

# 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. Als laatste had ik ook graag enkele van mijn kotgenoten bedankt voor de nodige afleiding tijdens de stressvolle perioden gedurende het academiejaar. In het bijzonder Angelo Pattyn en Jakob Sabbe, die zelf ook aan hun thesis werkten! Ook Sam Boeve, voor de erg hulpvolle adviezen gedurende het 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 mogelijk 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 locaties 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 en Garmin worden gelijkaardige privacy features geïmplementeerd. Bijvoorbeeld het verbergen van activiteiten voor andere gebruikers, of enkel activiteiten weergeven voor je volgers. Maar een ander veelgebruikte techniek is gekend als het implementeren van *EPZ's* (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 gevoelige 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 waterdichte implementatie, want hierbij worden bijhorende gegevens niet aangepast of mee verborgen. Bijvoorbeeld wordt de totale afgelegde afstand van een activiteit 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 afstand van de activiteiten, in combinatie met het stratenplan van het gebied. Dit soort aanvallen

---

worden *inferentieaanvallen* genoemd. Het traject afgelegd binnennin de EPZ kan worden afgeleid met behulp van de totale afstand van de activiteit en de zichtbare afstand, afgelegd buiten de EPZ. De afstand binnennin de EPZ kan worden gemapt op het stratenennetwerk, om zo alle mogelijke routes te bekomen die de gebruiker kan afgelegd hebben binnennin 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 inferentie-aanvallen wanneer de afstand niet gekend 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 wordt de gemiddelde snelheid en de totale duur gebruikt 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, worden smoothing- en map-matchingstrategieën bestudeerd om de best mogelijke resultaten te verkrijgen. Deze twee berekende waarden kunnen 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 hebben we kunnen aantonen dat een dergelijke aanval mogelijk is en een aanzienlijke nauwkeurigheidsscore behaalt, ondanks het ontbreken van de totale afstand.

**Kernwoorden:** gps-locaties, privacy, endpoint privacy zone, inferentie-aanval, snelheid

# Abstract

In a society where social media is so ubiquitous present, 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 bulletproof. 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, 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, multiple possible locations are 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 the 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 some 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>	x
<b>Figurenlijst</b>	xii
<b>Tabellenlijst</b>	xiii
<b>Lijst met afkortingen</b>	xiv
<b>1 Inleiding</b>	1
1.1 Situering . . . . .	1
1.2 Doelstelling . . . . .	3
<b>2 Achtergrond</b>	4
2.1 Fitnesstrackers . . . . .	4

2.1.1 Activiteiten . . . . .	4
2.1.2 Berekening Afstanden . . . . .	6
2.1.3 Algemeen Privacybeleid . . . . .	9
2.2 Endpoint Privacy Zones . . . . .	10
2.3 Literatuur . . . . .	11
<b>3 Setting aanval</b>	<b>14</b>
3.1 Definitie aanvaller . . . . .	14
3.1.1 Assumpties . . . . .	15
3.2 Identificeren van de EPZ . . . . .	17
3.3 Identificatie Entry Gates . . . . .	18
3.4 Bepalen nodige gegevens voor predictie . . . . .	19
3.4.1 Roadgraph en Distance Matrix . . . . .	19
3.4.2 Begin- en eindnodes . . . . .	20
3.4.3 Berekeningen afstand binnenin de EPZ . . . . .	21
3.5 Voorspellen locatie . . . . .	23
3.5.1 Filteren activiteiten . . . . .	23
3.5.2 Bepalen van de locatie . . . . .	24
3.5.3 Meest voorspelde locatie . . . . .	25
<b>4 Gebruikte data en afwijkingen</b>	<b>26</b>
4.1 Karakteristieken van de gebruikte dataset . . . . .	27

4.2 Mogelijke afwijkingen binnenin de dataset . . . . .	29
4.2.1 Mogelijke fouten bij gps-data . . . . .	29
4.2.2 Gps-fouten in de gebruikte dataset . . . . .	32
4.3 Technieken gps-data te verbeteren . . . . .	34
<b>5 Resultaten en Evaluatie</b>	<b>38</b>
5.1 Evaluatie van de aanval . . . . .	38
5.1.1 De grondwaarheid . . . . .	38
5.1.2 Manueel aanbrengen van een EPZ . . . . .	39
5.1.3 Bootstrapping . . . . .	40
5.1.4 Evaluatie metrieken . . . . .	41
5.2 Resultaten . . . . .	42
<b>6 Conclusies</b>	<b>46</b>
<b>A Uitleg over de appendices</b>	<b>50</b>

# Lijst van figuren

1.1 Voorbeeld Heatmap, vrijgegeven door Strava [31] . . . . .	2
1.2 Voorbeeldactiviteit Strava . . . . .	2
2.1 Verschil snelheid en tempo . . . . .	6
2.2 Data van een activiteit . . . . .	7
2.3 Voorbeeld van de werking van <i>Map Snapping</i> . . . . .	9
2.4 Voorbeeld Data smoothing with moving average . . . . .	9
2.5 Voorbeeld van de werking van een EPZ . . . . .	11
2.6 Voorbeeld translatie EPZ . . . . .	11
2.7 Voorbeeld filtering van punten binnen EPZ . . . . .	12
2.8 Mechanisme EPZ beschreven door Hassan et al. . . . .	12
3.1 Voorbeeld van de mogelijke scenarios bij een total distance attack scenario . . . . .	16
3.2 Voorbeeld van een inner distance attack situatie . . . . .	16
3.3 Voorbeeld werking k-means clustering [21] . . . . .	17
3.4 Voorbeeld van entry gates gevonden door k-means clustering en de identificatie van een EPZ . . . . .	18

---

3.5 Voorbeeld van het genereren van een roadgraph . . . . .	20
3.6 Haversine illustratie voor het berekenen van de afstand [37] . . . . .	22
3.7 Willekeurig voorbeeld van Distance Matrix met een voorspelling van 12,3m . . . . .	24
4.1 Geografische spreiding van de activiteiten in de dataset . . . . .	28
4.3 Post op sociale media van Strava die de evolutie van het totaal aantal activiteiten weergeeft [32] . . . . .	28
4.2 Cumulative Distribution Function (CDF) plot van het aantal activiteiten per gebruiker .	29
4.4 Voorbeelden van gps-drift . . . . .	30
4.5 Voorbeelden van Global Positioning System (gps)-bounce [29] . . . . .	31
4.6 Voorbeeld van zowel gps-drift en gps-bounce uit de gebruikte dataset . . . . .	31
4.7 Voorbeeld van signal loss uit de gebruikte dataset . . . . .	32
4.8 Verdeling van de afstanden tussen twee opeenvolgende gps-punten . . . . .	33
4.9 Verschil tussen de berekende afstand en de theoretische afstand voor één gebruiker	35
4.10 Verdeling van het verschil tussen de berekende afstand en de theoretische afstand buiten de EPZ . . . . .	37
5.1 Dhondt et al. bepaalt grafisch de trend van de afwijkingen bij het snappen van locaties op het wegennetwerk [6] . . . . .	39
5.2 Voorbeeld van een distributie van voorspellingen bepaald door het bootstrapalgoritme [35] . . . . .	41

# Lijst van tabellen

4.1	Overzicht van gebruikers en activiteiten . . . . .	27
4.2	Verdeling van de afstanden tussen twee opeenvolgende gps-punten . . . . .	34
5.1	Aanval volgens het model van Dhondt et al. [6] . . . . .	42
5.2	Aanval op basis van gegeven <i>outer distance</i> , en snelheid . . . . .	43
5.3	Aanval op basis van ruwe gps-locaties (geen smoothing) en snelheid . . . . .	43
5.4	Aanval op basis van gesmoothe gps-data en snelheid . . . . .	44
5.5	Attack result with different smoothing window sizes . . . . .	45

# 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

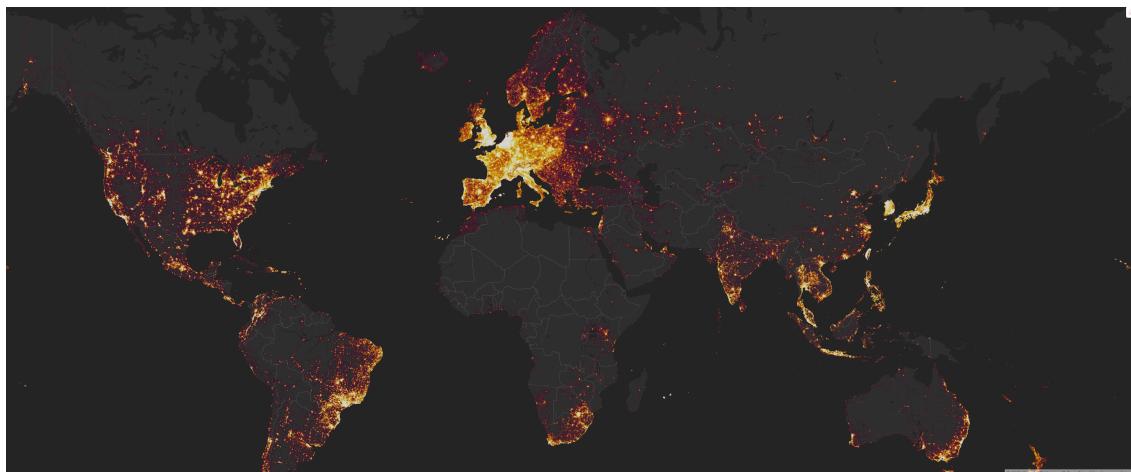
# **Hoofdstuk 1**

## **Inleiding**

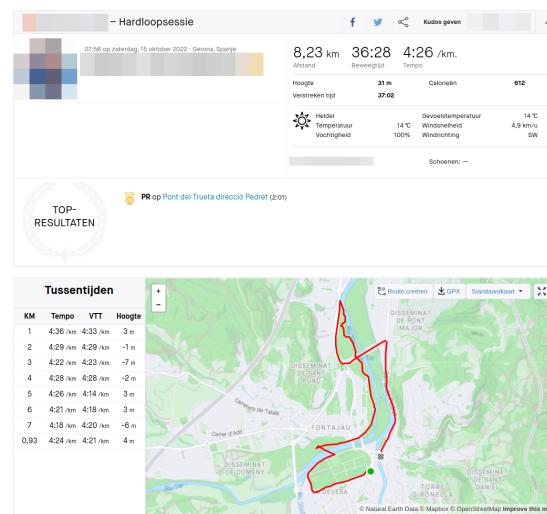
### **1.1 Situering**

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. In het werk van Howard en Park wordt sociale media gedefinieerd 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 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 zou kunnen leiden tot diefstal. Er zijn gevallen bekend van fietsdiefen die Strava gebruiken om fietsen te kunnen lokaliseren [34] [1]. Grootschaligere voorbeelden die zeker het vermelden waard



Figuur 1.1: Voorbeeld Heatmap, vrijgegeven door Strava [31]



Figuur 1.2: Voorbeeldactiviteit Strava

zijn, zijn de gevallen waarbij geheime militaire basissen ontdekt worden door het bestuderen van de heatmap<sup>1</sup> vrijgegeven door Strava [12]. Figuur 1.1 toont een voorbeeld van de heatmap van Strava.

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 EPZs. Hierbij wordt de weergegeven route voor de persoon die meekijkt gedeeltelijk verborgen.

<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 [33]. Het doel is om gebieden met een hoge dichtheid of veelvuldige locaties te identificeren en visueel te markeren.

Er wordt als het ware een deel van de route afgekapt. De echte begin- en eindpunten zullen binnenin het afgekapsdeel 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 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.

## 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 2.2 als privacy beveiligingsmechanisme. In het verleden werden enkele manieren beschreven om a.d.h.v. andere metadata zoals hoogtedata en afstanden de EPZ te omzeilen ([6], [35]). In deze thesis wordt meer in detail gegaan op het gebruik van snelheidsdata. Als basis voor deze aanval gebruiken we de inferentie aanval op de EPZ van 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 is eerst een berekeningsmechanisme nodig voor de afstanden die nodig zijn om de inferentie-aanval te kunnen uitvoeren. Daarna 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 schatten. Er is een analyse van de beschikbare data, en een bespreking en reflectie over de resultaten van de aanval.

