Turning It Off: context-driven prevention of passive WiFi-tracking

Het tegengaan van passive WiFi-tracking met smartphone-sensoren en AI

Open Universiteit,
Valkenburgerweg 177, 6419 AT Heerlen,
Faculteit Management, Science & Technology,
Masterscriptie Software Engineering,

28 april 2019

Ing. Aksel Harrewijn
Studentnummer: 851381635
Afstudeerbegeleider: Dr. ir. Hugo Jonker
Aanvullende examinatoren: Dr. Arjen Hommersom & Prof. Dr. Marko van Eekelen

Voorwoord

Doordat het technisch mogelijk is mensen te volgen, door handig van WiFi gebruik te maken, heeft dit geleid tot de scriptie “Turning It Off: context-driven prevention of passive WiFi-tracking”, die nu voor u ligt. Zo is het zaak gebleken nieuwe methoden en technieken te onderzoeken, om de strijd tegen WiFi-tracking aan te gaan. Interesse in smartphones, de sensoren die we daarop kunnen vinden en belangstelling voor machine-learning leiden tot nieuwe ideeën, die het mogelijk kunnen maken deze strijd te winnen.

Zo komen tijdens het onderzoek verschillende zaken aan bod, zoals onder meer WiFi-tracking-netwerken, kunstmatige intelligentie, en bij welke sensoren deze gebaat zijn op een smartphone. Doordat locatiedata steeds makkelijker clandestien uitwisselbaar is, hoop ik dat onze privacy beter geborgd kan worden met het verrichten van dit onderzoek.

Dit afstudeerwerk is geschreven voor een zo breed mogelijk publiek, terwijl op de meer relevante punten meer oefening van de lezer mag worden verwacht. Zo zal voor het grotere publiek met name hoofdstuk vier, over machine-learning, regressie en kernel-functies, lastiger te volgen zijn. Uiteindelijk hoop ik dat iedereen één of meerdere hoofdstukken van zijn gading kan vinden, die aanspreken, waardoor dit kan leiden tot een wereld die zich beter kan verdedigen tegen clandestiene schendingen van de privacy.

Voor het onderzoek is er een cursus over machine-learning van Andrew Ng, professor aan de Stanford University, gevolgd, om het nodige inzicht in machine-learning te verkrijgen. Deze inzichten hebben met veel plezier geleid tot de bevindingen, waarover u hopelijk zult gaan lezen. Hier moest veel werk voor worden verzet. Zo gaat mijn dank uit naar degenen die me daartoe hebben ondersteund.

In het bijzonder wil ik mijn bejaarde moeder bedanken, die geregeld voor het eten en de inwendige mens zorgde, waardoor het studeren mogelijk was, tijdens het drukke bestaan. Ook mijn afstudeerbegeleider, dr. ir. Hugo Jonker, wil ik graag bedanken, met wie ik geregeld discussieerde, terwijl de meter langer doorliep dan dat eigenlijk de bedoeling was. Vaak zonder dat we daar erg in hadden, soms terwijl de universiteit dicht ging.

Ook mijn nieuwe broodheer, Tradinco BV, "The Sensor Solution and Calibration Company", bij wie ik recent in dienst ging als Software Engineer IT & Elektronisch, wil ik graag bedanken voor de morele ondersteuning. Verder wil ik graag een goede vriend, Jeroen Noppen, bedanken, bij wie ik in huis experimenten mocht verrichten voor één van de onderzoekssituaties uit dit onafhankelijk onderzoek.

Ik wens u veel leesplezier toe.

Met vriendelijke groet,

Ing. Aksel Harrewijn

Rotterdam, 10 april 2019

Samenvatting

Om verschillende achtergronden te begrijpen dienen we methoden en technieken, om WiFi-tracking tegen te gaan, volgens een bepaald abstractieniveau inzichtelijk te maken. Dit doen we in de inleiding, waar de onderzoeksvraag en de onderzoekshypothese geformuleerd worden. Zo behandelen we in de inleiding tevens de tekortkomingen van veel gehanteerde technische maatregelen om WiFi-tracking tegen te gaan. Daardoor wordt het duidelijk dat aanvullend onderzoek, waaronder dit afstudeerwerk, wenselijk is.

In hoofdstuk twee vinden we een bespreking van eerder uitgevoerde relevante onderzoeken. Dit verschaft ons inzicht in de verschillende soorten WiFi-tracking-netwerken, active en passive WiFi-tracking-netwerken en welke onderverdelingen we daarin vinden. In dit hoofdstuk bespreken we bekende onderzoeken en mogelijkheden, waarmee het mogelijk is WiFi-tracking tegen te gaan of onmogelijk te maken. Voor dit laatste dient echter de mondiale IEEE 802.11-standaard aangepast te worden, om dit op een bruikbare en effectieve wijze mogelijk te maken, waardoor dit praktisch een onhaalbare zaak lijkt. Doordat we met dit exploratieve onderzoek andere tegenmaatregelen voorstellen ontleent dit onderzoek daaraan zijn waarde. Zo leggen we uit dat contextclassificatie kan bijdragen tot het tegengaan van WiFi-tracking, op een bruikbare en effectieve wijze, op basis van smartphone-sensor-input voor een software-toepassing.

Gedurende het onderzoek kiezen we ervoor om van een machine-learning-toepassing gebruik te maken, die geen gebruik maakt van externe systemen, waardoor diverse security-issues voor een belangrijk deel gedeeltelijk worden opgelost. Zo zijn dit security-issues die onze privacy in gevaar brengen en blijvend onze aandacht verdienen. In hoofdstuk drie werken we daarom naar een keuze uit de vele machine-learning-methodieken, voor de implementatie op een moderne Android-smartphone. Dit leidt naar hoofdstuk vier, om een keuze te maken uit lineaire en logistische regressie.

Hoofdstuk vijf gaat over de traceability en de connectivity, als maten van de bruikbaarheid en de effectiviteit, waarlangs een experimentele toepassing afgemeten wordt, voor dit exploratieve onderzoek. Dit doen we door de traceability en de connectivity van een machine-learning-toepassing te vergelijken met die van een passive-scanning-polling-app, in hoofdstuk negen.

In hoofdstuk zes wordt onderzocht welke smartphone-sensoren het meest geschikt zijn voor het beoogde doeleinde en hoe diverse sensormeetwaarden worden verwerkt. In hoofdstuk zeven geven we een korte samenvatting over de architectuur van de experimentele machine-learning-toepassing van enkele pagina’s, waarna we uitgebreid stil staan bij de exploratieve onderzoeksmethoden, in hoofdstuk acht. De resultaten, die daarmee gevonden worden, worden vervolgens gepresenteerd in hoofdstuk negen. Dit leidt tot de samengevatte conclusie over het gehele onderzoek, hoofdstuk tien.

Tot slot vinden we alle benodigde bijlagen en de programmatuur van de experimentele onderzoeks-toepassingen, die van een machine-learning-toepassing en een passive-scanning-polling-app.

Inhoud

1. Inleiding

Ongeveer driekwart van alle Nederlanders bezit tegenwoordig een smartphone. Daarmee kan men dag en nacht automatisch gebruik maken van WiFi-netwerken. Daardoor is het mogelijk de bewegingspatronen van smartphone-gebruikers in kaart te brengen. Dit komt doordat een smartphone-MAC-adres standaard constant uitgezonden wordt om een netwerkverbinding te maken. Zo wordt een MAC-adres constant uitgezonden voordat een access-point met het ontvangen MAC-adres een verbinding bewerkstelligd. Dit is bij wireless tracking-netwerken niet het geval: Het ontvangen MAC-adres wordt binnen een wireless tracking-netwerk alleen gebruikt voor logging en analyse, maar niet om een verbinding tot stand te brengen met een gebruiker. Daardoor zijn bedrijven, zoals Bluetrace, in staat bewegingspatronen van smartphones en hun gebruikers in kaart te brengen, zonder dat deze gebruikers zich daarvan bewust zijn.

Wanneer we in staat zijn een gebruiker in real-time te tracken kan dit een serieuze inbreuk op de privacy tot gevolg hebben. We kunnen hier denken aan woninginbraken, stalking, afpersing, aanwezigheidscontroles, ongewilde advertenties, het analyseren van het winkelgedrag, het volgen van groepen mensen en zelfs fysiek geweld, zoals booby-trapping.

Een Nederlands rapport van het College bescherming persoonsgegevens (CBP) [1] toont aan dat de naleving van de privacyregelgeving bij WiFi-tracking in belangrijke mate in het geding is. Zo maken niet alleen winkeliers, maar ook lokale overheden zich schuldig aan het ongeoorloofd verzamelen van persoonsgegevens en/of de verwerking en opslag daarvan, door gebruik te maken van WiFi-tracking. Een meer recent voorbeeld is de gemeente Eindhoven waar de bezoekersstromen in het uitgaansgebied met behulp van WiFi-tracking worden opgeslagen en geanalyseerd [30].

Zo rijst ook bij het CBP de vraag wat hiertegen gedaan kan worden en is het doel van dit onderzoek te bezien wat we tegen WiFi-tracking kunnen doen. We kunnen hier denken aan voor de hand liggende oplossingen, zoals:

1. Het uitzetten van WiFi;
2. Een power-friendly-mode-toepassing;
3. Een passive-scanning-polling-app;
4. Het wisselen van WiFi-MAC-adressen;
5. Het tegengaan van fingerprinting, door het aanpassen van firmware en/of de WiFi-drivers;
6. Het gebruik van mobiele 3G/4G-netwerken, in plaats van WiFi.

Het nadeel van deze voor de hand liggende oplossingen is dat ze geen van alle voldoen met betrekking tot de bruikbaarheid en de effectiviteit, waardoor verder onderzoek noodzakelijk is; hetgeen puntsgewijs toegelicht wordt. Zo is het altijd uitgeschakeld hebben van een WiFi-module zeer effectief om WiFi-tracking tegen te gaan, maar slecht bruikbaar doordat er daardoor geen WiFi-netwerkverbinding kan worden gemaakt.

Een power-friendly-mode-toepassing is één van de mogelijke oplossingen om WiFi-tracking in een bepaalde mate tegen te gaan. Deze staat ook wel bekend als een power-saving-mode-toepassing om het energieverbruik te minimaliseren. Zo'n oplossing zet minimaal de WiFi-verbindingsmodule uit op het moment dat een gebruiker inactief is. Doordat een power-friendly-mode-toepassing een verbindingsmodule binnen een vertrouwde omgeving uit kan zetten zal deze toepassing de bereikbaarheid in de weg staan. Dit is bijvoorbeeld het geval voor een Skype-toepassing of een app die aan de laatste nieuwsgaring doet in real-time. Een power-friendly-mode-toepassing zal ertoe leiden dat men op de inactieve momenten niet via WiFi bereikbaar is binnen een vertrouwde

omgeving, terwijl men op actieve momenten binnen een niet-vertrouwde omgeving vatbaar is voor WiFi-tracking. Dit staat de bruikbaarheid en de effectiviteit van deze oplossing in de weg.

