deterioration. The other metrics also demonstrate a similar pattern, with the attenuation being much less pronounced than when using raw GPS points. This indicates that the use of smoothing provides a significant added value. However, it's important to note that the optimal smoothing window size was determined empirically and tailored specifically to this dataset. For a different dataset, the optimal window size may vary.

## VII. CONCLUSION

This thesis demonstrates that inference attacks are possible based on speeds and GPS data, albeit with a lower success rate and increased uncertainty. If cumulative distance is available, a simple conversion (*inner distance = total time × average speed – outer distance*) allows the attack to be performed with a success rate of up to 81.43%, which is deemed acceptable. If cumulative distance data is not available, an additional conversion can be performed using GPS locations to still achieve a successful attack. By incorporating smoothing algorithms, the attack can also gain precision, resulting in a success rate of up to 75.0% for the respective radii and a smoothing window of 100. While this success rate is acceptable, it is lower than the success rate achieved by Dhondt et al. Hence, certain distance data, specifically cumulative distance and total distance, are not crucial for the successful execution of this attack model.

Dhondt et al. proposed several measures to ensure user privacy, such as rounding distance data or adding noise to make it unusable [1]. However, our attack model can bypass all the *Distance Focused Countermeasures* described by Dhondt et al., except for the *Shifting distances* countermeasure. The four distance focused countermeasures are:

- *Generalization* involves rounding the distance to a certain precision. Dhondt et al. suggests a precision of 500 meters.
- *Noisy Distances* is a technique where a random value is added or subtracted from the total distance.
- *Shifting Distances* shifts the visible start or end points of an activity by a random distance in a random direction, making the starting point uncertain.
- *Truncation* states that the hidden segment is not taken into account in the total distance. This effectively removes the hidden segment entirely from the activity.

The attacker can theoretically recompute the distance data and ultimately still succeed in the attack. In principle, we could expand these countermeasures by incorporating speed and/or time. For example, in generalization, which involves rounding a visible distance, the speed can undergo a similar manipulation. By considering additional factors beyond distance, we can enhance the effectiveness of privacy countermeasures.

In the case of the other category of countermeasures, namely *EPZ-Focused Countermeasures*, it is expected that these still remain effective countermeasures. These countermeasures include:

- *Increasing EPZ radii*, as the name suggests, involves enlarging the EPZ. We observe that the attack performs significantly less effectively with larger EPZ radii.
- *Complex EPZ shapes* implements an EPZ that is no longer a simple circle but takes on a more intricate form. This can include shapes such as polygons.

## VIII. FUTURE WORK

Some interesting future studies could be conducted to further improve the attack.

Implementing road snapping as a first possibility could enhance the accuracy of location data by snapping GPS points to the nearest road or path. This can help eliminate outliers and improve the overall quality of the data used in the attack model.

Using a dynamic window for smoothing is also a valuable addition. By adjusting the window size based on factors such as the density of GPS points or the speed of movement, the smoothing algorithm can better adapt to different scenarios and reduce the impact of outliers or irregularities in the data.

Furthermore, conducting an additional analysis to identify the external circumstances under which the attack is most successful is a proactive approach. This analysis can help identify patterns or vulnerabilities that contribute to the effectiveness of the attack model. By understanding these factors, appropriate measures can be taken to mitigate the risk and enhance user privacy in specific contexts or situations.

## REFERENCES

- [1] K. Dhondt, V. Le Pochat, A. Voulimeneas, W. Joosen, and S. Volckaert, “A run a day won’t keep the hacker away: Inference attacks on endpoint privacy zones in fitness tracking social networks,” [osf.io/3m5ut/](https://osf.io/3m5ut/), Nov 2022.
- [2] D. Vanmeldert, “Sportapp strava laat fietsdieven of stalkers nog altijd meekijken — vrt nws: nieuws,” <https://www.vrt.be/vrtnws/nl/2022/10/28/strava-kul/>, October 2022, (Accessed on 02/20/2023).
- [3] A. Bowden, “Cyclist who had five bikes stolen says thieves are looking for quick times on strava to try and find high-end bikes — warns other users to check their privacy settings — road.cc,” <https://road.cc/content/news/248798-cyclist-who-had-five-bikes-stolen-says-thieves-are-looking-quick-times-strava>, September 2018, (Accessed on 02/20/2023).
- [4] A. Hern, “Fitness tracking app strava gives away location of secret us army bases — gps — the guardian,” <https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases>, January 2018, (Accessed on 02/20/2023).
- [5] T. Verdonck, “Inferentie-aanvallen met hoogteprofielen tegen (endpoint) privacy zones in fitness tracking sociale netwerken,” Master’s thesis, KU Leuven. Faculteit Industriële Ingenieurswetenschappen, Leuven, 2022, book Title: Inferentie-aanvallen met hoogteprofielen tegen (endpoint) privacy zones in fitness tracking sociale netwerken.
- [6] A. B. Wajih Ul Hassan, Saad Hussain, “Analysis of privacy protections in fitness tracking social networks -or- you can run, but can you hide?” August 2018. [Online]. Available: <https://osf.io/3m5ut/>
- [7] “Strava logo,” Strava. [Online]. Available: <https://www.strava.com>
- [8] M. Neira and R. Murcio, “Graph representation learning for street networks,” 2022.
- [9] W. Sheppard and C. Soule, *Practical Navigation*. World Technical Institute, 1922. [Online]. Available: <https://books.google.be/books?id=8S0wAAAAYAAJ>