# **Hoofdstuk 2**

## **Achtergrond**

### **2.1 Fitnesstrackers**

De focus van deze scriptie ligt op mogelijke tekortkomingen/vulnerabilities betreffende 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, . . . 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. Dit gaat dan meer specifiek 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/>

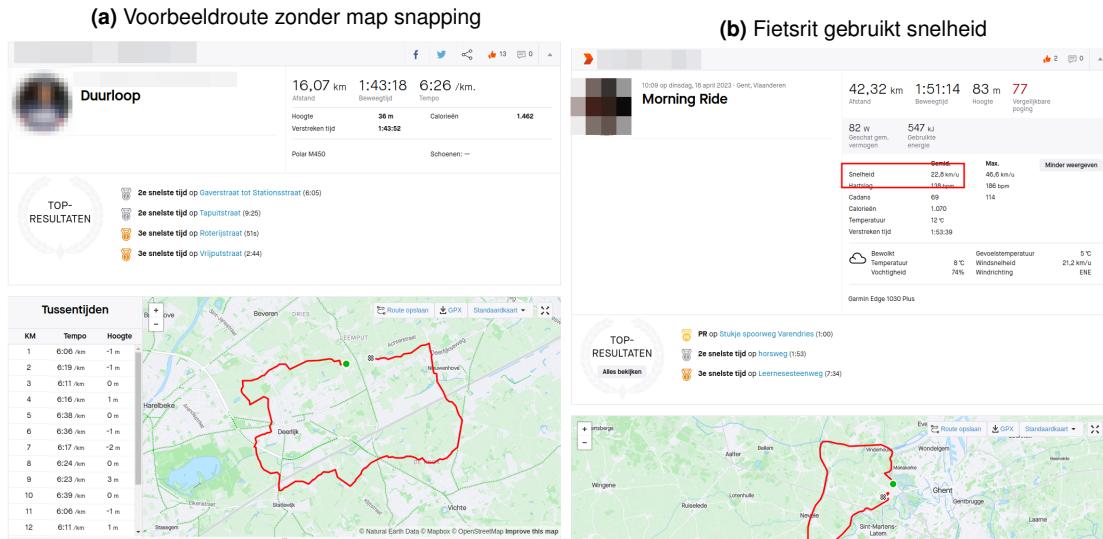
beschrijving aan toe te voegen. Ook een foto kan optioneel toegevoegd worden. De exacte datum en tijd van de start van de activiteit wordt hierbij ook weergegeven.

Rechts daarvan zijn de algemene basisstatistieken te zien. Deze 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 gelabeld zijn als *race*, in dat geval wordt de snelheid berekend in functie van de totaal verstreken tijd [26]. 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  $\text{tempo}(\frac{\text{min}}{\text{km}}) = \frac{t(\text{min})}{d(\text{km})}$ . 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 tempo  $\frac{\text{min}}{\text{km}}$ , 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 door gebruikers van de sport-app kan worden gemarkerd, gedeeld en vergeleken 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 snelheid. Optioneel kan hierbij ook nog een visualisatie van de afgelegde hoogte en de hartslag worden weergegeven, indien de gebruiker hiervoor met de juiste meetinstrumenten zijn sportactiviteit opneemt. De tussentijden en de grafiek van snelheid zijn qua inhoud gelijkaardig,



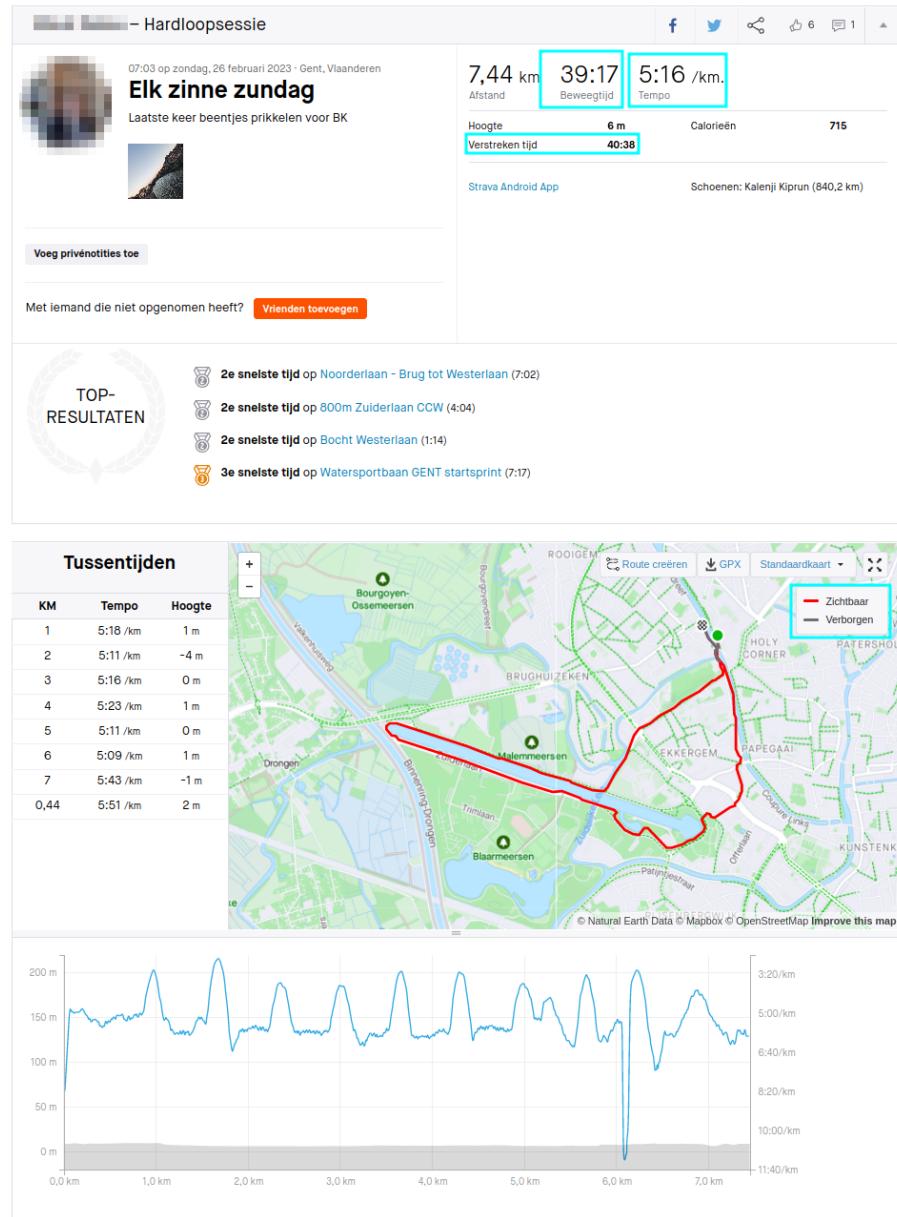
Figuur 2.1: Verschil snelheid en tempo

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 de 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 zijn zichtbare versie op een andere plek starten en eindigen. In de volgende Secties 2.1.3 & 2.2 wordt meer in detail ingegaan op de werking van deze methodiek.

Een laatste kanttekening die hierbij gemaakt moet worden, is dat voor een gebruiker verschillende eenheden mogelijk zijn om uit te kiezen. 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 dus 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 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 is om gps-data pas bij het uploaden te verwerken. De gps-data wordt dan geanalyseerd, en de nodige berekeningen worden uitgevoerd.

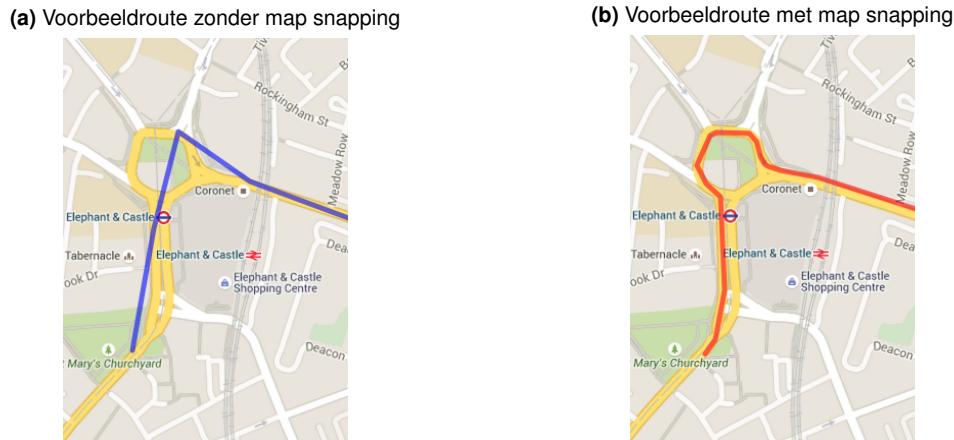
Het alternatief voor de GPS-calculated distance is de *Ground Speed Distance* methodiek. Deze afstand kan enkel worden bepaald in het geval van een fietsactiviteit met behulp van een capabele fietscomputer die omwentelingen van de wielen kan meten. De fietscomputer berekent dan de afstand door het aantal omwentelingen te vermenigvuldigen met de omtrek van het fietswiel [30].

De bovenstaande afstandsberekeningen zijn de twee technieken die de officiële supportdocumentatie van Strava beschrijft [30]. Echter blijkt wanneer we de afstand op deze manier manueel berekenen, afwijkende resultaten bekomen worden. Dit is zeer waarschijnlijk te wijten aan de pre-processing 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 er zullen technieken worden gebruikt om de resultaten hiervan te verbeteren. De twee meest waarschijnlijke technieken zijn *Map Snapping* en *Smoothing*.

Map Snapping of Snap to Roads is een techniek waarbij gps-punten worden verschoven naar de dichtstbijzijnde weg. Per gps-punt wordt dan gezocht naar de dichtste node op de desbetreffende *roadgraph*<sup>2</sup>, op Figuur 2.3 is de werking ervan te zien [4].

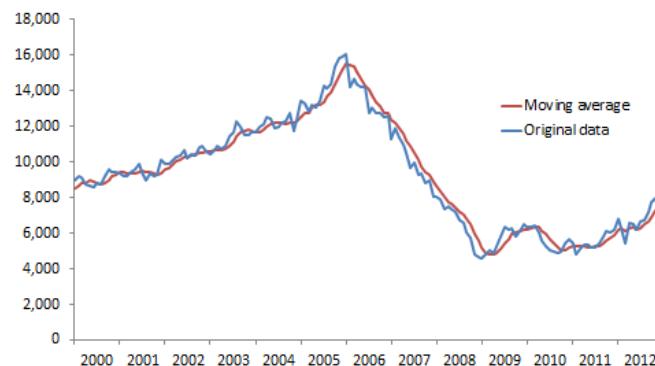
Daarnaast bestaat de kans dat er gebruik gemaakt wordt van smoothing. Smoothing is een proces dat ruwe gps-punten (of datapunten in het algemeen) op een traject probeert te optimaliseren opdat ze een vloeind 'curve' vormen. Dit wordt bekomen door ruis, schommelingen en onnauwkeurigheden te filteren uit 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 afhankelijk van welke implementatie gebruikt wordt voor het snappen. Het 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 [22]



**Figuur 2.3:** Voorbeeld van de werking van *Map Snapping*

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/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 ‘gesmoothed’ curve.



**Figuur 2.4:** Voorbeeld Data smoothing with moving average

### 2.1.3 Algemeen 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*. Als opmerking valt te melden dat in heel wat andere sport-applicaties vergelijkbare, zo niet dezelfde methodieken worden gebruikt. 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 staan meestal drie opties, *zichtbaar voor iedereen*, *zichtbaar voor volgers* en *zichtbaar voor niemand*. Er kan ook zelf een keuze gemaakt worden om specifieke elementen van een activiteit niet te delen met de buitenwereld, zoals bijvoorbeeld de zichtbaarheid van de route op de kaart [28].