Een andere oplossing is een passive-scanning-polling-app. Een dergelijke oplossing wacht een request van een WiFi-access-point af, voordat een proces, die tot een dataverbinding kan leiden, gestart wordt. Dit in tegenstelling tot het standaard active-scanning-WiFi-gebruik, waarbij dit andersom geldt; hetgeen het risico op tracking vergroot.

Het nadeel van een passive-scanning-polling-app is dat een gebruiker in mindere mate toch vatbaar blijft voor WiFi-tracking en Beacon Replay Attacks [2]. Dit geldt met name wanneer een gebruiker gebruik maakt van een WiFi-netwerk die opgebouwd is uit meerdere access-points die dezelfde SSID's uitzenden. Dit kan bijvoorbeeld binnen grote gebouwen het geval zijn. Dergelijke systemen kunnen niet alleen gevoelig zijn voor aanvallers. Dergelijke systemen kunnen gemakkelijk misbruikt worden om bijvoorbeeld het personeel te volgen [23]. Andere nadelen van een dergelijke app zijn het energieverbruik en het verbinden met verborgen WiFi-access-points die geen beacon-requests uitzenden. Dit laatste is met een zuivere passive-scanning-polling-app onmogelijk. In een bepaalde mate zitten deze zaken de bruikbaarheid en de effectiviteit van deze oplossing in de weg.

Een andere mogelijkheid om WiFi-tracking tegen te gaan is het regelmatig vervangen van WiFi-MAC-adressen op smartphones. Daarmee lijkt de inbreuk op de privacy van de gebruiker in eerste aanleg deels te worden opgelost, maar niet van het publiek, bijvoorbeeld om de bezoekersstromen bij een winkelschap in kaart te brengen. Mathy Vanhoef et al. [17] beschreef daarentegen hoe het originele MAC-adres van een smartphone door een tracking-systeem achterhaald kan worden, wanneer deze oplossing wordt gebruikt. Tevens beschreef hij hoe requests gefingerprint kunnen worden, waardoor de noodzaak van een smartphone-MAC-adres voor een WiFi-tracking-systeem wordt ondermijnd. Door deze zaken zal het wisselen van MAC-adressen minder effectief zijn om WiFi-tracking tegen te gaan, dan dat vaak wordt gehoopt.

Inmiddels zijn er vele artikelen verschenen over het fingerprinten van smartphones en het tegengaan daarvan. Ondanks dit gegeven staan de methoden en technieken, die binnen de genormaliseerde WiFi-netwerken toegepast worden, een waterdichte oplossing in de weg. Volgens Mathy Vanhoef et al. [17] kan fingerprinting op een effectieve worden tegengegaan door de WiFi-hardware aan te passen en door de firmware en/of de drivers op smartphones aan te passen. Dit impliceert dat de medewerking van smartphone-fabrikanten gewenst is om deze oplossingsrichting te doen slagen, wanneer er geen andere oplossingsrichting geformuleerd kan worden om de probleemstelling te tackelen. Zelfs wanneer men in staat is om een dergelijke oplossing te realiseren kan dit leiden tot een wapenwedloop met de fabrikanten van tracking-systemen en wireless tracking-access-points. Daardoor zal deze oplossingsrichting minder aantrekkelijk zijn.

In plaats van WiFi te gebruiken is het ook mogelijk mobiele 3G/4G-netwerken te gebruiken. Het gebruik van 3G/4G-netwerken kent echter zijn eigen security -en privacy-issues. Naast tracking kunnen we onder meer denken aan Denial of Service en eavesdropping [4]. Bovendien is het gebruik van dergelijke netwerken vaak een stuk duurder dan het gebruik van WiFi.

Met betrekking tot WiFi-tracking is het gebruik van WiFi risicovol, doordat niet alleen routers als een wireless tracking-acces-point clandestien geconfigureerd kunnen worden, maar ook laptops en smartphones met behulp van de juiste software. Dit is software die mogelijk door een aanvaller wordt geïnstalleerd. Om dit mogelijk te maken moet deze apparatuur de monitor-mode ondersteunen uit de IEEE 802.11-standaard (2012) [14, 19]; hetgeen in zeer veel gevallen het geval is. Verder worden er ook wireless WiFi-tracking-acces-points door hardware-fabrikanten aangeboden.

Een voorbeeld zijn die van BlueMark. Het tegengaan van WiFi-tracking is daardoor een serieuze probleemstelling.

Van systemen die van wireless access-points gebruik maken is het onzeker wat er met de vergaarde gegevens, die nodig zijn om een verbinding tot stand te brengen, gebeurt. Deze gegevens kunnen gelinkt worden aan smartphone-gebruikers door een clandestien WiFi-tracking-netwerk en in verkeerde handen vallen. We kunnen ons daartegen wapenen door de WiFi-module van een smartphone uit te schakelen en alleen in te schakelen wanneer we de omgeving vertrouwen. Dit blijkt in de praktijk vaak erg lastig te zijn, doordat het tijdig uitschakelen vaak wordt vergeten. Daarom stellen we in dit onderzoek een nieuwe oplossing voor: Het automatiseren daarvan met behulp van contextclassificatie, door middel van machine-learning, waarvan de bruikbaarheid en de effectiviteit benoemd wordt.

Om WiFi-tracking tegen te gaan maken we gebruik van contextclassificatie, waarvan de input geleverd wordt door sensoren op een smartphone. De meest gehanteerde definities over context en een context-aware systeem zijn van K. Dey [5] en luiden respectievelijk:

> *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*

> *A system is context-aware if it uses context to provide relevant information and/or services to the user, where dependency depends on the user's task.*

Er zijn vele artikelen over context-aware systemen op verschillende platformen en in verschillende infrastructuren verschenen, maar weinig met het doel slimme security-features in te zetten op smartphones. Door machine-learning op een smartphone te gebruiken moet het mogelijk zijn een context te classificeren en daar bepaalde acties aan te verbinden, zoals het aan –en uitschakelen van een WiFi-module. Daarmee kan een slag tegen WiFi-tracking gewonnen worden, op de momenten waarop een getrainde machine-learning-toepassing een willekeurige omgeving vertrouwd of niet. Dit zonder daarbij afhankelijk te zijn van externe systemen, die aangevallen en gespooft kunnen worden of tracking mogelijk maken, waaronder we ook GPS kunnen vinden, die alleen buitenshuis goed werkt.

Om in alle gevallen tot een juiste contextclassificatie te komen hebben we een dynamische oplossing nodig, doordat contexten constant kunnen veranderen. Dit is echter niet mogelijk wanneer we alle parameters, waarmee een contextclassificatie gemaakt kan worden, in grote databases vastleggen, want een toepassing die op deze wijze werkt zal niet met alle veranderingen om kunnen gaan. Dit impliceert dat we in dergelijke gevallen een misplaatst vertrouwen dienen te hebben in de data, die van elke sensor aangeleverd wordt. Een machine-learning-toepassing zal daarentegen wel op een dynamische wijze met de veranderende contexten om kunnen gaan, waardoor deze mogelijkheid tot een gewenste oplossing leidt. Het gebruik van een machine-learning-oplossing zal er tevens toe bijdragen dat de vertrouwelijke contextdata minder gevoelig zal zijn voor mogelijke aanvallers, doordat de harde locatiedata van verschillende omgevingsclassificaties in een database ontbreekt. Zo is dit bijvoorbeeld voor GPS-data het geval.

We concluderen dat er enkele mogelijkheden zijn om de juiste momenten te bepalen waarop een WiFi-verbinding bewerkstelligd mag worden. Het liefst met behulp van een app die een gebruiker

blijvend wil gebruiken, doordat deze bruikbaar en effectief is. Om dit doel te bereiken zien we twee mogelijkheden:

1.) Een passive-scanning-polling-app;
2.) Een machine-learning-toepassing die gebruik maakt van contextclassificatie, eventueel als aanvulling op een passive-scanning-polling-app.

Wanneer een gebruiker gebruik maakt van een WiFi-netwerk, die opgebouwd is uit meerdere access-points, die dezelfde SSID's gebruiken, zal een systeem, die van contextclassificatie gebruik maakt, minder snel leiden tot WiFi-tracking binnen dit netwerk. Voor een toepassing die enkel van passive scanning gebruik maakt ligt dit anders. Dit komt doordat WiFi-netwerken, die uit meerdere access-points bestaan en dezelfde SSID's gebruiken, misbruikt kunnen worden om bijvoorbeeld het personeel te volgen [23].

Tijdens ons onderzoek willen we de volgende onderzoeksvraag beantwoorden:

> In hoeverre kunnen we met behulp van machine-learning automatisch WiFi-tracking tegengaan op een bruikbare en effectieve wijze?

De oplossing, waarmee het gewenste antwoord gevonden wordt, moet voldoen aan bepaalde kenmerken, zoals connectivity en traceability. De snelheid waarmee een classificatie gemaakt wordt op de momenten waarop dit moet, waardoor een verbinding aan –of uitgeschakeld kan worden, is naast het energieverbruik één van de aspecten die de connectivity en traceability bepalen.

Met een machine-learning-toepassing en een passive-scanning-polling-app wordt WiFi-tracking tegengegaan, door willekeurige verbindingen met WiFi-access-points tegen te gaan. Daarbij geldt dat hoe minder dit mogelijk is, met betrekking tot de frequentie en de duur waarop dit gebeurt, hoe beter het is. Daarmee stellen we dat we WiFi-tracking in een bepaalde mate op een effectieve wijze tegengaan.

Dit brengt ons tot de volgende onderzoekshypothese:

> Machine-learning kan net zo effectief en bruikbaar zijn om tracking tegen te gaan als een passive-scanning-polling-app. Uiteraard hangt dit af van hoe beide toepassingen geoptimaliseerd zijn. Het doel van dit project is dan ook om aan te tonen dat machine-learning een effectief middel kan zijn in deze strijd door het doen van exploratief onderzoek.

Dit onderzoeksverslag is als volgt opgebouwd: Na deze inleiding geven we in hoofdstuk twee een overzicht van de achtergronden en de verschillende onderzoeken, die voldoende raakvlakken hebben met het eigen onderzoek. In hoofdstuk drie wordt de onderzoeksmethodogie uiteen gezet. In hoofdstuk vier beschrijven we de gehanteerde machine-learning-methodieken en bespreken we hoe we een contextclassificatie verfijnen. In hoofdstuk vijf wordt uitgewerkt wat we onder de begrippen connectivity en traceability precies verstaan en hoe we deze volgens meetbare schalen meer concreet maken. In hoofdstuk zes bespreken we waarom bepaalde sensoren wel of niet gebruikt worden. Dit wordt gevolgd door de architectuur van de onderzochte oplossing in hoofdstuk zeven. De exploratieve onderzoeksmethoden worden beschreven in hoofdstuk acht. De onderzoeksresultaten die daarmee behaald worden, vinden we in hoofdstuk negen. Vervolgens beschrijft hoofdstuk tien de conclusies en de aanbevelingen die naar aanleiding van de onderzoeksresultaten gedaan worden.