## APPENDIX

### A. Individual attack results

Radius (m)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	89.86	28.61	17	86.06	352.08	2.06	0.58	30.69
400	79.1	56.82	21	93.21	469.76	2.26	1.01	28.31
600	70.37	79.94	24	96.52	502.42	2.32	1.12	27.15
800	73.33	113.68	27	97.05	670.53	2.47	1.31	26.86
1000	68.64	166.74	33	97.84	777.57	2.62	1.54	27.25
1200	58.33	180.97	27	98.15	684.71	2.55	1.57	26.25
1400	57.14	235.76	33	98.61	782.30	2.54	1.82	25.31

TABLE II: Attack according to the model by Dhondt et al. [1]

Radius (m)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	81.43	35.96	15	86.01	322.32	1.91	0.68	28.33
400	79.71	51.38	21	93.78	445.30	2.26	0.92	27.80
600	70.77	96.94	23	95.78	542.48	2.33	1.18	27.34
800	65.83	113.18	30	97.28	703.00	2.48	1.41	27.38
1000	62.39	191.47	31	97.60	698.69	2.62	1.62	27.31
1200	57.98	212.06	36	97.86	850.01	2.62	1.76	27.13
1400	49.15	270.35	29	98.54	648.70	2.51	1.72	24.90

TABLE III: Attack based on given *outer distance*, and speed

Radius (m)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	72.06	59.92	21	81.89	473.05	2.22	1.01	33.43
400	2.08	351.85	17	90.71	446.35	2.15	1.67	27.80
600	4.55	473.15	27	92.46	734.62	2.57	2.17	30.67
800	2.13	651.38	42	95.06	1161.95	2.87	2.32	30.84
1000	0.00	737.93	37	96.84	994.80	2.76	2.22	29.69
1200	0.00	955.79	22	97.63	592.09	2.54	2.28	25.16
1400	0.00	986.46	25	98.08	697.50	2.44	2.21	23.70

TABLE IV: Attack based on raw GPS locations (no smoothing) and speed

Radius (m)	Smoothing Window (n)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	100	75.0	61.37	20	82.22	450.15	2.15	1.04	32.57
600	100	58.97	141.04	29	94.89	692.52	2.47	1.61	29.51
800	100	56.34	217.13	36	96.30	773.61	2.80	1.94	30.30
1000	100	41.27	234.27	35	97.43	802.93	2.69	1.87	29.13
1200	100	44.12	278.00	39	98.06	953.93	2.73	1.92	27.86
1400	100	32.81	294.24	34	98.28	841.94	2.82	2.06	27.51

TABLE V: Attack based on smoothed GPS data and velocity, with an empirically determined optimal smoothing window  $n = 100$

**Bijlage C**

**Poster**

# Time Is Running Out

## Assessing Temporal Privacy of Privacy Zones in Fitness Tracking Social Networks

### Situering

Fitnesstrackers zoals Strava zijn niet meer weg te denken uit deze moderne wereld. Dit zijn sociale netwerken die sportactiviteiten zoals lopen en zwemmen kunnen registreren, en delen met vrienden. Deze brengen echter ongewilde privacy bezorgdheden met zich mee. Gedurende deze thesis wordt onderzoek gedaan naar het ongewild vrijgeven van persoonlijke locaties (woonlocaties, militaire basissen, ...) ondanks de maatregelen van Strava.



### Vraagstelling

- Strava maakt gebruik van *Endpoint Privacy Zones* (EPZ) om gevoelige locaties te verbergen. Dit mechanisme bestaat eruit om alle routesegmenten binnen een bepaalde radius te verbergen. De vraag is of het mogelijk is om deze privacy zone te omzeilen zonder gebruik te maken van afstandsdata, maar door gebruik te maken van snelheidsdata.



Endpoint Privacy Zones

### Werkwijze

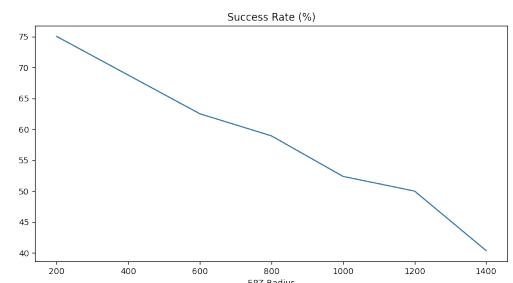
- Aan de hand van snelheden buiten de EPZ en de **totale gemiddelde snelheid**, wordt de totale afgelegde afstand bepaald. Via **GPS-coördinaten** wordt kan ook de totale afstand buiten de EPZ berekend worden. Beide afstanden afgetrokken van elkaar geven de afstand binnenvin de EPZ. Deze afstand kan met behulp van het **wegennetwerk** gebruikt worden om de verborgen route te reconstrueren



Mechanisme Aanval

### Resultaten

- Het is zeker mogelijk om in een groot deel van de cases via deze gedachtengang gevoelige locaties te achterhalen. Afhankelijk van de omvang van de zone kan een succes rate van 75% bekomen worden. Dit wil zeggen dat voor 75% van de gevallen de locatie kan bepaald worden tot op 22,95m nauwkeurig.
- De precisie van een aanval gaat echter drastisch achteruit bij het verhogen van de grootte van de EPZ.



Resultaten aanval in functie van grootte EPZ

FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN  
TECHNOLOGIECAMPUS GENT  
Gebroeders De Smetstraat 1  
8200 GENT, België  
tel. + 32 50 66 48 00  
[iiw.gent@kuleuven.be](mailto:iiw.gent@kuleuven.be)  
[www.iiw.kuleuven.be](http://www.iiw.kuleuven.be)