## 2.2 Endpoint Privacy Zones

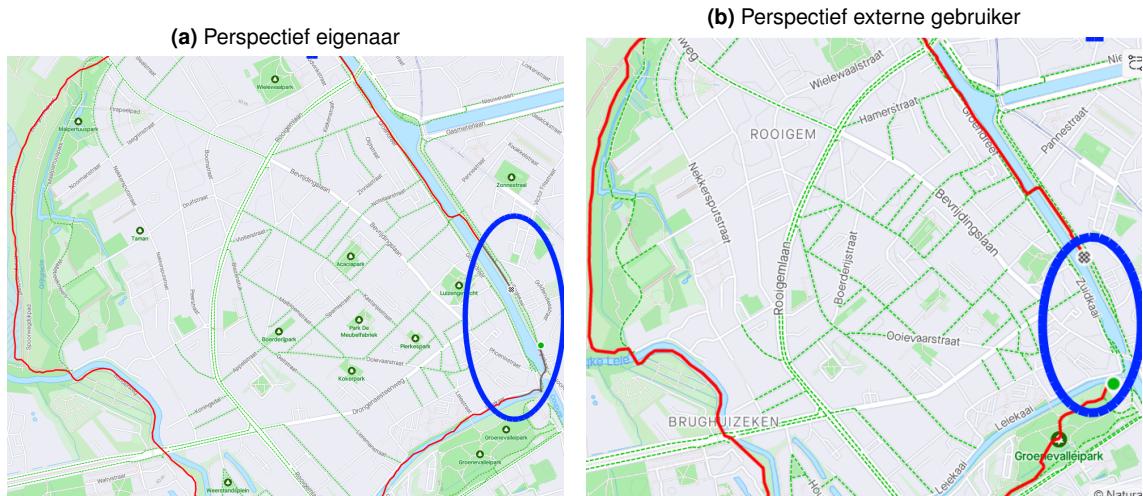
Een tweede belangrijke maatregel is het gebruik van de EPZs. Een EPZ is een cirkelvormige zone met een bepaalde straal rond een gps-punt. Het punt in kwestie zal dus de betreffende *gevoelige locatie* zijn. De straal van deze cirkel<sup>3</sup> kan worden gekozen door de gebruiker, en in het geval van *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 ook andere 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 2.5 zijn de verschillende perspectieven te zien, hoe het er als uploader uitziet, en hoe het eruit ziet voor een andere 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.

De methodiek die fitnesstrackers toepassen bij het opzetten van een EPZ werkt als volgt, de gevoelige locatie wordt genomen als beginlocatie. Hieruit zal a.d.h.v. de op voorhand vastgelegde EPZ-straal een cirkel worden opgesteld. 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 2.6. Het translateren van deze cirkel wordt ook *spatial cloaking* genoemd.

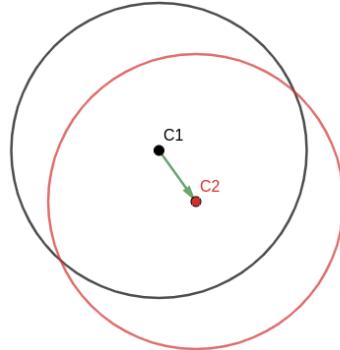
Daarna worden alle punten vertrekende vanaf de gevoelige locatie tot aan de rand van de EPZ,

---

<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.



**Figuur 2.5:** Voorbeeld van de werking van een EPZ



**Figuur 2.6:** Voorbeeld translatie EPZ

en vanaf de rand van de EPZ tot aan de gevoelige locatie verwijderd van het zichtbare traject. Merk op dat punten die de EPZ doorkruisen, maar niet vertrekken of aankomen bij de gevoelige locatie niet worden gefilterd. Een voorbeeld van deze filtering is te zien op Figuur 2.7.

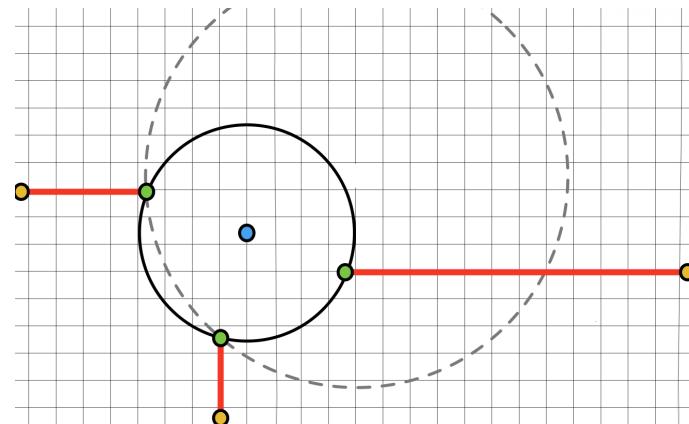
## 2.3 Literatuur

In het verleden is al wat onderzoek verricht in de richting van de doeltreffendheid van EPZs bij fitnesstrackers. Hassan et al. beschreef in 2018 een implementatie van EPZ waarbij het centrum van de zone de gevoelige locatie is. M.a.w. het identificeren van deze zone is dus voldoende om de gevoelige locatie te achterhalen [36]. 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 wordt gefocust op de reconstructie van de cirkel op basis van 3 punten op de rand, wat te zien is op Figuur 2.8. Deze 3 randpunten worden dus bekomen door begin/eindpunten te nemen



**Figuur 2.7:** Voorbeeld filtering van punten binnen EPZ

van activiteiten, volgens het perspectief van gebruiker die geen eigenaar is. Deze begin/eindpunten zullen zich altijd op de rand van de cirkel begeven. Door het toepassen bekwam Hassan et al. een succes rate tot 95.1 Spatial cloaking werd er aangehaald als mogelijke verdediging tegen dit soort aanvallen.



**Figuur 2.8:** Mechanisme EPZ beschreven door Hassan et al.

Een onderzoek door Mink et al. in 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 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. voerde tevens ook een studie in 2022 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. Een inferentie aanval, wordt er beschreven die gebruikmaakt van de totale afstand die terug te vinden bij de activiteit. Het principe van deze inferentie aanval wordt uitvoerig beschreven in Sectie 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.

Deze aanpak toonde aan dat de beschreven countermeasure door Hassan et al. niet feilloos is, en dat deze kan worden omzeild met een successrate van 85%. Aangezien activiteiten nog steeds totale afstanden van een volledige route vrijgeven, kan de afstand afgelegd binnenin de EPZ geïnfereerd worden. Deze data ligt dan ook aan de grond van de inferentie aanval volgens Dhondt et al.

Het meest recente werk in dit domein is de thesis van Verdonck [35]. Deze thesis bouwt in grote mate verder op de paper van Dhondt et al., maar er wordt alternatieve data gebruikt. Er wordt een onderzoek gedaan in hoeverre de resultaten kunnen worden bekomen door het gebruik van hoogtedata om beschermd locaties te achterhalen. Via de kennis van hoogtedata van het stratenplan kan via de gekende hoogteverschillen een inferentie aanval opgezet worden, 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 hoogtevername in heel wat regios niet zo significant, wat de successrate niet ten goede komt.

# **Hoofdstuk 3**

## **Setting aanval**

In dit hoofdstuk wordt de setting alsook de werking van de aanval beschreven. De aanval is sterk gebaseerd op de aanvallen van Dhondt et al. en Verdonck [35, 6]. Deze aanvallen worden inferentie-aanvallen 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 binnenv de privacy zone. Allereerst zal kort de mogelijkheden van een aanvaller in de huidige setting worden besproken. 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 de werking van een aanval wordt 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 zijn. Dit is data zoals afgelegde afstand, snelheid, tempo, ... Aangezien de aanval gaat over het omzeilen EPZs worden activiteiten beschouwd die gecloaked zijn. De aanvaller heeft dus geen zicht op de reële start- en/of eindlocatie. Zijn 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. Deze gebruikt dan voornamelijk afgelegde weg als basis.

De aanvaller die in deze thesis wordt beschreven, 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 onderzoeken in hoeverre een aanval nog mogelijk is wanneer de afstandsdata onbruikbaar zou zijn. Een alternatieve aanpak wordt dus onderzocht om de inferentieaanval alsnog succesvol te kunnen uitvoeren.

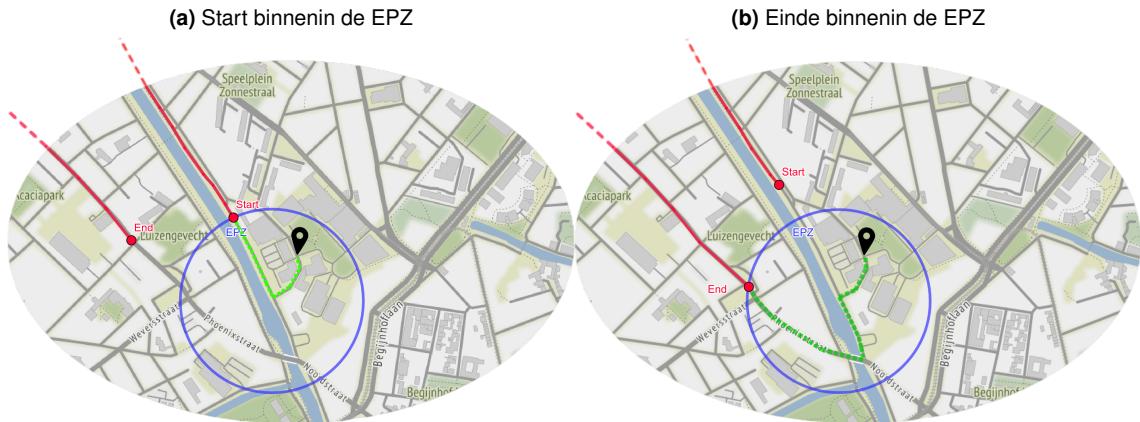
De beschreven aanvaller is beschreven in een theoretisch kader. We bestuderen onder welke omstandigheden de aanval mogelijk is, en welke maatregelen effectief blijken. Het globale scenario waar we van uitgaan is: wat als fitnesstrackers de afstandsgegevens op een bepaalde manier zouden verbergen of onbruikbaar maken, door bijvoorbeeld afrondingen te maken, of onzekerheid toe te voegen.