2. Achtergronden en gerelateerd onderzoek

WiFi maakt het mogelijk wireless netwerkverbindingen te maken tussen een smartphone en een acces-point volgens de IEEE 802.11-standaard. Deze standaard beschrijft verschillende methoden waarop een dergelijke verbinding tot stand kan worden gebracht en hoe een dataverbinding moet worden onderhouden tussen verschillende eindgebruikers. Zo onderscheiden we active en passive WiFi-scanning om een verbinding tot stand te brengen, waardoor WiFi-tracking mogelijk is.

Volgens Lorenz Schauer et al. [14] maken smartphones standaard gebruik van active scanning. Wanneer active scanning gebruikt wordt, zendt een smartphone een probe-request uit over de verschillende radiokanalen, waarna de smartphone op een mogelijke respons wacht van een access-point. Daardoor is het mogelijk een proces te starten die tot een wireless datanetwerkverbinding leidt.

Wanneer een smartphone van active scanning gebruik maakt, zal een gevulde SSID-lijst, een Preferred Network List (PNL), voorkomen dat alle in bereik zijnde WiFi-access-points moeten reageren voor de totstandkoming van een dataverbinding. Door van PNL's gebruik te maken behoeft een smartphone minder radiosignalen te analyseren op access-point-responses en daardoor minder energie te verbruiken. Zo is het tevens mogelijk een PNL leeg te laten, waardoor alle WiFi-access-points dienen te reageren op een probe-request [13]. In dit laatste geval spreken we daarom ook wel van een broadcast-probe-request. Desalniettemin zal een broadcast-probe-request sneller tot een WiFi-dataverbinding leiden, dan wanneer van passive scanning gebruik gemaakt wordt [24].

Het is ook mogelijk passive scanning te gebruiken, in plaats van de standaard gehanteerde active scanning. Lorenz Schauer et al. [14] beschreef tevens hoe passive scanning werkt. Volgens Lorenz Schauer et al. [14] wacht een smartphone over de verschillende radiokanalen op een beacon-request, voordat er een proces gestart wordt die tot een wireless dataverbinding leidt. Een beacon-request is vergelijkbaar met een probe-request, met het belangrijkste verschil dat een beacon-request door een wireless access-point uitgezonden wordt om een proces te starten.

Passive scanning kent een hoog energieverbruik, doordat een smartphone op alle mogelijke beacon-request moet wachten om ze te analyseren. Daardoor zal een dataverbinding minder snel tot stand komen, dan wanneer van active scanning gebruik gemaakt wordt. Active scanning kent een lager energieverbruik [24], maar maakt tracking door passive WiFi-tracking-netwerken makkelijker, doordat er probe-requests worden uitgezonden.

2.1. WiFi-tracking-netwerken

We onderscheiden verschillende soorten WiFi-tracking-netwerken, passive en active tracking-netwerken. Passive WiFi-tracking-netwerken maken gebruik van probe-requests die door smartphones worden uitgezonden en door deze netwerken worden opgevangen. Deze probe-requests worden door smartphones uitgezonden, die van active scanning gebruik maken, om met een WiFi-access-point een dataverbinding te maken. Zo dienen de begrippen active scanning en passive WiFi-tracking niet met elkaar verward te worden.

Active WiFi-tracking-netwerken werken anders. Ze gebruiken diensten of smartphone-toepassingen van derden, waardoor de locaties van willekeurige WiFi-access-points in het veld bij benadering beschikbaar worden gesteld over een datanetwerk.

Binnen een wireless tracking-systeem worden op bepaalde tijdsintervallen de in bereik zijnde MAC-adressen van smartphones of WiFi-access-points verzameld. Voor passive WiFi-tracing-netwerken

zijn dit de MAC-adressen van smartphones. Voor active WiFi-tracking-netwerken zijn dit de MAC-adressen van WiFi-access-points. Deze informatie kan vervolgens nauwgezet worden vastgelegd in een tuple <SID, MAC, timestamp> met de netwerk-identifier (SID). Dit stelt de eigenaar van het wireless tracking-systeem in staat de bewegingspatronen van smartphone-gebruikers te analyseren [13].

2.1.1. Passive WiFi-tracking-netwerken

Probe-requests worden door WiFi-tracking-access-points van passive WiFi-tracking-netwerken ontvangen, waardoor de harde locatiedata van een smartphone direct beschikbaar is. Deze probe-request worden gebruikt voor logging en analyse door het WiFi-tracking-netwerk. Dit gebeurt aan de hand van de onbeveiligde gegevens, die we binnen de frame-header van een control-management-frame vinden, waaruit een probe-request is opgebouwd. Daartoe dienen de exacte locaties van deze WiFi-tracking-access-points bekend te zijn in een hardware-netwerk, bij degene die het passive WiFi-tracking-netwerk onderhoudt. Dergelijke access-points worden uitsluitend voor de ontvangst van probe-requests door een passive WiFi-tracking-netwerk ingezet of ze bieden tevens de mogelijkheid frames van een WiFi-datanetwerkverbinding, die tot stand wordt gebracht, uit te luisteren.

Naast probe-request kunnen ook andere frames voor dit doeleinde gebruikt worden, die door een smartphone uitgezonden worden, tijdens of na de totstandkoming van een WiFi-verbinding. Deze frames bevatten in de header dezelfde ongecodeerde informatie als in een probe-request voor passive WiFi-tracking-systemen. Daardoor kan een geavanceerd passive WiFi-tracking-netwerk gebruik maken van een dataverbinding, die met passive scanning tot stand is gebracht, door deze verbinding uit te luisteren. Zo zijn geavanceerde passive WiFi-tracking-netwerken in staat een smartphone-gebruiker, die van passive scanning gebruik maakt, te tracken.

Niet alleen routers kunnen als een passive WiFi-tracking-acces-point geconfigureerd worden, maar ook laptops en smartphones met behulp van de juiste software. Deze apparatuur moet dan wel de monitor-mode ondersteunen uit de IEEE 802.11-standaard (2012) [13, 19]. Daarnaast worden er ook wireless tracking-acces-points door hardware-fabrikanten aangeboden. Een voorbeeld hiervan is BlueMark. Zo is een passive WiFi-tracking-netwerk alleen uit te breiden door nieuwe passive WiFi-tracking-acces-points erin op te nemen.

2.1.2. De nauwkeurigheid van passive WiFi-tracking-systemen

In de praktijk kunnen de bewegingspatronen van smartphones met behulp van passive WiFi-tracking worden bepaald en geanalyseerd. Enkele punten die een negatieve impact hebben op de nauwkeurigheid van de resultaten in dit proces zijn attenuatie of verzwakking van het radiosignaal door obstakels, afstand en/of atmosferische ruis, versterking van het radiosignaal in tunnels en de verschillende sterkten van het WiFi-signaal, dat door verschillende smartphones en niet-portable devices wordt uitgezonden. Zo dient de ruw verkregen data gefilterd en opgeschoond te worden, voordat de analyse van de bewegingspatronen plaatsvindt, door gebruik te maken van verschillende methoden en technieken [14]. Deze methoden en technieken liggen echter buiten de scope van dit onderzoek.

Volgens Junxing Zhang et al. [15] kan men met minimaal één wireless tracking-access-point nauwkeurig de in bereik zijnde smartphone-locatie bepalen. Dit kan door gebruik te maken van location-distinction en een algoritme voor de analyse. Met behulp van location-distinction is het mogelijk nauwkeurig een verplaatsing van een smartphone waar te nemen. Een location-distinction-algoritmetoepassing maakt hierbij gebruik van de multipath-radio-channel-parameters tijdens de actieve datacommunicatie. Deze parameters komen voort uit de wetenschap dat verschillende radiogolven van een radiosignaal meerdere paden afleggen van verschillende lengte. Daardoor zullen

deze golven op verschillende tijdstippen door een ontvanger ontvangen worden [15]. Daarmee is het mogelijk dat de locaties, die bij een dergelijke beweging horen, tot op een meter nauwkeurig bepaald worden [15, 16].

2.1.3. Active WiFi-tracking-netwerken

Active WiFi-tracking-netwerken werken anders. Ze gebruiken diensten van derden, waardoor de locaties van willekeurige WiFi-access-points in het veld bij benadering beschikbaar worden gesteld, over een datanetwerk [12]. Een goed voorbeeld is de database van de Google-location-service, waarin de harde locatiedata van WiFi-access-points bijgehouden en opgevraagd wordt, door gebruik te maken van smartphone-toepassingen [7]. Dit geschiedt aan de hand van de unieke parameters die we van elke WiFi-access-point in het veld vinden. Zo maakt een active WiFi-trackingnetwerk gebruik van smartphone-toepassingen en/of de diensten van derden, waardoor de locatiedata aan dit tracking-netwerk doorgeven worden. Daardoor is een active WiFi-tracking-netwerk mondiaal uit te breiden, zonder dat daar extra hardware voor nodig is. Zo dienen de begrippen active WiFi-tracking-netwerken en active scanning door een smartphone niet met elkaar verward te worden.

Smartphone-toepassingen die active WiFi-tracking mogelijk maken kunnen door een aanvaller geïnstalleerd worden, terwijl het niet duidelijk is wat er met de vergaarde gegevens gebeurt, door de aanbieder van een location-service. Op vele smartphones is het daarom mogelijk het onderhoud en het gebruik van een location-service-database uit te schakelen, maar dit wordt vaak door smartphone-gebruikers vergeten. Met behulp van een oplossing, waarmee een WiFi-module uitgeschakeld wordt, om met name passive WiFi-tracking tegen te gaan, zijn de nadelige gevolgen hiervan bijkomend te beperken.

2.1.4. Conclusie

WiFi-tracking door een active WiFi-tracking-netwerk is te ondervangen, doordat het gebruik maakt van aanvullende softwaretoepassingen op een smartphone, terwijl dit voor een passive WiFi-tracking-netwerk niet geldt. Dit komt doordat een passive WiFi-tracking-netwerk, in plaats van deze diensten, gebruik maakt van WiFi-ontvangers in een hardware-netwerk, waarmee de relevante locatiedata uitgeluisterd en bepaald wordt [12]. Daardoor zijn clandestiene passive WiFi-tracking-netwerken niet te detecteren.