### 3.1.1 Assumpties

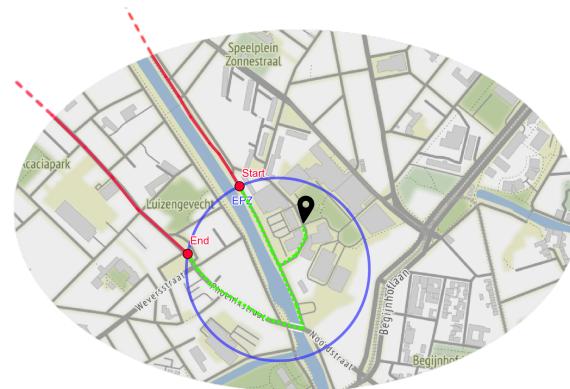
Om de aanval te kunnen uitvoeren, moeten enkele assumpties worden gemaakt. Dhondt et al. maakte al enkele assumpties om de inferentieaanval succesvol uit te voeren. Voor dit onderzoek is het dus logisch om dezelfde veronderstellingen te maken. Zo kunnen we een nuttige vergelijking maken. De eerste assumptie bestaat eruit opdat de zichtbare begin -en eindpunten op de cirkel moeten liggen. Ten tweede moet de beschermd locatie op de roadgraph liggen. Hij kan niet buiten het voor ons te mappen gebied liggen, bijvoorbeeld in een bos waar geen pad ligt. 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. maakt nog een laatste assumptie over start- en eindpunten, die hetzelfde moeten zijn. Dit is echter niet van toepassing op dit onderzoek. Het onderzoek focust zich op activiteiten waar slechts één deel van het traject gecloaked 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 zowel 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 aflegt van de start tot de rand van de EPZ en van de rand van de EPZ tot de eindlocatie is dan ook een vereiste. Op de Figuur is 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 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

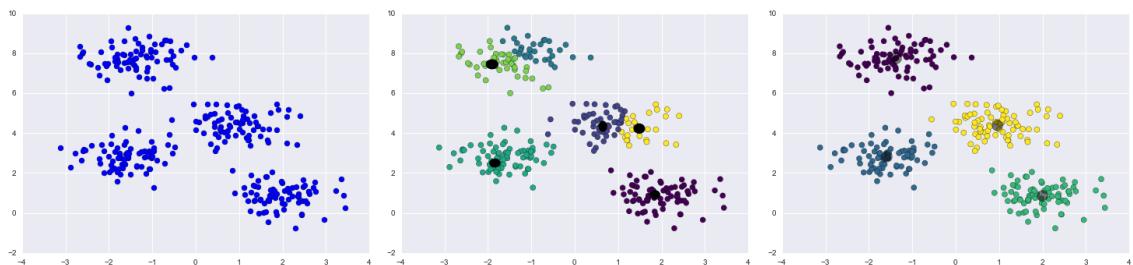
Deze thesis baseert zich ook voor een stuk op gemiddelde snelheden en tempo's. Hierdoor stellen we volgende bijkomende assumptie voor: een gebruiker mag niet stilstaan binnnenin de EPZ. Platformen zoals Strava hebben namelijk een ingebouwde functie die bij het uploaden van een activiteit tijdstippen 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 worden 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.

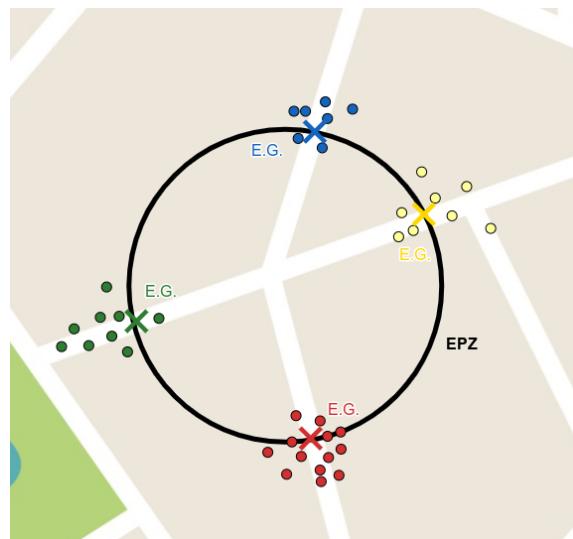
K-means clustering is een unsupervised machine learning techniek die veel wordt gebruikt bij het clusteren van data. Het is een iteratief proces waarbij het algoritme  $k$  clusters tracht te creëren waarbij de datapunten in elke cluster zo dicht mogelijk bij het gemiddelde van die cluster liggen [17].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

In de context van het identificeren van de EPZ zal het gebruikt worden om gps-punten te groeperen op basis van hun locaties. In plaats van te werken met een centraal punt, zal er gewerkt worden met een cirkelvormige zone. De euclidische afstand zal in dit geval berekent worden ten opzichte van de rand van deze cirkel. Bij elke iteratie zal een nieuwe cirkel worden beschouwd, en zullen de afstanden opnieuw worden berekent. Het proces wordt herhaald totdat een ietwat 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, 35].



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

Het algoritme zal na de identificatie van de EPZ ook nog nakijken of er niet meer dan één EPZ te vinden is. Er wordt onderzocht of punten die meegenomen zijn in de beschouwing van de huidige EPZ, toch niet horen bij een mogelijke andere EPZ van de user. Als controle wordt van elk eind- of beginpunt de Euclidische afstand berekent tot de rand van de bijhorende gevonden EPZ. Indien deze kleiner is dan de grootst mogelijke radius, dan wordt verondersteld 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:** Voorbeeld van entry gates gevonden door k-means clustering en de identificatie van een EPZ

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 erg kan verkleinen. Indien het algoritme één of meerdere EPZs vindt, dan zullen er enkel voorspellingen gebeuren in de regio binnenin. 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 stratenplan 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 is waar de gebruiker de EPZ kan betreden en/of verlaten. Deze ‘zones’ vormen zich dan ook rond wegen die de EPZ betreden. Op Figuur 3.4 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 middelpunten 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:

1. Selecteer een willekeurig niet-bezocht punt uit de 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 de 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 doet dit aan de hand van twee parameters: de maximale afstand tussen een punt en een ander punt in de dataset, wat in deze context vastgelegd is op 22.9 meter en een minimaal aantal punten dat de cluster moet bevatten, in dit geval één.

## 3.4 Bepalen nodige gegevens voor predictie

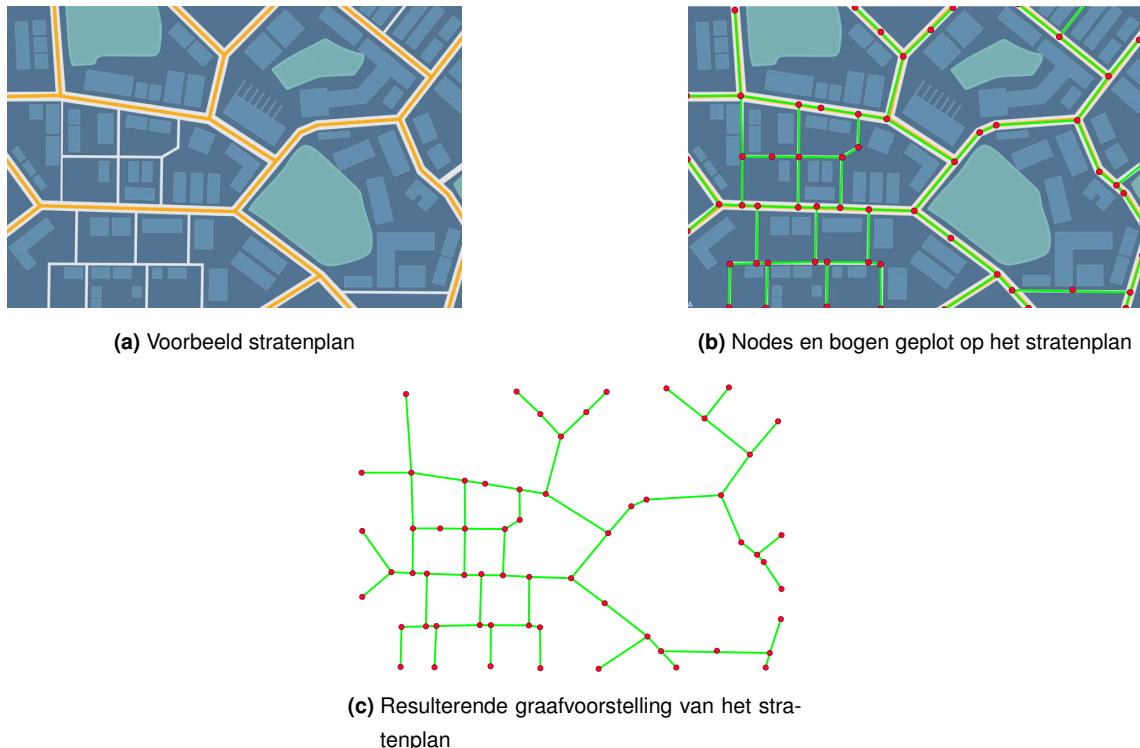
Na het bepalen van de EPZs voor de gebruiker wordt overgegaan tot het berekenen en achterhalen van de bijhorende gegevens die nodig zijn om de gevoelige locatie te voorspellen. Hiervoor wordt verder ingegaan op de methodiek in de paper Dhondt et al., maar er worden enkele gegevens op een andere manier 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 3.5 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, en deze kunnen dan worden verbonden. Indien geen EPZs geïdentificeerd zijn, dan wordt de omgeving die moet worden omgezet naar een graaf een stuk ruimer genomen. 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 node 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 densiteit, hoe meer nodes,

en dus ook hoe preciezer. Om voorspellingen te maken is wel een bepaalde precisie vereist, dus mag deze waarde niet te hoog zijn. Empirisch werd gekozen voor een waarde van  $3.0m$ .

**Figuur 3.5:** Voorbeeld van het genereren van een roadgraph



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>1</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

<sup>1</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. 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 [15].

het zichtbare eindpunt. Volgens één van de voorafgaand gemaakt assumptie vertrekt of eindigt de sporter in de EPZ. 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. En omgekeerd, indien hij 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>2</sup>, en de cumulatieve tijd tot dit punt<sup>3</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 hier besproken wordt twee belangrijke gegevens ter beschikking moeten zijn. Met name het straatnetwerk met de mogelijks gevuldte routes, wat werd besproken in Sectie 3.4.1, en de afstand die wordt afgelegd binnenin de EPZ. Deze afstand benoemen we ook als de *inner distance* (dit is niet te verwarren met de *inner distance 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 moet dit echter gebeuren met een tussenstap. 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 snelheid- en tijdsgegevens kan dit via basisformules gebeuren. Gebruik makend van het gemiddelde tempo kan de voorgaande formule worden omgevormd tot:

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

---

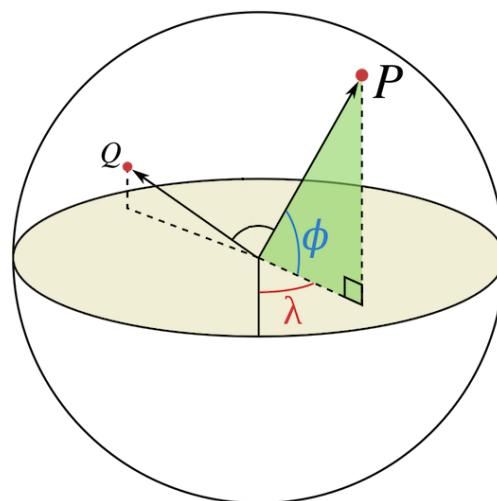
<sup>2</sup>De totale afstand afgelegd vanaf het begin van de activiteit tot en met het punt in kwestie.

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

In opzicht van het tweede scenario, waarbij alle afstandsgegevens verborgen zijn, ontbreekt nu ook de *outer distance*. We bepalen deze dan ook via de gps-coördinaten. Dit gebeurd door de som te nemen van de afstanden van alle opeenvolgende punten. Let wel dat we de afstand tussen twee gps-punten berekenen gebruik makend van de *haversine* formule. Formule 3.1 is een uitwerking van de formule. Deze berekent de afstand tussen twee punten op een bolvormig oppervlak, in dit geval de aarde. Om de afstand tussen twee punten op het aardoppervlak te berekenen, moeten de breedte- en lengtegraden van elk punt worden omgezet naar radialen. Vervolgens worden deze waarden ingevoerd in de formule, samen met de straal van de aarde ( $r$ ), meestal genomen als 6.371km. De formule berekent dan de haversine 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 te berekenen tussen de punten P & Q op Figuur 3.6 [23].

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 van 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.

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



**Figuur 3.6:** Haversine illustratie voor het berekenen van de afstand [37]

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 te voeren is de berekening van de totale inner distance voldoende. Maar bij het uitvoeren van

een *inner distance attack* zijn twee aparte 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 deze moeten worden gebundeld tot één locatie, voor alle activiteiten.

### 3.5.1 Filteren activiteiten

Voorafgaand aan het voorspellen van de locatie, is het belangrijk dat enkel voorspellingen gebeuren met activiteiten die een nuttige voorspelling kunnen voortbrengen. De andere activiteiten zouden enkel de accuraatheid van de voorspelling naar beneden halen. Dit gaat dan over activiteiten waarbij niet de kortste route binnenin de EPZ wordt gevolgd. 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 kan in zekere mate worden opgevangen door te stellen dat een activiteit enkel wordt gebruikt wanneer de nog af te leggen afstand binnenin de EPZ kleiner dan de maximaal mogelijk af te leggen weg. In de andere gevallen zal de activiteit worden gefilterd. 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 voor een stuk gevallen in rekening brengen waarbij de activiteit 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/eindpunt van de activiteit, gelegen op de rand van de EPZ.

	Node1	Node2	Node3	Node4	Node5
EG1	5.23	14.00	7.90	3.22	12.54
EG2	20.41	37.91	90.89	8.43	3.76
EG3	9.71	22.33	12.61	80.54	73.66
EG4	16.75	4.30	11.69	57.60	71.04
EG5	88.62	7.31	92.43	4.18	5.36

Figuur 3.7: Willekeurig voorbeeld van Distance Matrix met een voorspelling van 12,3m

Ook worden de zichtbare eind- en beginpunten van de activiteiten gecontroleerd naar compatibiliteit met de road graph. Alle eind- en beginpunten worden gesnapt op de roadgraph. Ze worden dus vervangen door de dichtstbijzijnde node op de graaf. Indien het verschil in afstand tussen de originele locatie en de gesnachte locatie te groot is, wordt de activiteit gefilterd. Dit wijst dan op een te grote afwijking tussen de activiteit en de road graph, of op inaccurate gps-data.

Als laatste wordt nog gekeken naar afwijkingen bij de E.G.'s. Indien bij 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 grote spreiding bij ten opzichte van de E.G., 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 berekent 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 de nodes op het 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 gevogelde route is dan een node, die mogelijk de gevoelige locatie kan voorstellen. In de praktijk kan dit mechanisme op een simpele manier worden toegepast door het gebruik van de vooraf berekende distance matrix. De berekende inner distance zal worden vergeleken met de afstanden in de distance matrix, en de nodes in de EPZ die het dichtst bij deze afstand liggen zullen worden geselecteerd als kandidaten voor de gevoelige locatie. Een willekeurig voorbeeld hiervan is te zien op Figuur 3.7, voor een afstand van 12.30m, voor EG1. Het resultaat in dit voorbeeld zal Node3 zijn. In dit voorbeeld wordt nu slechts één node voorspelt, maar in de praktijk kunnen dit er meerdere zijn. Indien dit proces herhaald wordt voor alle beschikbare activiteiten, met mogelijk verschillende Entry Gates, zullen er verschillende kandidaten worden gevonden. Ook zullen bepaalde nodes meer dan één keer worden voorspeld. In de volgende stap zullen alle predicties worden samengenomen in één voorspelling.

### 3.5.3 Meest voorspelde locatie

Een verzameling van nodes is nu bekomen, die normaliter ook de gevoelige locatie bevat. Om deze te bepalen, wordt regressie-analyse toegepast, aan de hand van de Least Absolute Deviations (LAD) methode. Het resultaat van deze regressie-analyse zal een gps-locatie zijn, die onze *eindvoorspelling* zal vormen.

De LAD methode 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 van de vergelijking staat afwijkingen tussen de waarden, de 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.

$$LAD : \min \sum_{i=1}^n |y_i - \hat{y}| \quad (3.2)$$

$$OLS : \min \sum_{i=1}^n (y_i - \hat{y})^2 \quad (3.3)$$

In de 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 het aantal voorspelde over het algemeen locaties beperkt is.

In de context van deze thesis kan Vergelijking 3.2 worden toegepast voor elke voorspelde locatie. Alle voorspelde locaties zullen worden beschouwd als mogelijke *eindvoorspelling*. Deze zal dan in de vergelijking  $\hat{y}$  voorstellen. De absolute waarde van het verschil tussen de eindvoorspelling en alle overige punten zal worden opgeteld. Er wordt gezocht naar het punt waarbij de som van de absolute waarde minimaal is, en dit is dan de resulterende predictie, de **eindvoorspelling**.

## **Hoofdstuk 4**

# **Gebruikte data en afwijkingen**

De aanval die beschreven werd in Hoofdstuk 3 testen we op groep aantal gebruikers met hun desbetreffende activiteiten, om zo zinvolle conclusies te trekken over de vatbaarheid van gebruikers op fitnessplatformen voor deze aanval. Maar het is dan ook essentieel dat een correcte dataset wordt gebruikt, en dat de eigenschappen die deze dataset heeft worden beschreven. We onderzoeken eigenschappen en mogelijke afwijkingen of onregelmatigheden opdat een gefundeerde conclusie kan worden gevormd, die eventueel bepaalde eigenschappen van de dataset mee in rekening brengt.

Aangezien deze thesis voor een stuk verder bouwt op het onderzoek van Dhondt et al., is het evident om verder te werken op deze dataset [6]. Deze dataset werd volledig zelf gescraped vanaf de officiële Application Programming Interface (API) 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 en voor alle activiteiten, 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 gecloaked<sup>2</sup> is, of reeds verwijdert is, dan zal deze worden overgeslagen en de volgende worden genomen. Voor elke activiteit die succesvol werd gevonden, wordt daarna de gebruiker beschouwd. Van alle bekomen gebruikers worden dan de rest van de activiteiten afgehaald en bijgehouden in één grote dataset. De gegevens worden ook geanonimiseerd opgeslagen, zodat de gebruikers niet meer kunnen worden geïdentificeerd. De dataset bevat dus geen namen of andere persoonlijke gegevens, enkel ID’s.

---

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

<sup>2</sup>Een gecloakede activiteit is een activiteit waar al een EPZ op is aangebracht. Aangezien deze thesis uncloaked activities nodig heeft 5.1.2 zijn ze dus niet bruikbaar. Men kan zien of een activiteit gecloaked is indien er een verschil is tussen de zichtbare afstand en de totale afstand.

In totaal werd een dataset van 4000 gebruikers verzameld. 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.

## 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 relatief een grote spreiding van activiteiten over de hele wereld, wat een goede basis is voor het testen van de aanval. Let wel, de dataset is met 101 gebruikers echter wel relatief klein, wat een vertekend beeld kan geven van de werkelijkheid.

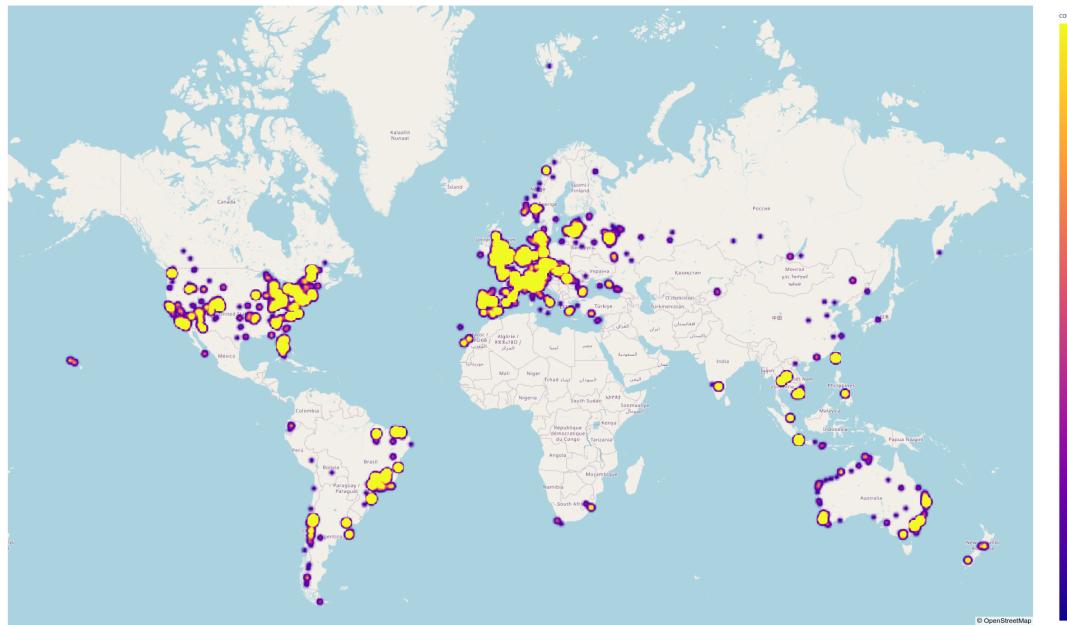
Op Tabel 4.1 zijn enkele globale statistieken met betrekking tot gebruikers en de bijhorende activiteiten van de dataset weergegeven. Er valt op dat de dataset per gebruiker toch meestal een groot aantal activiteiten ter beschikking zijn. De gemiddelde gebruiker bevat 411 activiteiten, de mediaan is 296. Volgens de inferentieaanval beschreven in Hoofdstuk 3 resulteert een gebruiker met meer activiteiten over het algemeen in een accuratere aanval. Op Figuur 4.2 is de CDF plot<sup>3</sup> te zien die het aantal activiteiten per gebruiker weergeeft. 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.

Waarde	Aantal
Total number of users	101
Total number of activities	41 554
Average # activiteiten per user	411
Median of # activities per user	296
Maximal # activities for single user	2946
Minimal # activities for single user	31

**Tabel 4.1** Overzicht van gebruikers en activiteiten

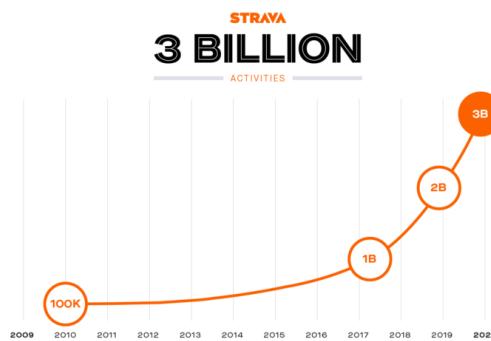
Let wel, alhoewel niet expliciet vermeld door Dhondt et al., is er een vermoeden dat er bewust gezocht werd naar gebruikers met een zo groot mogelijk aantal activiteiten per gebruiker. Dit is

<sup>3</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.

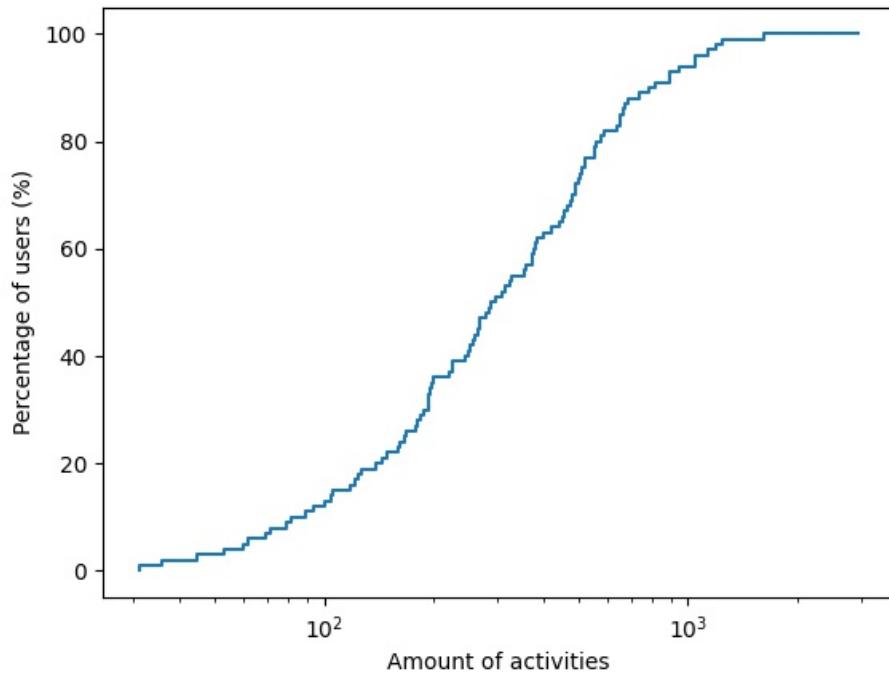


**Figuur 4.1:** Geografische spreiding van de activiteiten in de dataset

een logische keuze, aangezien de aanval een hogere kans op slagen heeft bij users die meer activiteiten hebben. Wanneer we dit vergelijken met cijfers uit een studie die Strava zelf voerde in 2020, is er toch een mismatch terug te vinden [32]. Het persbericht, waarvan Figuur 4.3 is overgenomen, stelt dat Strava in 2020 iets meer dan 50 miljoen gebruikers had, die samen in totaal drie miljard activiteiten op het platform hebben geplaatst. Indien we deze waarden omrekenen naar een gemiddelde, komen we uit op een ruwe geschatte 60 activiteiten per gebruiker ( $\frac{3 \cdot 10^9}{5 \cdot 10^7} = 60$ ). Dit is een stuk lager dan de gemiddelde 411 activiteiten per gebruiker in de dataset. De conclusies die dus getrokken worden uit deze steekproef mogen niet zomaar veralgemeend worden naar de volledige gebruikersbasis van Strava.