2.2. MAC-adres, fingerprinting en het tegengaan van WiFi-tracking

Een mogelijke oplossing om WiFi-tracking tegen te gaan is een app waarmee het smartphone-MAC-adres regelmatig veranderd kan worden, volgens Gruteser et al. [25]. Wanneer het orginele MAC-adres niet door een tracking-systeem achterhaald wordt, zal het gespoofte MAC-adres opgemerkt en gebruikt worden in de analyse door het WiFi-tracking-systeem. Daarmee lijkt de inbreuk op de privacy van een smartphone-gebruiker in eerste aanleg deels te worden opgelost, maar niet van het publiek, bijvoorbeeld om de bezoekersstromen bij een winkelschap in kaart te brengen. Zo beschreef L. Schauer et al. [13] hoe de omvang van verschillende bezoekersstromen kan worden ingeschat.

Mathy Vanhoef et al. [17] beschreef daarentegen hoe het originele MAC-adres van een smartphone door een tracking-systeem achterhaald kan worden. Dit is het geval bij een smartphone waarop een app draait, waarmee het smartphone-MAC-adres regelmatig veranderd kan worden. Een smartphone die het MAC-adres en daarmee het probe-request van een smartphone kan veranderen, zal daartoe eerst in veel gevallen geroot of ge-jail-breaked moeten worden. Verder beschreven Mathy Vanhoef et al. [17] eveneens hoe probe-requests gefingerprint kunnen worden, waardoor de noodzaak van een smartphone-MAC-adres voor een tracking-systeem wordt ondermijnd. Deze

punten maken een oplossing, waarmee het smartphone-MAC-adres kan worden veranderd om tracking tegen te gaan, minder aantrekkelijk.

Volgens Mathy Vanhoef et al. [17] kan fingerprinting van een probe-request worden tegengegaan door de WiFi-hardware aan te passen en door de firmware en/of de drivers op smartphones aan te passen. Een voorbeeld hiervan is het onderzoek van Lindqvist et al. [24], waardoor WiFi-tracking tegengegaan kan worden door gebruik te maken van geëncrypte probe-requests en gedeelde sleutels tussen vertrouwde access-points en een smartphone. Een voorwaarde hierbij is dat een probe-request die versleuteld wordt niet opnieuw gebruikt wordt, waardoor anders fingerprinting mogelijk is. Daarmee wordt de IEEE 802.11-standaard in een beperkte mate aangepast of geüpgrade, terwijl het bestaande IEEE 802.11-protocol compatible blijft.

Volgens Mathy Vanhoef et al. [17] is de medewerking van smartphone-fabrikanten gewenst om WiFi-hardware, firmware -en/of driver-aanpassingen te doen slagen, wanneer er geen andere oplossingsrichting geformuleerd kan worden, om de probleemstelling te tackelen. Zelfs wanneer men in staat is om een dergelijke oplossing te realiseren kan dit leiden tot een wapenwedloop met de fabrikanten van tracking-systemen en wireless tracking-access-points. Derhalve zal zo'n oplossingsrichting minder aantrekkelijk zijn. Wanneer een dergelijke oplossingsrichting mogelijk is, zonder in een voortdurende wapenwedloop te vervallen, lijkt de inbreuk op de privacy van de smartphone-gebruiker voor een belangrijk gedeelte te zijn opgelost. Dit geldt echter niet voor het publiek, waarvan de bewegingspatronen kunnen worden geanalyseerd, bijvoorbeeld om de bezoekersstromen bij een winkelschap in kaart te brengen. Zo kunnen we een onderscheid herkennen tussen *individuele privacy* en *publieke privac*y, binnen de mogelijkheden van tracking-systemen.

2.3. Het tegengaan van passive WiFi-tracking met contextclassificatie

Van systemen die van wireless access-points gebruik maken is het onzeker wat er met de vergaarde gegevens, die nodig zijn om een verbinding tot stand te brengen, gebeurt. Deze gegevens kunnen gelinkt worden aan smartphone-gebruikers door clandestiene passive WiFi-tracking-netwerken en in verkeerde handen vallen, terwijl deze netwerken niet te detecteren zijn. We kunnen ons tegen deze inbreuk op de individuele en de publieke privacy wapenen door de WiFi-module van een smartphone uit te schakelen en alleen in te schakelen wanneer we de omgeving vertrouwen. Dit blijkt in de praktijk vaak erg lastig te zijn, doordat het tijdig uitschakelen vaak door een gebruiker wordt vergeten. Daarom stellen we in dit onderzoek een nieuwe oplossing voor: Het automatiseren daarvan met behulp van contextclassificatie, om de omgevingssituatie van een gebruiker in kaart te brengen. Sinds de eerste contextclassificatiesystemen, zoals Sensay [8] op een mobiele telefoon, zijn er vele artikelen over deze systemen verschenen, maar – voor zover bekend – geen van alle met het doel slimme security-features in te zetten om WiFi-tracking tegen te gaan.

Om in alle gevallen tot een juiste contextclassificatie te komen hebben we een dynamische oplossing nodig, doordat contexten constant kunnen veranderen. Dit is echter niet mogelijk wanneer we alle parameters waarmee een contextclassificatie gemaakt zou kunnen worden in grote databases vastleggen, want een toepassing die op deze wijze werkt zal niet met alle veranderingen om kunnen gaan. Dit impliceert dat we in dergelijke gevallen een misplaatst vertrouwen dienen te hebben in de data die van elke sensor aangeleverd wordt. Een machine-learning-toepassing zal daarentegen wel op een dynamische wijze met de veranderende contexten om kunnen gaan, waardoor deze mogelijkheid tot een gewenste oplossing leidt.

2.3.1. De context

Om WiFi-tracking tegen te gaan maken we gebruik van contextclassificatie. De meest gehanteerde definities over context en een context-aware systeem zijn van K. Dey [5] en luiden respectievelijk:

> *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*

> *A system is context-aware if it uses context to provide relevant information and/or services to the user, where dependency depends on the user's task.*

2.3.2. Voordelen van contextclassificatie om WiFi-tracking tegen te gaan

Naar aanleiding van de context kan de situatie, waarop een WiFi-module moet worden uitgeschakeld, in het bereik van een vertrouwde wireless acces-point liggen. Voor een context-classificatie-oplossing is dit het geval wanneer een smartphone zich tevens in het bereik van een niet-vertrouwde wireless access-point bevindt. De context die zich binnen het bereik van een vertrouwde access-point bevindt wordt daardoor in dit geval niet vertrouwd.

Wanneer een gebruiker gebruik maakt van een WiFi-netwerk, die opgebouwd is uit meerdere access-points, die dezelfde SSID's gebruiken, zal een systeem die van contextclassificatie gebruik maakt, minder snel leiden tot WiFi-tracking, binnen dit netwerk. Voor een toepassing die enkel van passive scanning gebruik maakt ligt dit anders. Dit komt doordat WiFi-netwerken, die uit meerdere access-points bestaan en dezelfde SSID's gebruiken, misbruikt kunnen worden, om bijvoorbeeld het personeel te volgen [23].

2.3.3. Self-supported en infrastructure-supported systemen

Volgens S.W. Loke et al. [5] kunnen we voor een toepassing, die gebruik maakt van contextclassificatie, gebruik maken self-supported en infrastructure-supported systemen. Wanneer een context-aware systeem volledig op een smartphone geïntegreerd is en de context waargenomen wordt met behulp van smartphone-sensoren dan kunnen we volgens hem spreken van een *self-supported-context-aware systeem*.

Hierbij maakt hij het onderscheid met *infrastructure-supported-context-awareness,* waarbij het context-aware systeem gedistribueerd is over een hardware-netwerk. Wanneer van infrastructure-supported-context-awareness gebruik gemaakt wordt impliceert dit dat er externe datacommunicatie moet plaatsvinden over een datanetwerk. Daardoor bestaat er op een smartphone de mogelijkheid dat er WiFi-tracking plaats kan vinden en de mogelijkheid dat sommige implementaties in het netwerk gespooft worden. Een voorbeeld hiervan is een machine-learninig-oplossing die van binary classification gebruik maakt, zoals beschreven door Mesenia et al. [20] in 2017. Zo dienen we niet van een infrastructure-supported-context-aware systeem gebruik te maken, maar van een self-supported-context-aware systeem.

2.3.4. De architectuur van een willekeurig self-supported-context-aware systeem

Volgens K. Dey [4] en M. Baldauf et al. [9] kan een willekeurig self-supported-context-aware systeem uit vijf lagen bestaan, om de context te classificeren:

1. Applicatielaag.
2. Inference / Storage / Management.
   Volgens Chaari et al. [10] zal de Inference-laag, die door Baldauf et al. [9] ook wel

Storage -en Managementlaag wordt genoemd, beschrijven welke acties en/of bewerkingen er toegestaan zijn of minimaal uitgevoerd dienen te worden, volgens een context.

3. Preprocessing.
   De gevonden meetwaarden worden hier mogelijk geaggregeerd tot nieuwe resultaten, waarmee de omgevingscontext wordt vastgesteld. Hier kunnen de context-interpreter en de contextmanagermodule tezamen geclassificeerd worden als een context-classifier.

4. Raw Data Retrieval.
   Hier is het mogelijk verschillende API's van verschillende soorten sensoren te gebruiken om tot gewenste meetwaarden te komen.

5. Sensors.

2.4. De bruikbaarheid

Een smartphone-toepassing, waarmee WiFi-tracking wordt tegengegaan, dient bruikbaar en effectief te zijn. Volgens Frøkjær et al. [26] wordt de bruikbaarheid bepaald door de effectiviteit, de efficiency en de gebruikerstevredenheid, tenzij voor domeinspecifieke onderzoeken in bijzondere gevallen anders nodig is. Zo wordt volgens Frøkjær et al. de effectiviteit normaliter uitgedrukt in de mate waarop een compleet en accuraat resultaat kan worden neergezet, volgens bepaalde doelstellingen die een gebruiker mag verwachten. Dit is in ons geval: Het uitschakelen van een WiFi-module, op de momenten dat een gebruiker de omgeving niet vertrouwd, en het aanschakelen van een WiFi-module, wanneer dit wel expliciet het geval is.

Volgens Frøkjær et al. [26] heeft de effectiviteit een relatie met de kwaliteit, de mate waarop de juiste resultaten tot stand komen. Omgekeerd heeft de effectiviteit daardoor ook een relatie tot het aantal fouten die een toepassing maakt. Voor de efficiency geldt volgens Frøkjær et al. [26] dat deze wordt uitgedrukt als de relatie tussen de effectiviteit en de resources, waarmee deze effectiviteit wordt bereikt. Dit met betrekking tot de tijd die nodig is om een bepaalde taak uit te voeren. Om de bruikbaarheid te bepalen verwoordt de gebruikerstevredenheid de mate waarin een gebruiker zich tevreden en comfortabel voelt met het gebruik van een applicatie, volgens Frøkjær et al. [26]. De gebruikerstevredenheid is subjectief en kan met questionnaires in kaart worden gebracht.

Volgens Nayebi et al. [27] hanteren fabrikanten daarentegen hun eigen modellen, waarmee de bruikbaarheid van smartphone-toepassingen kan worden bepaald. Ondanks dat de bruikbaarheid volgens Gafni et al. [28] geen consistent begrip is in de wetenschap kunnen we de bruikbaarheid en de effectiviteit volgens Nayebi et al. [27] op drie manier bepalen. Deze drie manieren zijn laboratoriumexperimenten, veldonderzoek en kwantitatief onderzoek. Naar aanleiding van het onderzoek van Duh et al. [29] concludeert Nayebi et al. [27] dat veldonderzoek en kwantitatief onderzoek van deze drie de beste resultaten bieden. Om dergelijke onderzoeken uit te voeren, volgens een opgezet onderzoekmodel op een experimentele smartphone-toepassing, dienen we de begrippen bruikbaarheid en effectiviteit vooraf te definiëren.

3. Methodologie

Met dit onderzoek wordt het tegengaan van automatische WiFi-tracking onderzocht, met behulp van contextclassificatie door een machine-learning-toepassing, en in hoeverre we dit op een bruikbare en effectieve wijze kunnen doen. Zo rijst de vraag hoe we dit kunnen doen. Dit brengt ons tot de volgende onderzoekshypothese:

> Machine-learning kan net zo effectief en bruikbaar zijn om tracking tegen te gaan als een passive-scanning-polling-app. Uiteraard hangt dit af van hoe beide toepassingen geoptimaliseerd zijn. Het doel van dit project is dan ook om aan te tonen dat machine-learning een effectief middel kan zijn in deze strijd door het doen van exploratief onderzoek.

Van de vele machine-learning-methodieken bespreken we er een aantal, naar aanleiding van de opgedane kennis uit de machine-learning-cursus van Andrew Ng[1] en het standaardwerk van Peter Flach [3], ter voorbereiding van het afstuderen. Uit de machine-learning-methodieken maken we een keuze voor een goede oplossingsrichting. Daartoe vatten we de verschillende machine-learning-methodieken op een bepaald abstractieniveau samen. Zo bespreken we eveneens de nadelen van verschillende machine-learning-methodieken. Hieruit blijkt dat regressie voor dit verkennend onderzoek het meest geschikt is. Dit beweegt ons een apart hoofdstuk aan regressie te wijden. Daartoe vatten we in dit hoofdstuk de volgende onderwerpen samen:

- Bayesiaanse netwerken (§ 3.1.);
- Clustering (§ 3.2.);
- Beslisbomen (§ 3.3.);
- Lineaire en logistische regressie (§ 3.4.);
- Support Vector Machines (SVM's) (§ 3.5.);
- Neurale netwerken (§ 3.6.).

Zo gaat hoofdstuk vier over lineaire en logistische regressie, in de verwachting een machine-learning-oplossing te doen slagen.

Daarom dienen we te definiëren wat een vertrouwde en een niet-vertrouwde context precies is, om een machine-learning-toepassing succesvol te laten zijn. De meest gehanteerde definities die over context gaan, komen uit het onderzoek van K. Dey [5]. Doordat zijn definities op een hoog abstractie-niveau contexten definiëren, zijn we in staat daar een invulling aan te geven, binnen het vervolg van het onderzoek. De meest gehanteerde definities over context en een context-aware systeem van K. Dey [5] luiden respectievelijk:

> *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*

> *A system is context-aware if it uses context to provide relevant information and/or services to the user, where dependency depends on the user's task.*

Deze definities stellen ons eveneens in staat om de begrippen bruikbaarheid, effectiviteit of de begrippen traceability en connectivity te definiëren voor een machine-learning-toepassing, waardoor

[1] Andrew Ng, Associate Professor, Stanford University, Chief Scientist, on-line Coursera-cursus: "Machine Learning", https://www.coursera.org/learn/machine-learning, Stanford, USA, 2016

er conclusies aan te verbinden zijn. Dit houdt in dat een machine-learning-oplossing gevalideerd moet worden. Dit kunnen we doen met automatisch gegenereerde testdata, maar dit is minder realistisch, doordat een machine-learning-toepassing in dit geval leert van een simulatie. Zo is het beter een machine-learning-oplossing niet te trainen binnen een gecontroleerde omgeving, maar tijdens het echte leven. We kiezen er daarom voor om een machine-learning-toepassing met minimaal honderd verschillende contexten uit het echte leven te trainen, voordat we onze conclusie presenteren, wetende dat een machine-learning-oplossing in theorie nooit is uitgeleerd.

Doordat er vele machine-learning-methodieken bestaan, bespreken we de belangrijkste machine-learning-methodieken uit de opgedane kennis ten behoeve van het afstuderen. Daarbij vatten we deze methodieken op een bepaald abstractieniveau samen, om vervolgens hieruit een keuze te maken voor een goede oplossingsrichting. Dit wordt in de conclusie, paragraaf 3.7., waar de voor -en nadelen van de verschillende machine-learning-methodieken worden besproken, verder toegelicht.

3.1. Bayesiaanse netwerken

Binnen Bayesiaanse netwerken wordt veelvuldig de regel van Bayes gebruikt [39]. Hierbij wordt P(B|A) uitgedrukt als de kans op B, gegeven A. De regel van Bayes luidt:

$$P(B|A) = \frac{\mathrm{P(A|B)\ P(B)}}{\mathrm{P(A)}} \qquad (1)$$

waarin:

- P(A) de kans is dat A zich voordoet en P(B) de kans is dat B zich voordoet.

Dit betekent dat deze twee laatste parameters bekend moeten zijn om een Bayesiaans netwerk te laten functioneren. Zo kan de kans op nekklachten, P(nekklachten), 0,05 zijn, terwijl het kan voorkomen dat uit een verzameling van duizend personen niemand deze klachten ervaart, waarna het mogelijk is de kans op nekklachten met dit inzicht te updaten. Dit geschiedt doorgaans op een geautomatiseerde wijze, aan de hand van een dataverzameling.

De kans op het één, gegeven het andere, kunnen we in de praktijk schematisch weergeven in een graaf die opgebouwd is uit knopen, de wijze waarop een Bayesiaans netwerk wordt voorgesteld. De waarden die bij deze knopen horen worden veelvuldig berekend met behulp van de regel van Bayes; hetgeen niet voor de knopen zonder ouders geldt.

Zo is de kans op een bepaalde contextclassificatie, of de parameters daarvan die deze classificatie omschrijven, afhankelijk van de wijze waarop een gebruiker een toepassing gebruikt en configureert. Dit is onder meer afhankelijk van het aantal gehanteerde sensoren op een smartphone. Wanneer we dit met een Bayesiaans netwerk op een goede wijze willen ondervangen impliceert dit dat er veel geïmplementeerde logica door een smartphone-gebruiker geconfigureerd moet worden. Daardoor zal de omvang, de effectiviteit en de complexiteit van dit netwerk toenemen.

3.2. Clustering

Clustering is een *soort* methode, waarmee we kunnen bepalen welke waarden tot een bepaald cluster behoren [40]. Doordat clustering een soort methode is, kunnen we verschillende machine-learning-algoritmen gebruiken, om aan deze methode invulling te geven. Bayesiaanse netwerken zijn hiervan een voorbeeld. Zo onderscheiden we verschillende methoden en technieken om het aantal clusters te bepalen, wanneer we van de clusteringmethode gebruik maken.

Een uitzondering hierop is de veelgebruikte K-means-methode. Wanneer we K-means gebruiken dienen we het aantal clusters zelf te kiezen. Er bestaat geen waterdichte methode die deze keuze ondervangt, ondanks dat de elbow-methode [41] hier soms een indicatie kan bieden. Met behulp van

de K-means-clusteringmethode kan een T-shirtfabrikant bijvoorbeeld bezien welke aantallen van een bepaalde standaard T-shirtmaat voor de markt geproduceerd moet worden, bijvoorbeeld naar aanleiding van een grafiek die de verschillende mensen uitzet tegen hun gewicht en lengte. Zo kan de clusteringmethode ook worden ingezet wanneer er geen sprake is van opeengehoopte dataclusters.

3.3. Beslisbomen

Beslisbomen zijn een machine-learning-methode waarbij elke sensorwaarde van één sensor uit een training-set geclassificeerd wordt in een nader te bepalen uniek bereik, die een andere niet overlapt. Daardoor is het mogelijk een training-set onder te verdelen in verschillende subsets.

Zo worden in opeenvolgende subsets alle sensorwaardebereiken die we bij elke sensor vinden opgenomen en worden de gewenste uitkomsten, de classificaties, daartegen uitgezet. Dit geschiedt eerst voor de eerste kolom die de sensormeetwaarden van een sensor bevat, vervolgens voor de tweede, enzovoorts. Dit net zolang totdat er voor elk van de takken, waaruit deze boomstructuur bestaat, een pure subset ontstaat; hetgeen inhoudt dat er alleen gelijke classificaties in deze subset genoteerd staan. Daarmee is het algoritme van beslisbomen voltooid, zijn alle bladeren van de boomstructuur, welke op de wortel na geheel uit subsets bestaat, gevonden en is de beslisboom gereed voor gebruik.

Zo brengen beslisbomen de nodige optimalisatieproblemen met zich mee en bestaat het risico dat er zeer grote beslisbomen ontstaan, wanneer er sprake is van vele training-examples in een training-set. Doordat we in de praktijk zeer veel verschillende contexten vinden is dit het geval, waardoor een grote complexiteit te verwachten valt.

Een ander probleem treedt op wanneer één of meerdere sensorwaarden van een context net buiten de bepaalde bereiken valt binnen een beslisboom om de uitslag te bewerkstelligen. Dit betekent dat de beslisboom opnieuw moet worden vastgesteld, nadat de training-set en de wortel van de beslisboom met deze context is bijgewerkt. Ook dit zorgt voor de nodige rekenkundige overhead, vooral bij grote beslisbomen.

Beslisbomen zijn daardoor beter toe te passen wanneer:

- de sensormeetwaarden een discreet bereik kennen en niet continu zijn;
- we te maken hebben met een beperkt aantal contexten met hun sensormeetwaarden.

3.4. Lineaire en logistische regressie

Lineaire regressie is beter in staat om met continue sensormeetwaarden om te gaan dan dat voor beslisbomen het geval is. Een uitkomst die met behulp van lineaire regressie tot stand komt is per definitie continu. Daardoor is het mogelijk bepaalde bereiken, die aan deze uitkomst worden gerelateerd, te classificeren als WiFi-module aan of als WiFi-module uit. De uitkomst waarnaar we op zoek zijn wordt daardoor discreet, met behulp van lineaire regressie.