**Figuur 4.3:** Post op sociale media van Strava die de evolutie van het totaal aantal activiteiten weergeeft [32]



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

## 4.2 Mogelijke afwijkingen binnenin de dataset

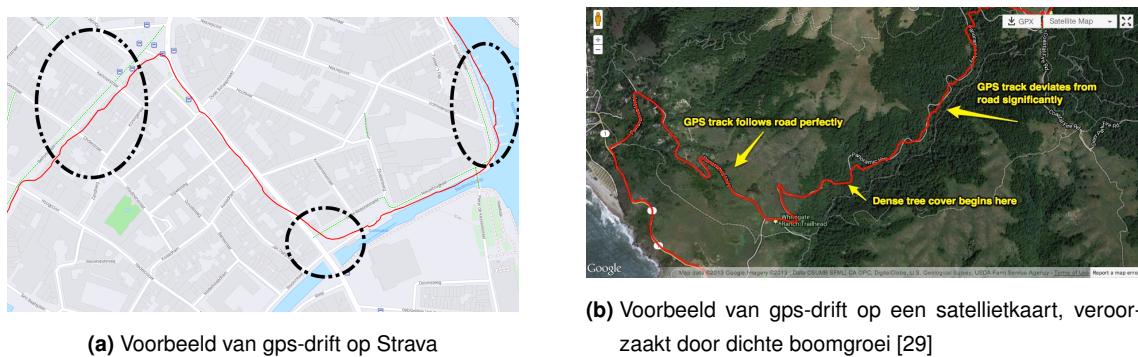
Doordat de dataset niet explicet werd gecheckt op onnauwkeurigheden en een willekeurige sample is, Doordat deze dataset niet door Strava zelf werd vrijgegeven, maar manueel afgehaald werd via scraping, is er een grote kans op activiteiten die afwijkingen of fouten vertonen. Zeker door het belang van gps-data, die een grote kans heeft op fouten, in deze studie is het belangrijk om de dataset te analyseren op deze 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 deze satellieten te bereiken, de locatie van een gebruiker kan bepalen [29]. Door de 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 frequentie waarmee locatie wordt bepaald, wat afhankelijk is van het gebruikte toestel kan meespelen.

### 4.2.1 Mogelijke fouten bij gps-data

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. Een voorbeeld vanuit het Strava-platform is terug te vinden op Figuur 4.4a. 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 dichtbevolkte omgevingen, en omgevingsfactoren zoals hoge bomen. Om dit tegen te gaan, zou eventueel de eerder beschreven methode van map-snapping kunnen worden toegepast.

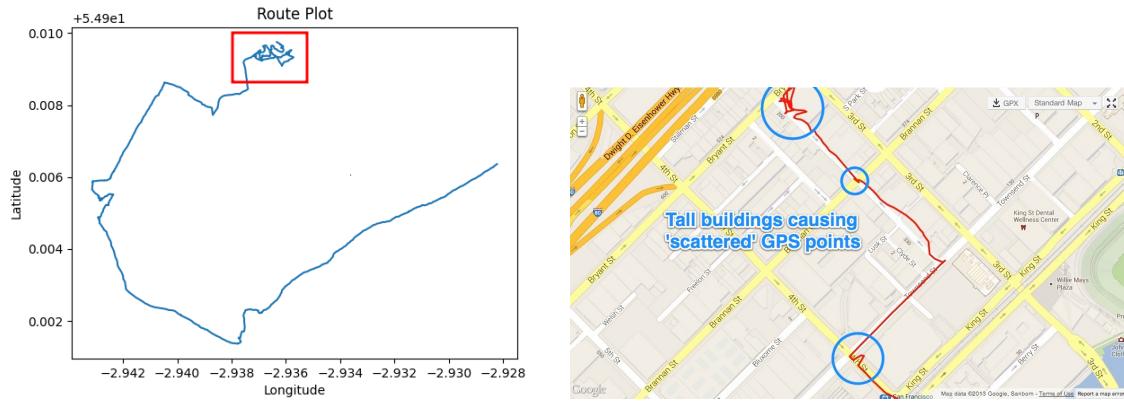


**Figuur 4.4:** Voorbeelden van gps-drift

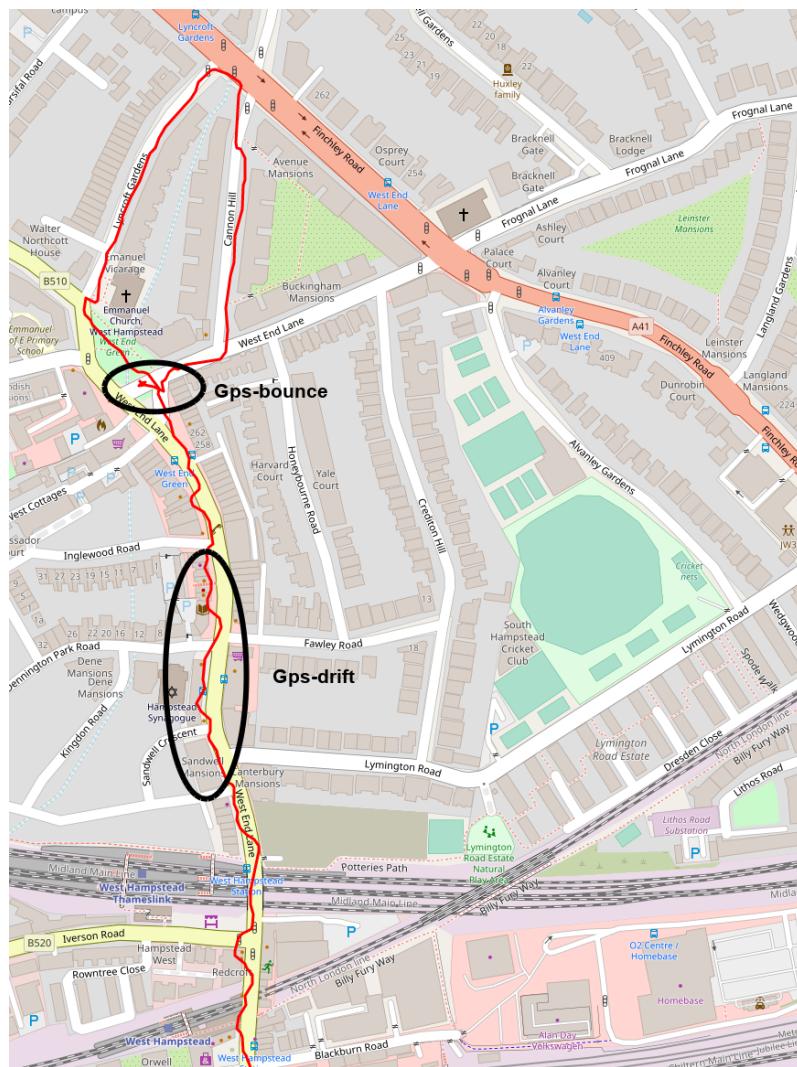
Gps-bouncing is een fenomeen veroorzaakt door hoge gebouwen. Het gps-signaal zal hierbij over en weer weerkaatsen tussen de gebouwen, op weg naar de satelliet. Hierdoor verondersteld het apparaat afgelegde afstand bij het berekenen van de locatie, door de extra vertraging. De uitkomst van het traject 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 4.5. Om dit fenomeen op zijn beurt tegen te gaan, is het best om smoothing te toe te passen bij het berekenen.

Er zijn ook voorbeelden waarbij beide fenomenen voorkomen, zoals te zien is op Figuur 4.6. Zoals te zien is op deze figuur, zal indien op een intuïtieve manier de afstand wordt berekend (door gewoon het afstandsverschil tussen twee opeenvolgende punten te nemen), een significant verschil zijn tussen de afstand die de gebruiker werkelijk afgelegd heeft, en de berekende afstand.

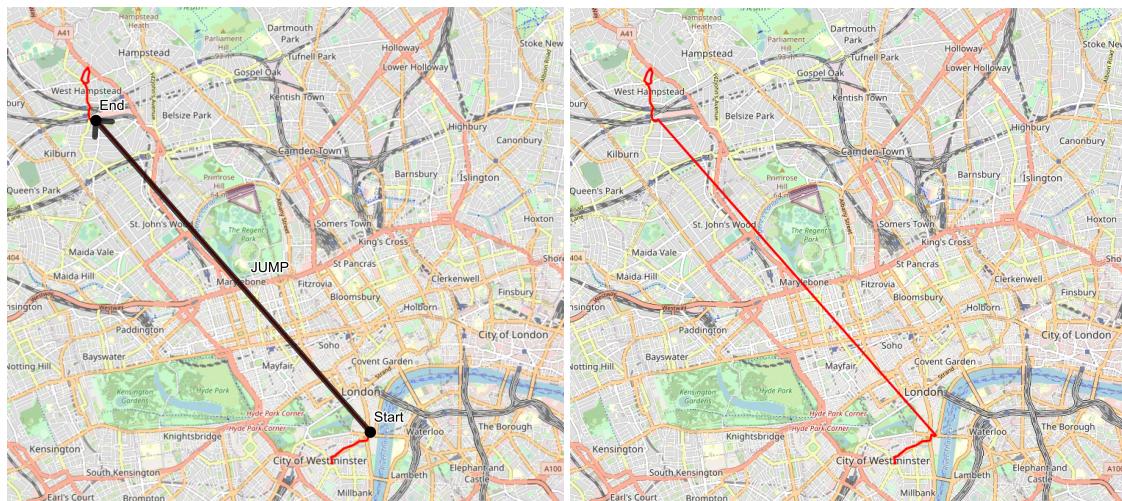
Een laatste fenomeen dat kan optreden is gps-signal loss. Hierbij gaat het signaal van de gebruiker verloren, en wordt pas op een later tijdstip terug een nieuw signaal verzonden, waardoor een sprong werd gemaakt. In dit geval zou map matching opnieuw een goede oplossing om dit te omzeilen. Een tweede 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