Zo is logistische regressie eveneens beter in staat om met continue sensormeetwaarden om te gaan, dan dat voor beslisbomen het geval is. Welke van de twee regressiemethoden in de praktijk beter toepasbaar is dient te worden onderzocht. Dit geschiedt in het volgende hoofdstuk, hoofdstuk vier.

Beide regressiemethoden maken van verschillende meetbare omgevingseigenschappen, waarmee de context kan worden bepaald, gebruik. Aan de uitkomst van beide regressiemethoden ligt een

kansberekening ten grondslag om tot een contextclassificatie te komen. Daarentegen geldt dat niet alle omgevingseigenschappen even zwaar wegen om tot een contextclassificatie te komen.

Tijdens een trainingsperiode van een machine-learning-toepassing dienen we daarom onderzoek te doen naar de verschillende factoren die de gewichten bepalen. Dit doen we onder meer voor $\theta_1, \ldots, \theta_n$, de gewichten die aan de sensormeetwaarden, $x_{1,}, \ldots, x_n$, toegekend worden. Zo moet het mogelijk worden een intelligent leerproces te gebruiken, waardoor er acties aan een context-classificatie verbonden worden. Het aan of uit zetten van een verbindingsmodule is hiervan een voorbeeld. Daartoe wordt met behulp van $\theta_1, \ldots, \theta_n$ en $x_{1,}, \ldots, x_n$ een hypothesefunctie opgesteld, waarmee de gewichten, $\theta_1, \ldots, \theta_n$, worden geoptimaliseerd.

3.5. Support Vector Machines (SVM's)

Naast lineaire en logistische regressie bestaan er ook andere machine-learning-algoritmen, waarmee een hypothese kan worden opgesteld. Met een SVM is het mogelijk een discrete uitkomst te classificeren. Daardoor vertoont een SVM overeenkomsten met de reeds besproken logistische regressie. Een SVM maakt daarentegen gebruik van supportvectoren om de θ_j's te optimaliseren.

Deze vectoren worden beschreven door de datapunten die het dichtste bij een decision-boundary, een scheiding tussen verschillende groepen mogelijke contextclassificaties, liggen [43]. Daardoor is het mogelijk een discision-boundary te beschrijven met een rechte die haaks op deze datapunten staat en even ver van deze datapunten staat in een punt A. Zo worden alle datapunten, die tot één van de twee mogelijke classificaties in een training-set horen, van elkaar gescheiden. Met het vaststellen van de decision-boundary volgt dat de geoptimaliseerde θ_j's worden vastgesteld. Dit kan bijvoorbeeld in een tweedimensionale ruimte zijn, mits deze verzameling datapunten te scheiden is. In de gevallen waarin dit niet mogelijk is worden kernel-functies gebruikt, zoals dit ook voor de besproken logistische regressie geldt en in hoofdstuk vier uiteengezet wordt. De afstand tussen de decision-boundary en een supportvector wordt de maximummarge genoemd.

SVM's hebben ten opzichte van beide genoemde regressiemethoden hun voor -en nadelen, namelijk:

SVM:

1. Is makkelijker te implementeren, wanneer de datapunten uit de training-set consistent en zonder anomalieën te scheiden zijn in twee groepen, naar hun twee mogelijke classificaties;

2. Geen kansberekening voor een classificatie-uitkomst: -1 voor een negatieve classificatie, die onder het bereik van beide maximummarges valt, en +1 voor een positieve classificatie, die boven het bereik van beide maximummarges valt;

3. Kent een complexiteit van $O(n^2 \bullet k)$, waarbij k gelijk is aan het aantal supportvectoren [36].

Logistische regressie:

1. Is moeilijker te implementeren, wanneer de datapunten uit de training-set consistent en zonder anomalieën te scheiden zijn, in twee groepen naar hun twee mogelijke classificaties;

2. Maakt gebruik van kansberekening om tot een classificatie-uitkomst te komen;

3. Kent een complexiteit van $O(n^3)$;

4. Kent eveneens eigenschappen waardoor maximummarges kunnen worden ingebouwd [35].

De snelheid van SVM en logistische regressie zal niet alleen afhangen van de methoden en technieken die daarvoor worden gebruikt. Het hangt ook af van de wijze waarop ze geïmplementeerd worden en het gebruikte platform. Hierdoor is niet bekend welke sneller is.

3.6. Neurale netwerken

Een vereenvoudigde voorstelling van een neuraal netwerk kunnen we zien als een logistische regressietoepassing, binnen een eenlaags neuraal netwerk. Daardoor worden diepe neurale netwerken doorgaans ingezet, wanneer het niet haalbaar is andere methoden in te zetten, vanwege de zeer grote hoeveelheden data die verwerkt dienen te worden in een toepassing, om zaken aan te leren. We kunnen hier denken aan beeldherkenning, handgeschreven postcodeherkenning of een auto die zelfstandig moet kunnen leren rijden.

Neurale netwerken maken gebruik van verschillende nodes (neuronen). De architectuur van neurale netwerken wordt mede bepaald door het aantal mogelijke input-parameters voor de eerste architectuurlaag en het aantal mogelijke output-classificaties voor de laatste architectuurlaag. Door de verschillende neuronen binnen de verschillende neuronenarchitectuurlagen, die uit drie of meer lagen bestaan, te initialiseren met willekeurige waarden kan de output vergeleken worden met de werkelijke output. Daarmee worden de willekeurig gekozen neuroneninitialisatiewaarden, naar aanleiding van een kostfunctie, aangepast. Dit doen we met behulp van het continue backpropagation-proces voor de verschillende neuronenlagen, tot het punt waarop de toepassing is uitgeleerd. Dit is een punt waarop de kostfunctie, welke de te verwachten fout verwoordt, een zo laag mogelijke waarde bereikt. Langs deze weg is het bijvoorbeeld mogelijk dat een auto zelfstandig kan leren rijden, naar aanleiding van een chauffeur die dit voor de toepassing voor doet. Bij neurale netwerken geldt in de regel dat het toevoegen van meerdere neuronenlagen een grotere garantie geeft op het gewenste succes.

3.7. Conclusie

Doordat we in ons geval veel verschillende contexten vinden, die met name gekarakteriseerd worden door hun sensormeetwaarden die een continu bereik hebben, bieden beslisbomen geen goede oplossingsrichting.

Bij neurale netwerken geldt in de regel dat het toevoegen van meerdere neuronenlagen een grotere garantie geeft op het gewenste succes. Dit komt neer op een prijs in de vorm van de benodigde processing-power, door de complexiteit en overhead van de grote hoeveelheden data die verwerkt worden, voor het leerproces van diepe neurale netwerken. Daarmee is het twijfelachtig of neurale netwerken geschikt zijn voor een implementatie op een mobiele telefoon. Wanneer er sprake is van een variërend aantal output-classificaties, tijdens het gebruik van een dergelijke toepassing, zal dit daartoe bijdragen. Om deze redenen kiezen we niet voor neurale netwerken in het verdere verloop van dit onderzoek.

Met behulp van de clusteringmethode is het mogelijk grote hoeveelheden data te verwerken. Met deze methode is het mogelijk gegevens te classificeren, op basis van hun gemeenschappelijke kenmerken binnen een bepaald bereik. Dit leidt tot classificatieproblemen, wanneer we bij een specifieke context enkel geïnteresseerd zijn in de binaire classificatie waar of onwaar, op basis van de verschillende contexteigenschappen.

Ondanks dat een Bayesiaans netwerk equivalent kan zijn aan een machine-learning-toepassing die van logistische regressie gebruik maakt [42], kiezen we voor het laatste. Door van logistische regressie gebruik te maken verwachten we dat de benodigde logica gemakkelijker te implementeren is, waardoor de omvang en de complexiteit afneemt. Wanneer we een Bayesiaans netwerk gebruiken zal de omvang, de effectiviteit en de complexiteit daarentegen toenemen.

Doordat logistische regressie met kansberekening werkt zijn de uitslagen van een machine-learning-oplossing die hiermee werkt gemakkelijker te analyseren, want een SVM kent als enige mogelijk uitslag één of min één. Doordat niet bekend is of een SVM sneller is dan een toepassing die gebruik maakt van regressie is er, omwille van het verkennend onderzoek, besloten het eerst met logistische regressie te proberen; hetgeen in hoofdstuk vier uiteengezet wordt.

4. Contextclassificatie, regressie en machine-learning

We hebben een dynamische oplossing nodig om in alle gevallen tot een juiste contextclassificatie te komen, doordat contexten constant kunnen veranderen. Dit is echter niet mogelijk wanneer we alle parameters, waarmee een contextclassificatie gemaakt zou kunnen worden, in grote databases vastleggen, want een toepassing die op deze wijze werkt, zal niet met alle veranderingen om kunnen gaan. Dit impliceert dat we in dergelijke gevallen een misplaatst vertrouwen dienen te hebben in de data die van elke sensor aangeleverd wordt. Een machine-learning-toepassing zal daarentegen wel op een dynamische wijze met de veranderende contexten om kunnen gaan, waardoor deze mogelijkheid tot een experimentele oplossing leidt. Het gebruik van een machine-learning-oplossing zal er tevens toe bijdragen dat de vertrouwelijke contextdata minder gevoelig zal zijn voor mogelijke aanvallers, doordat de harde locatiedata van verschillende omgevingsclassificaties in een database ontbreekt.

Er zijn vele artikelen over context-aware systemen op verschillende platformen en in verschillende infrastructuren verschenen, maar – voor zover bekend – geen van alle met het doel slimme security-features in te zetten om WiFi-tracking tegen te gaan. Slimme security-features leiden ertoe dat een security-toepassing enige beredenatie dient uit te voeren, voordat er mogelijk tot de bijbehorende security-maatregelen zal worden overgegaan. Zo zal er in een beperkte mate gebruikersinteractie benodigd zijn voor de configuratie van een intelligent leerproces.

Met behulp van verschillende algoritmen is het mogelijk een machine-learning toepassing te trainen, waardoor deze leert met veranderende omstandigheden om te gaan en een voorspelling te doen over een uitkomst. Dit trainen geschiedt aan de hand van gevalideerde voorbeelden met de bijbehorende uitkomsten uit de praktijk. Deze verzameling wordt een training-set genoemd en de elementen daaruit training-examples. Deze verzameling kan gedurende de levensloop van een experimentele toepassing worden uitgebreid, waardoor ook het leerproces van deze oplossing opnieuw moet worden gestart.

Er bestaan vele algoritmen, waarmee het mogelijk is een machine-learning-toepassing te realiseren. Doordat we van een self-supported-context-aware systeem gebruik maken op een smartphone, om het WiFi-gebruik te beperken (§2.3.3.), is het noodzakelijk te bezien welke algoritmen in aanmerking komen op dit platform.

We onderscheiden verschillende soorten machine-learning-algoritmen. Volgens Andrew Yan-Tak Ng zijn de belangrijkste soorten in het werkveld:

1. Logistische regressie of binaire classificatie, waarbij de voorspellende uitkomst alleen de discrete classificatie waar of onwaar kan opleveren, naar aanleiding van een logistisch regressiealgoritme. Zo is een logistische regressie-uitkomst altijd discreet, bijvoorbeeld WiFi-module aan of uit. We spreken van logistische regressie wanneer het eindresultaat een keuze oplevert, uit één of meerdere verschillende classificaties die mogelijk zijn.

2. Lineaire regressie, waarvan de voorspellende uitkomst een continue waarde is, waardoor het mogelijk is een classificatie te maken binnen een bepaald bereik, afhankelijk van hoe een contextclassificatie gelabeld is. Daardoor is het eindresultaat een discrete classificatie, zoals dit ook bij zuivere logistische regressie geldt; hetgeen onderzocht en uiteengezet wordt.

3. Clustering (§ 3.1.2.), waarmee we kunnen bepalen welke waarden uit een grafiek tot een bepaald cluster behoren.

4. Diepe neurale netwerken (§ 3.1.6.), wanneer het niet haalbaar is andere methoden in te zetten, vanwege de zeer grote hoeveelheden data die verwerkt dienen te worden.

Doordat logistische regressie en lineaire regressie in aanmerking komen om WiFi-tracking tegen te gaan op een smartphone, worden deze twee soorten machine-learning-algoritmen als eerste in dit hoofdstuk opeenvolgend behandeld. Daarmee wordt inzichtelijk gemaakt waarom logistische regressie een betere oplossingsrichting biedt om WiFi-tracking tegen te gaan.

4.1. Lineaire regressie

De verschillende meetbare omgevingseigenschappen, waarmee de context kan worden bepaald, zoals de gemeten sensorwaarden, zullen in verschillende opzichten op verschillende tijdstippen veranderen. Daarentegen geldt dat niet alle omgevingseigenschappen even zwaar zullen wegen om tot een continue hypothese-uitkomst en een classificatie te komen. Zo zullen we in staat zijn een contextclassificatie te maken op het moment dat een continue hypothesefunctie-uitkomst binnen een bepaald bereik valt. Daartoe worden de verschillende contextclassificaties, die aan deze bereiken gerelateerd worden, door een gebruiker geconfigureerd en gelabeld.

Deze bereiken worden aan een mogelijke contextclassificatie gekoppeld, door de overeenkomstige minimale en de maximale waarden te bepalen, tijdens het leerproces van een machine-learning-oplossing. Dit impliceert dat een gebruiker meerdere invoerbevestigingen dient te doorlopen, wanneer een nieuwe training-example succesvol toegevoegd en geconfigureerd wordt, tijdens het leerproces. Zo wordt het aantal interventies dat door een gebruiker uitgevoerd wordt tot een minimum beperkt, interventies die anders nodig zijn om ongewenste tegenstrijdigheden te voorkomen.

4.1.1 Lineaire regressie en zijn hypothese

Tijdens een trainingsperiode van een machine-learning-toepassing dienen we onderzoek te doen naar de verschillende factoren die de gewichten bepalen. Dit doen we voor θ_0, …, θ_n, de gewichten die aan een constante, x_0, en de grootheden van de sensormeetwaarden, x_1, …, x_n, toegekend worden. Zo moet het mogelijk worden een intelligent leerproces te gebruiken, waardoor er acties aan een contextclassificatie verbonden kunnen worden. Het aan of uit zetten van een verbindingsmodule is hiervan een voorbeeld.

Wanneer lineaire regressie gebruikt wordt uit zich dit in een hypothese, een functie die uit meerdere termen van producten, θ_n en x_n, bestaat [45]:

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots\ldots\ldots + \theta_{n-1} x_{n-1} + \theta_n x_n \qquad (2)$$

waarbij geldt dat x_0 altijd gelijk aan één is.

Hiermee kunnen nieuwe uitkomsten voorspeld worden met een zo laag mogelijke fout. De fout bestaat hier uit het verschil tussen de voorspelde waarden, $h_\theta(x)$, en de reeds bekende echte waarden in een training-set, die continu zijn.

4.1.2. De fout en de kostfunctie, $J(\theta)$, van lineaire regressie

Nadat de meest optimale regressiehypothese is vastgesteld tijdens een trainingsperiode wordt de fout van deze hypothese berekend met behulp van de kostfunctie. Deze luidt:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left[\, h_\theta\left(x^{(i)}\right) - \, y^{(i)} \,\right]^2 \tag{3}$$

waarbij:

$y^{(i)}$ = de reeds bekende echte waarde die bij de i$^{\text{ste}}$ training-example hoort;

$h_\theta\left(x^{(i)}\right)$ = de berekende waarde die volgens een hypothesefunctie bij $y^{(i)}$ hoort;

m = het aantal training-examples.

Hiermee wordt geconcludeerd dat het convexe gradient-descent-algoritme, die we in de volgende paragraaf behandelen, tot de meest accurate binnen zijn soort behoort, om een lineaire regressiehypothese vast te stellen. Dit komt doordat de doorrekening van het convexe gradient-descent-algoritmeresultaat slechts tot één geoptimaliseerde foutmargeminimum leidt van een dergelijke regressiehypothese. Dit houdt in dat op dit minimum alle θ_j 's geoptimaliseerd zijn. Zo kennen andere manieren mogelijk het nadeel dat de doorrekening tot één van de meerdere minima kan leiden, bijvoorbeeld volgens een met bergen en dalen gevulde 3D-grafiek, waarin θ_1 en θ_2 uitgezet zijn ten opzichte van $J(\theta_1\,, \theta_2\,) = \, J(\theta)$. Daardoor is het mogelijk dat bij andere manieren het meest optimale foutmargeminimum, die tot convergentie leidt, gemist wordt; hetgeen een negatieve impact heeft op $J(\theta)$, de fout of de kostfunctie.

4.1.3. Het convexe gradient-decent-algoritme

We kunnen de lineaire regressiehypothese met de kleinste fout stapsgewijs bepalen, door het convexe gradient-descent-algoritme te gebruiken. Deze zal voor elke θ_j tot convergentie leiden en daarmee tot een bepaalde limiet reiken, waarmee de θ_j's die in de hypothese, $h_\theta(x)$, toegepast worden vastgesteld worden. Voor de hypothese met de kleinste foutmarge luidt het convexe gradient-descent-algoritme:

$$\theta_j \;:=\; \theta_j \;-\; \alpha\frac{1}{m}\sum_{i=1}^{m}\left[\, h_\theta\left(x^{(i)}\right) - \, y^{(i)} \,\right] x_j^{(i)} \tag{4}$$

waarbij:

θ_j = het gewicht of de factor van één waarde, bijvoorbeeld een sensorwaarde (licht, geluid, vochtigheid, et cetera) binnen de j$^{\text{ste}}$ term van de hypothese;

α = een redelijk gekozen stapgrote, daar we anders niet tot de gewenste convergentie komen;

i = de training-example-index;

m = het aantal training-examples;

$x_j^{(i)}$ = de j$^{\text{ste}}$ feature (bijvoorbeeld een sensorwaarde) van de i$^{\text{ste}}$ training-example;

$y^{(i)}$ = de reeds bekende echte waarde, die bij de i$^{\text{ste}}$ training-example hoort;

$x_0^{(i)}$ = 1;

$h_\theta\left(x^{(i)}\right)$ = De voorspelde waarde, volgens een hypothesefunctie die bij $y^{(i)}$ hoort. Deze zal na elke sommatie en θ_j-toekenning een nieuwe opleveren met de nieuwe θ_j.

Voor het starten van deze cyclus is het mogelijk voor elke θ_j, die geoptimaliseerd dient te worden, een willekeurige waarde in te geven. Deze cyclus zal voor elke θ_j eindigen, wanneer de convergentie bereikt is. Zo is het met behulp van het convexe gradient-descent-algoritme en een training-set mogelijk om voor elke θ_n van de lineaire regressiehypothese, $h_\theta(x)$, een optimale waarde te vinden. Deze is gelijk aan het resultaat van θ_j, na het doorlopen van de cyclus. Met een niet te groot gekozen stapgrootte α vergewissen we ons ervan dat we deze convergentie niet voorbij schieten.

Met behulp van feature-scaling (§ 6.2.1.) is het mogelijk dit proces te vergemakkelijken en te versnellen. Met behulp van mean-normalization is het mogelijk alle features van de training-examples zo te schalen dat convergentie, met behulp van het convexe gradient-descent-algoritme, gemakkelijker te bereiken is met minder iteraties.

Wanneer voor alle θ_j's de cyclus van het convexe gradient-descent-algoritme doorlopen wordt, kan de lineaire regressie-hypothese, $h_\theta(x)$, met de gevonden θ_j's geoptimaliseerd worden. Daardoor zal de hypothese een zo klein mogelijke fout kennen. Met behulp van de vastgestelde hypothese zijn de voorspellingen daardoor getalsmatige representaties van reële schattingen. Derhalve moet het mogelijk zijn om een contextclassificatie aan de uitkomst of een bepaald dynamisch bereik van een dergelijke hypothese te relateren.

De vastgestelde hypothese, $h_\theta(x)$, zou zich enigszins aan kunnen passen ,naar mate er meer bevestigde meetgegevens beschikbaar komen in de training-set met training-examples. Dit is mogelijk door het convexe gradient-descent-algoritme opnieuw te gebruiken. Zo is dit procedé als een intelligente leerproces te kenmerken. Een geïmplementeerde toepassing zal daardoor leren van het gedrag.

Met behulp van de kostfunctie, $J(\theta)$, wordt geconcludeerd dat het convexe gradient-descent-algoritme tot de meest accurate binnen zijn soort behoort, om een lineaire regressiehypothese vast te stellen. Dit komt doordat de doorrekening van het convexe gradient-descent-algoritmeresultaat slechts tot één geoptimaliseerde foutmargeminimum leidt van een dergelijke regressiehypothese. Dit houdt in dat op dit minimum alle θ_j 's geoptimaliseerd zijn. Zo kennen andere manieren mogelijk het nadeel dat de doorrekening tot één van de meerdere minima kan leiden, bijvoorbeeld volgens een met bergen en dalen gevulde 3D-grafiek, waarin θ_1 en θ_2 uitgezet zijn ten opzichte van $J(\theta_1, \theta_2) = J(\theta)$. Daardoor is het mogelijk dat bij andere manieren het meest optimale punt met de kleinste fout die tot convergentie leidt gemist wordt; hetgeen een negatieve impact heeft op $J(\theta)$, de kostfunctie. Dit geldt echter niet voor het Normal-Equation-algoritme.