**Figuur 4.5:** Voorbeelden van gps-bounce [29]



**Figuur 4.6:** Voorbeeld van zowel gps-drift en gps-bounce uit de gebruikte dataset



**Figuur 4.7:** Voorbeeld van signal loss uit de gebruikte dataset

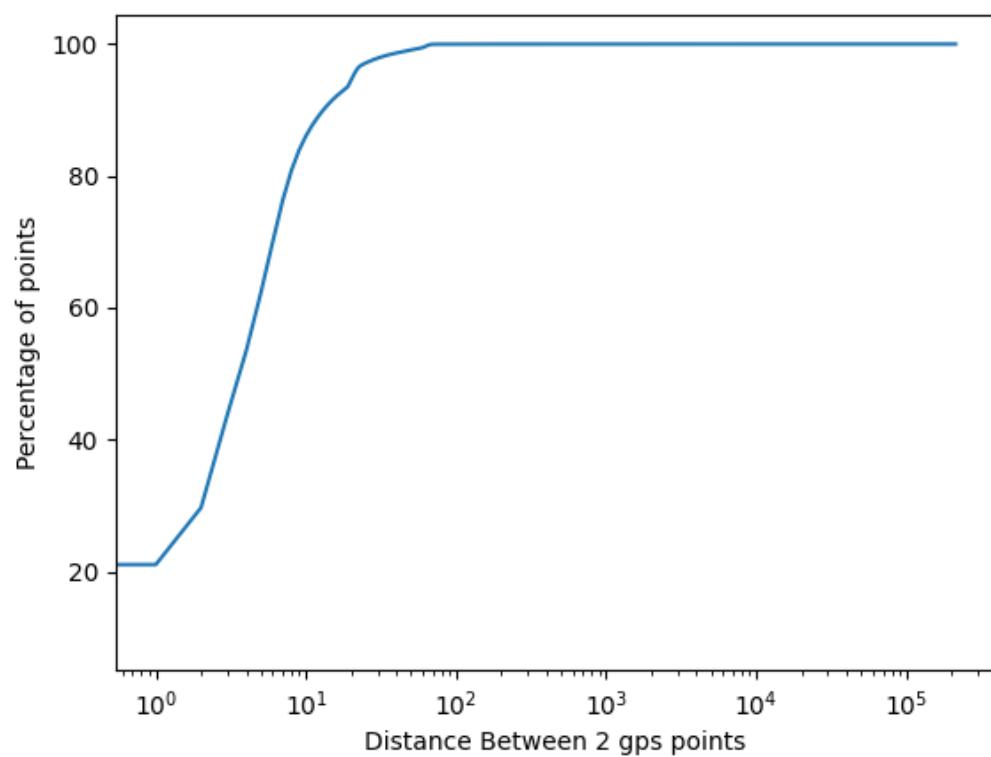
berekening van de afstand. In het geval van een pauze zal mapmatching geen oplossing zijn, maar zou deze ‘spongafstand’ best weggelaten worden in de berekeningen. Een voorbeeld hiervan is terug te vinden op Figuur 4.7.

#### 4.2.2 Gps-fouten in de gebruikte dataset

Allereerst onderzoeken we naar 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 4.8. De gemiddelde afstand tussen twee opeenvolgende locaties is  $6.41m$ , met een standaardafwijking van  $42.53m$ . Het gemiddelde is relatief laag, wat kan wijzen op accurate gegevens, maar de hoge standaardafwijking wijst op grote schommelingen. Op de grafiek en de tabel is te zien dat de meeste afstanden onder de  $20m$  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 afstand 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  $50m$  bedraagt, dan ligt  $0.9\%$  van de data boven deze drempel. Per activiteit zou dit dan resulteren op gemiddeld  $23$  afwijkende punten, wat zeker kan zorgen voor een significante afwijking op de resulterende afstand.

Om het aantal gps-afwijkingen in de dataset te bepalen die relevant zijn voor de aanval, wordt ook het verschil onderzocht tussen de berekende afstand afgelegd buiten de EPZ (het zichtbare traject) en de theoretisch afgelegde afstand buiten de EPZ, die af te lezen valt uit de dataset via de cumulatieve afstand<sup>4</sup>. Een eerste visualisatie is te zien op Figuur 4.9. De figuur illustreert de

<sup>4</sup>Er wordt gesproken van een theoretische waarde, maar deze is eigenlijk de berekende waarde volgens het platform.



**Figuur 4.8:** Verdeling van de afstanden tussen twee opeenvolgende gps-punten

Total number of gps-points	$1.070 \cdot 10^8$
distance between 2 gps points > 10m	13.91%
distance between 2 gps points > 20m	5.44%
distance between 2 gps points > 50m	$9.02 \cdot 10^{-1}\%$
distance between 2 gps points > 75m	$2.21 \cdot 10^{-2}\%$
distance between 2 gps points > 100m	$1.30 \cdot 10^{-2}\%$
distance between 2 gps points > 150m	$8.04 \cdot 10^{-3}\%$
distance between 2 gps points > 200m	$6.071 \cdot 10^{-3}\%$
distance between 2 gps points > 500m	$2.600 \cdot 10^{-3}\%$
distance between 2 gps points > 1000m	$1.345 \cdot 10^{-3}\%$
distance between 2 gps points > 2000m	$4.056 \cdot 10^{-4}\%$

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

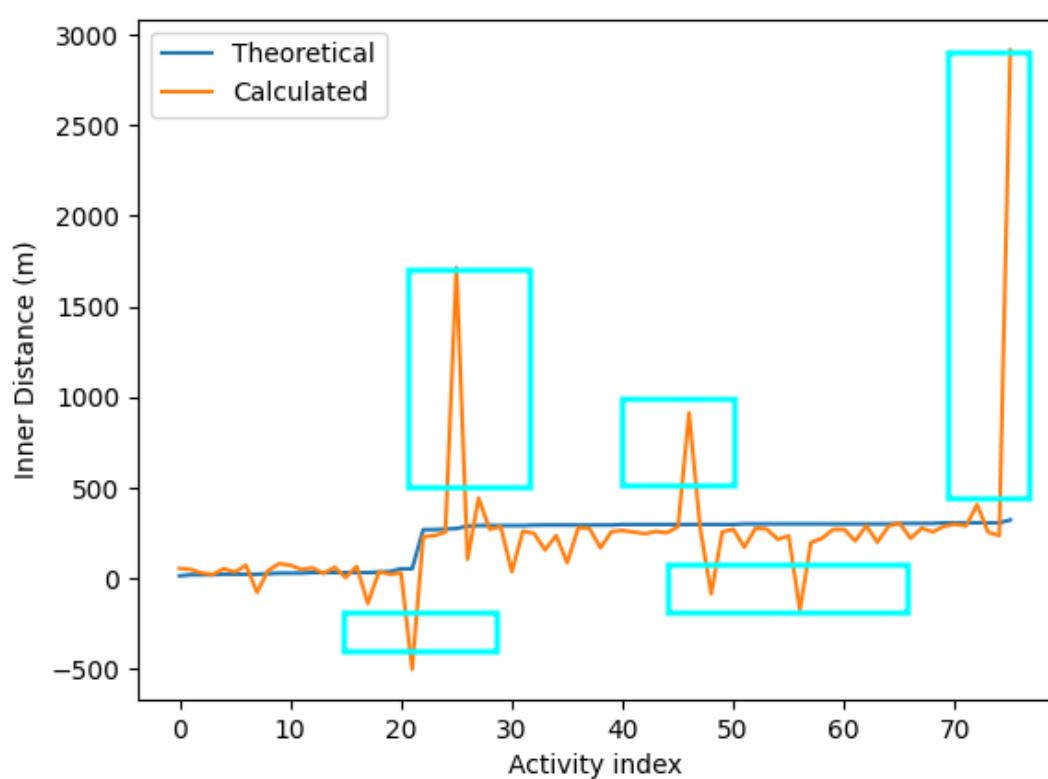
schommelingen tussen de handmatig berekende afstand en de theoretische afstand van één gebruiker. De pieken duiden op duidelijk sterk afwijkende berekende afstanden, en dus ook op grote gps-fouten. Maar ook de schommelingen die iets minder opvallend zijn duiden op grote inaccuraties tussen de berekende en theoretische afstanden. Voor de volledige dataset wordt gebruik gemaakt van CDF-plots. De resultaten worden weergegeven op Figuur 4.10. Figuur 4.10a bevat de verdeling voor alle activiteiten, gebruik makend van een logaritmische schaal. Figuur 4.10b toont 95% van de activiteiten met de kleinste verschillen, om een beter beeld te krijgen van de grootte van de meeste verschillen. Op de grafieken 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 relevant 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.

### 4.3 Technieken 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 fitness-platformen gebruik maken van technieken om de gps-data te verbeteren. De besproken technieken waren *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.

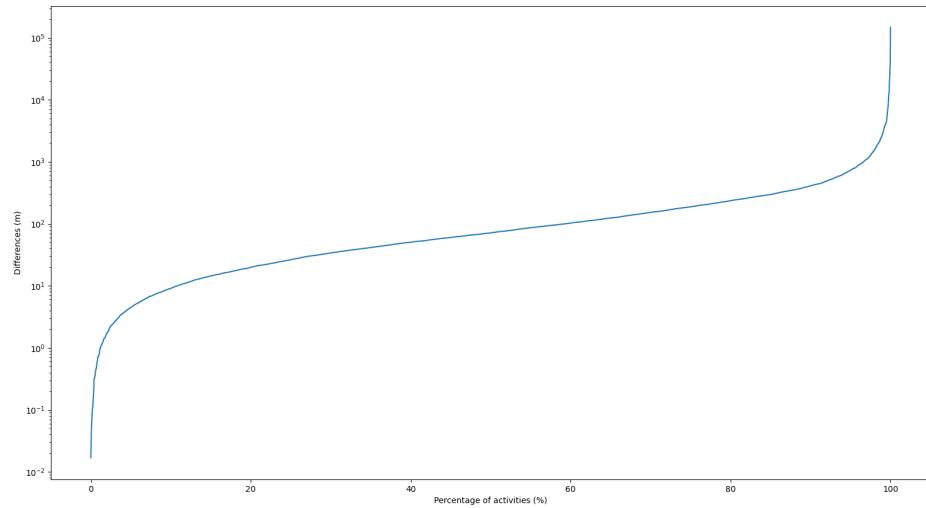
---

We beschouwen deze dus als referentie.

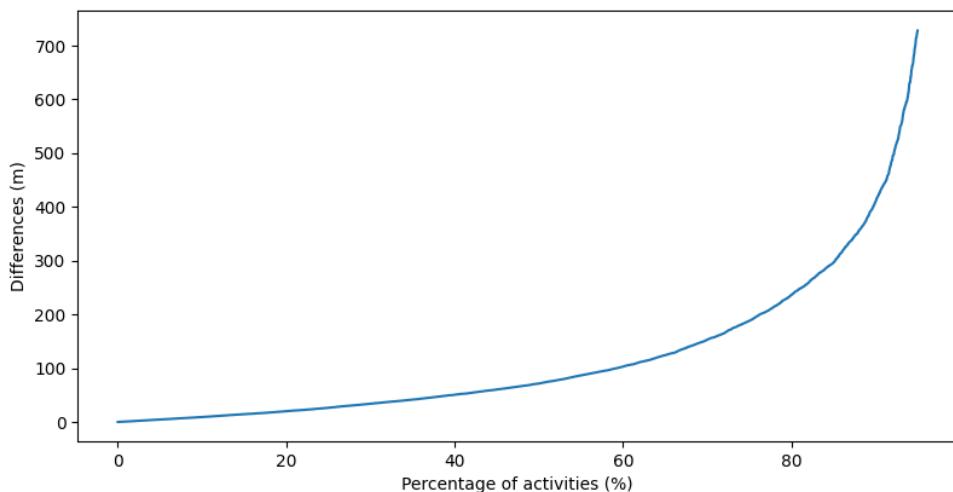


**Figuur 4.9:** Verschil tussen de berekende afstand en de theoretische afstand voor één gebruiker

Daarnaast passen we ook een 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 wel niet vanzelfsprekend, aangezien dit wel kan worden meegenomen door de berekeningen van de platformen. Stel dat wij een sprong van  $50m$  laten vallen, maar de platformen laten deze wel meetellen, dan zullen we de afwijking op het eindresultaat enkel maar verhogen. Het drempel voor de filtering moet dus hoog genoeg zijn om deze voorvallen te vermijden. In dit onderzoek kozen we voor een drempelwaarde van  $200m$ .



(a) 100% van de activiteiten (logaritmische schaal)



(b) 95% van de activiteiten

**Figuur 4.10:** Verdeling van het verschil tussen de berekende afstand en de theoretische afstand buiten de EPZ

# **Hoofdstuk 5**

## **Resultaten en Evaluatie**

De aanval werd tot hertoe al volledig beschreven en toegelicht, en de dataset die gebruikt wordt om de aanval werd ook al besproken. Dit hoofdstuk bespreekt hoe de evaluatie wordt aangepakt, wat de bekomen resultaten zijn en wat ze betekenen.

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

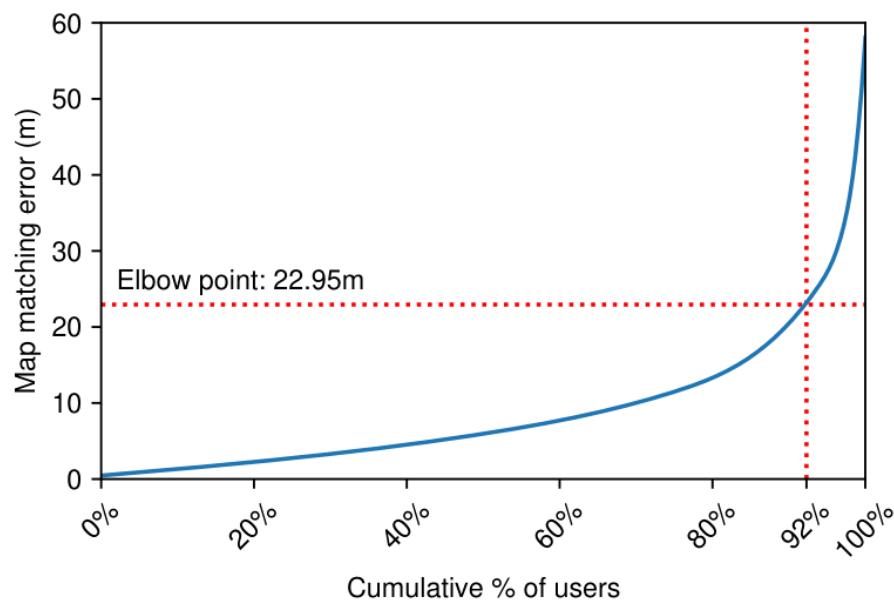
Het doel van de aanval is om een locatie te voorspellen waar een gebruiker bij zijn activiteiten vertrekt of aankomt, ondanks het feit dat deze locatie wordt verborgen door het gebruik van een EPZ. Maar indien we dit zouden uitstellen op activiteiten die al een EPZ bevatten zouden we de bekomen resultaten niet kunnen verifiëren. Daarom zullen we de aanval uitstellen 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 dat tracht verborgen te worden. De grondwaarheid is dus de locatie die we trachten te achterhalen. We bepalen deze locatie door alle activiteiten van een gebruiker te overlopen, en indien 15 of meer begin- of eindpunten binnen een straal van 50 meter liggen, dan berekenen we hiervan het gemiddelde en wordt dit gemiddelde gemapt op de roadgraph. Dit

punt gematcht op de roadgraph stelt dan een grondwaarheid voor van deze gebruiker. Hassan et al. stelden dat 50 meter vergelijkbaar is met een breedte van een gemiddeld perceel, en dat dit dus een goede benadering is van de grondwaarheid [36, 35]. 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 deze afwijking in een CDF te plotten, en op zoek te gaan naar het elleboogpunt [6]. Het elbow point 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. Deze drempelwaarde is toepasselijk voor 92% van de gebruikers.



**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, om zo de evaluatieproces te versimpelen. 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 de aangebrachte EPZ op een realistische manier wordt aangebracht, op een manier die de werkelijkheid weerspiegelt.

Sectie 2.2 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 aanval 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 richting, en kiezen we dus willekeurig. 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.

BETROUWBAARHEIDSINTERVALLEN => VRAGEN THOMAS

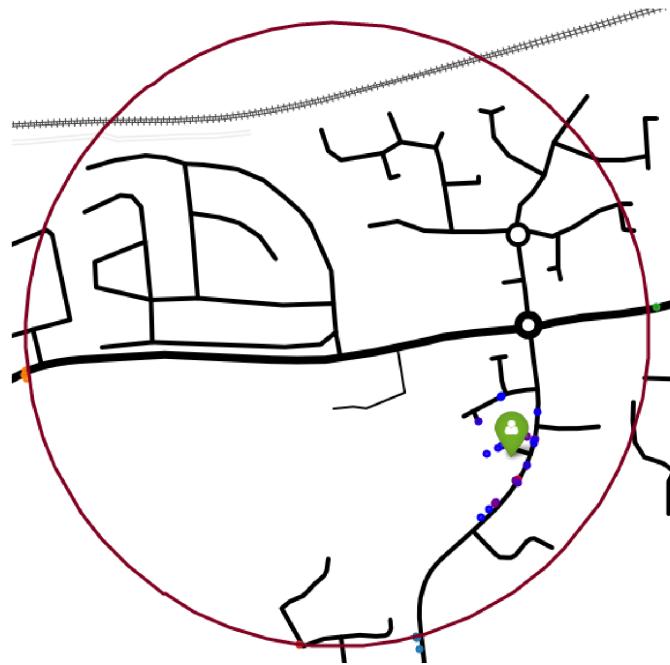
dit is hoe ik het interpreteer

### 5.1.3 Bootstrapping

Bij het uittesten van de aanval wordt de aanval niet zomaar gedraaid op alle activiteiten per gebruiker. Dit zou vertekende resultaten kunnen geven. Per gebruiker wordt een betrouwbaarheidsinterval berekent via bootstrapping [6, 35]. Per gebruiker wordt de set met manueel verwerkte activiteiten 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 bepaalde activiteiten helemaal niet bevatten. Dit gebeurt 1000 keer, en er worden dus 1000 verschillende sets gemaakt. Voor elke set zal dan 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 [35]. Op de figuur zijn de individuele voorspelling aangeduid met een blauwe tot paarse kleur, afhankelijk van hoe frequent ze voorspeld werden (hoe meer naar de paarse of rode kleur ligt, hoe frequenter de node voorspeld

<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 [36].

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 marker.



**Figuur 5.2:** Voorbeeld van een distributie van voorspellingen bepaald door het bootstrapalgoritme [35]

#### 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.

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. De succes rate is dus het percentage van de aanvallen waar de correcte locatie zich binnen deze straal 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, 35]. 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 succes rate kan hoog zijn, maar de correctheid kan nog steeds hoog zijn. 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 *Accuracy* definiëren we als de breedte van het betrouwbaarheidsinterval [6, 35]. Met de breedte van het betrouwbaarheidsinterval 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 bij een aanval [6, 35]. De mogelijke eindlocaties voor de aanval zijn simpelweg alle nodes in de graafvoorstelling, eventueel begrensd door de EPZ. Degene na de aanval zijn al degene die voorspelt worden. De reduction is dus een percentage die het verschil tussen de twee sets aangeeft. Hoe hoger de reduction, 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.

De *Uncertainty Region* is de som van oppervlaktes van de unie<sup>2</sup> van de onzekerheidsregio’s rond de voorspelde nodes [6, 35]. 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 elk punt die op deze node gemapt is een mogelijkheid bestaan dat deze eigenlijk ergens anders in de een zone van drie meter rond de node ligt. Hoe groter deze waarde, hoe groter de onzekerheid van de aanval, aangezien dit wijst op weinig overlap.

De laatste evaluatiefactor is de *Degree of Anonymity* [6, 35]. 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  $p$  voorspelt wordt. Dit zal voor vele nodes gelijk zijn aan 0. De maximale entropie komt voor indien elke node exactly evenveel voorspelt wordt, en is dus gelijk aan  $\frac{1}{\#nodes}$

## 5.2 Resultaten

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
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]

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

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

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

**Tabel 5.4** Aanval op basis van gesmoothed gps-data en snelheid

Radius (m)	Smoothing Window (m)	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

Radius (m)	Smoothing Window (n)	Success Rate (%)	Correctness (m)	Accuracy	Reduction (%)	Uncertainty Region ( $m^2$ )	Certainty	Spatial Certainty	Degree of Anonymity (%)
200	5	53.33	88.56	28	77.23	613.0	2.45	1.28	37.86
400	5	43.48	140.37	38	90.07	869.30	2.69	1.56	34.53
600	5	44.14	176.17	34	94.33	772.82	2.76	1.78	31.28
800	5	40.86	207.75	44	96.54	1047.14	2.95	1.86	31.82
1000	5	33.01	279.75	34	97.20	898.48	2.67	2.00	28.57
1200	5	30.10	345.21	43	97.41	1021.74	2.86	2.07	28.92
1400	5	23.66	402.62	33	98.20	788.23	2.68	1.91	26.67

**Tabel 5.5** Attack result with different smoothing window sizes

## **Hoofdstuk 6**

## **Conclusies**

# 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>. (Accessed on 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://statomic.github.io/statistiekCursusNotas/index.html>. (Accessed on 05/14/2023).
- [4] Croft, J. (2015). Snapping gps tracks to roads. <https://www.jamesrcroft.com/2015/06/snapping-gps-tracks-to-roads/>. (Accessed on 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>. (Accessed on 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. [osf.io/3m5ut](https://osf.io/3m5ut).
- [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>. (Accessed on 04/18/2023).
- [8] Early, J. (2020). Smoothing and interpolating noisy gps data. <https://jeffreyearly.com/smoothing-and-interpolating-noisy-gps-data/>. (Accessed on 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>. (Accessed on 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>. (Accessed on 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>. (Accessed on 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>. (Accessed on 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>. (Accessed on 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>. (Accessed on 05/01/2023).
- [22] Seiler, K. M. (2022). Haul road mapping from gps traces.
- [23] Sheppard, W. and Soule, C. (1922). *Practical Navigation*. World Technical Institute.
- [24] Strava, I. (2021a). Strava-privacybeleid. <https://www.strava.com/legal/privacy>. (Accessed on 02/20/2023).

- [25] Strava, I. (2021b). Strava's year in sport 2021 charts trajectory of ongoing sports boom. <https://blog.strava.com/nl/press/yis2021/>. (Accessed on 02/26/2023).
- [26] 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>. (Accessed on 02/26/2023).
- [27] 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->. (Accessed on 05/14/2023).
- [28] Strava, I. (2023a). Activity privacy controls – strava support. <https://support.strava.com/hc/en-us/articles/216919377-Activity-Privacy-Controls>. (Accessed on 02/27/2023).
- [29] Strava, I. (2023b). Bad gps data – strava support. <https://support.strava.com/hc/en-us/articles/216917707-Bad-GPS-Data>. (Accessed on 03/01/2023).
- [30] Strava, I. (2023c). How distance is calculated – strava support. <https://support.strava.com/hc/en-us/articles/216919487-How-Distance-is-Calculated>. (Accessed on 03/01/2023).
- [31] Strava, I. (2023d). Strava global heatmap. <https://www.strava.com/heatmap#7.00/-120.90000/38.36000/hot/all>. (Accessed on 05/17/2023).
- [32] Strava, I. (February). 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/>. (Accessed on 05/14/2023).
- [33] TechTarget (2011). What is heat map (heatmap)? — definition from techtarget. <https://www.techtarget.com/searchbusinessanalytics/definition/heat-map>. (Accessed on 05/17/2023).
- [34] 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/>. (Accessed on 02/20/2023).
- [35] 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.
- [36] Wajih Ul Hassan, Saad Hussain, A. B. (2018). Analysis of privacy protections in fitness tracking social networks -or- you can run, but can you hide?
- [37] Zheng, J. (2019). Distance application in data science - jingwen zheng. <https://jingwen-z.github.io/distance-application-in-data-science/>. (Accessed on 05/07/2023).

## **Bijlage A**

# **Uitleg over de appendices**

Bijlagen worden bij voorkeur enkel elektronisch ter beschikking gesteld. Indien essentieel kunnen in overleg met de promotor bijlagen in de scriptie opgenomen worden of als apart boekdeel voorzien worden.

Er wordt wel steeds een lijst met vermelding van alle bijlagen opgenomen in de scriptie. Bijlagen worden genummerd het een drukletter A, B, C,...

Voorbeelden van bijlagen:

Bijlage A: Detailtekeningen van de proefopstelling

Bijlage B: Meetgegevens (op USB)

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)