4.1.4. Het Normal-Equation-algoritme

Een gelijkwaardig accuraat alternatief op het reeds genoemde gradient-descent-algoritme is het Normal-Equation-algoritme, een matrixfunctie. Dit komt doordat de doorrekening van dit algoritme slechts tot één geoptimaliseerd resultaat leidt van een hypothese, waarbij de fout volgens de kostfunctie zo klein mogelijk is. Het Normal-Equation-algoritme luidt:

$$\theta = (X^T X)^{-1} X^T y \tag{5}$$

waarbij: X = de training-examples-matrix uit de training-set, vaak de design-matrix genoemd;

y = de bijbehorende bevestigde uitkomstenmatrix;

$x_0^{(i)} \epsilon\, X$;

$x_0^{(i)}$ = 1

Wanneer x_0 , …, x_n en $y^{(1)}$, …, $y^{(n)}$ van de verschillende training-examples uit een training-set bekend zijn, hebben de design-matrix (X) en de bevestigde uitkomstenmatrix (y), binnen deze matrixfunctie (θ), respectievelijk de volgende vormen:

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \qquad \theta = [\theta_0 \quad \cdots \quad \theta_n]$$

waarbij:

- m = het aantal training-examples uit een training-set;
- n = het aantal sensoren van de training-examples;
- $x_1^{(1)}$, …, $x_n^{(m)}$ de gemeten waarden hiervan zijn.

4.1.5. Voor -en nadelen van het gradient-descent -en het Normal-Equation-algoritme

Het gebruik van het gradient-descent-algoritme en het Normal-Equation-algoritme hebben elk hun voor -en nadelen, die we als volgt kunnen samenvatten:

gradient-descent-algoritme:

1. α dienen we zelf te bepalen;
2. gebruikt vele iteraties;
3. Ook wanneer we een hoop input-features hebben werkt dit algoritme goed.

Normal-Equation-algoritme:

1. α behoeft niet meer gekozen te worden;
2. Er zijn geen iteraties nodig;
3. We dienen matrixberekeningen uit te voeren;
4. Het algoritme is erg langzaam, wanneer we met vele features te maken hebben;
5. Het algoritme kent een complexiteit $O(n^3)$, vanwege "$(X^T X)^{-1}$".

4.1.6. Mogelijke alternatieve algoritmen

Naast de laatste twee bovengenoemde methoden bestaan er ook nog andere gelijksoortige methoden om een hypothese te bewerkstelligen. Voorbeelden zijn de conjugate-gradient-methode, het Broyden–Fletcher–Goldfarb–Shanno-algoritme (BFGS) en het Limited-memory-BFGS-algoritme. Deze zijn alle zeer moeilijk te implementeren, maar ze draaien sneller dan het gradient-descent-algoritme, zonder dat daarvoor een α benodigd is.

4.1.7. Conclusie

Een machine-learning-toepassing is dynamisch, doordat het continu kan leren van nieuwe input. Wanneer een experimentele toepassing dynamisch is en het altijd een continue uitkomst heeft, zal het bereik tussen de minimale en de maximale waarde daarvan telkens anders zijn, veroorzaakt door toegevoegde training-examples. Aan deze dynamische uitkomst, waarvan het bereik tussen de minimale en de maximale waarde telkens anders is, zijn bereiken te relateren om een classificatie te maken. Dit impliceert dat deze mogelijke classificaties daarop telkens aangepast moeten worden om ze met een dynamische en een continue hypothese-uitkomst in een relatie te brengen. Dit doen we door de labels, die bij mogelijke classificaties in een training-set horen, binnen een bepaald bereik, daarop aan te passen. Dit om WiFi-module aan of WiFi-module uit te classificeren, wanneer lineaire regressie geïmplementeerd en getraind wordt.

Bij lineaire regressie is een contextclassificatie, om een WiFi-module aan of uit te zetten, daardoor indirect aan de training-example-sensorwaarden gerelateerd. Om een WiFi-module aan of uit te zetten dient een contextclassificatie echter direct gerelateerd te worden aan de individuele training-example-sensorwaarden en daardoor niet aan een dynamische en continue hypothese-uitkomst, $h_\theta\left(x^{(i)}\right)$.

Het aanpassen van de classificatielabels, waardoor dit probleem bij lineaire regressie ondervangen wordt, leidt ertoe dat een implementatie met een logistische regressiemethode een eenvoudigere oplossingsrichting oplevert. Zo is de dynamische hypothese-uitkomst, $h_\theta\left(x^{(i)}\right)$, van logistische regressie altijd discreet en direct gerelateerd aan de individuele training-example-sensorwaarden in een training-set. Anders komen we misschien tot de conclusie dat we uiteindelijk alsnog met logistische regressie bezig zijn, wanneer we met lineaire regressie starten.

4.2. Contextclassificatie met logistische regressie

Bij logistische regressie, een soort van binaire classificatie, zal de voorspellende uitkomst alleen de classificatie waar (één) of onwaar (nul) opleveren, naar aanleiding van een logistisch regressiealgoritme. Logistische regressie kan alleen ingezet worden wanneer de training-examples, die met de ene binaire classificatie gelabeld zijn ten opzichte andere, te scheiden zijn in verschillende dataclusters en het mogelijk is deze scheiding te beschrijven. In voorkomende gevallen zal het daardoor mogelijk zijn een voorspelling te doen over de discrete classificatie waar of onwaar.

4.2.1. De hypothesefunctie

De hypothesefunctie, $h_\theta(x)$, voor logistische regressie, waarmee de kans op één van de twee mogelijkheden ingeschat wordt, luidt:

$$\left.\begin{array}{c} h_\theta\left(x^{(i)}\right) = g\left(\theta^T x^{(i)}\right) \\ g(z) = \frac{1}{1+e^{-z}} \end{array}\right\} \Rightarrow h_\theta\left(x^{(i)}\right) = \frac{1}{1+e^{-\theta^T x^{(i)}}} \qquad (6)$$

waarbij: θ^T = de transpose van de gewichtenmatrix (θ_0, …, θ_n);

$x^{(i)}$ = de feature -of sensorwaardenmatrix van een training-example i;

$g(z)$ = de Sigmoid-functie.

Derhalve beschrijft de logistische hypothesefunctie, $h_\theta\left(x^{(i)}\right)$, de kans, waarop een training-example met de ene classificatie gelabeld is of met de andere, naar aanleiding van een continue training-example-uitkomst $\theta^T x^{(i)}$. Zo is de kans op een ware classificatie groter of gelijk aan 50% wanneer $\theta^T x^{(i)} \geq 0$ en kleiner dan 50% wanneer $\theta^T x^{(i)} < 0$. Om bijvoorbeeld een WiFi-module aan of uit te schakelen wordt dit in beide gevallen afgerond naar de classificatie aan (één) of de classificatie uit (nul).

4.2.2. De kostfunctie, $J(\theta)$, van zuivere logistische regressie

De fout wordt bepaald door de kostfunctie en luidt voor logistische regressie:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log\left[\, h_\theta\left(x^{(i)}\right)\right] + \left(1 - y^{(i)}\right) \log\left[\, 1 - h_\theta\left(x^{(i)}\right)\right] \qquad (7)$$

waarbij: $y^{(i)}$ = de classificatiewaarde van het label, die aan de i^{ste} training-example door een gebruiker toegekend is, altijd één of nul.

4.2.3. Het covexe gradient-decent-algoritme

Uit de voorgaande kostfunctie wordt het convexe gradient-descent-algoritme herleid, waarmee de geoptimaliseerde logistische regressiehypothese, $h_\theta(x)$, wordt opgesteld. Deze zal voor elke θ_j tot convergentie leiden en daarmee tot een bepaalde limiet reiken, waarmee de θ_j's, die in de logistische hypothese, $h_\theta(x)$, toegepast worden, berekend worden. Dit gaat analoog aan de stapsgewijze wijze, waarop dit ook voor lineaire regressie geldt (§ 4.1.3.). Het convexe gradient-descent-algoritme om de logistische regressie te bepalen luidt derhalve:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left[h_\theta\left(x^{(i)}\right) - y^{(i)} \right] x_j^{(i)} \qquad (8)$$

waarbij: $x_0^{(i)} = 1$;

$y^{(i)} \epsilon \{0, 1\}$

θ_j = het gewicht of de factor van één waarde, bijvoorbeeld een sensorwaarde (licht, aardmagnetische veldsterkte, etc.) binnen de j^{ste} term van de hypothese;

α = een redelijk gekozen stapgrote, daar we anders niet tot de gewenste convergentie kunnen komen;

i = de training-example-index;

m = het aantal training-examples;

$x_j^{(i)}$ = de j^{ste} feature (bijvoorbeeld een sensorwaarde) van de i^{ste} training-example;

$y^{(i)}$ = de classificatiewaarde van het label, die aan de i^{ste} training-example toegekend is;

$x^{(i)}$ = de featurematrix (van sensorwaarden), die bij de i^{ste} training-example hoort;

θ^T = de transpose van de gewichtenmatrix (θ_0, …, θ_n), die bij de featurematrix hoort;

$h_\theta\left(x^{(i)}\right) = \frac{1}{1 + e^{-\theta^T x^{(i)}}}$ Dit is de waarde die volgens een hypothesefunctie die bij $y^{(i)}$ hoort. Deze zal na elke sommatie en θ_j-toekenning een nieuwe opleveren met de nieuwe θ_j.

De decision-boundary-functie is uit de geoptimaliseerde logistische regressiehypothese-functie, $h_\theta(x)$, te herleiden, naar aanleiding van de daarin gehanteerde θ_j's. Voor de classificaties, waarvan de continue uitkomsten op de decision-boundary vallen, geldt dat de kans op een ware classificatie 50% is. Zo zal in de gevallen waarin een dataclusterscheiding mogelijk is tussen de training-examples, die met de ene binaire classificatie gelabeld zijn ten opzichte andere, de decision-boundary de exacte scheiding hiertussen beschrijven.

4.2.4. **Newton's logistische regressiemethode**

Een alternatief op het reeds genoemde convexe gradient-descent-algoritme, om een logistische

regressiehypothese te bepalen, is Newton's logistische regressiemethode. Newton's logistische regressiemethode kan als een matrixfunctie worden samengevat [31, 32, 44]:

$$[\theta]^{(t+1)} = [\theta]^{(t)} - \left[\frac{\frac{d}{d\theta} J(\theta)}{\frac{d^2}{d^2\theta} J(\theta)}\right]^{(t)} \tag{9}$$

Daardoor is het mogelijk bij elke volgende iteratie (t + 1) dichter tot een punt te geraken waarbij convergentie optreedt, het punt waarop de fout, $J(\theta)$, zo klein mogelijk is en de θ_j's geoptimaliseerd zijn [31, 32].

Ondanks dat bij elke iteratie de fout, $J(\theta)$, tussen t en t+1 afneemt, zal het doorlopen van deze cyclus in theorie oneindig zijn. In de praktijk betekent dit dat het doorlopen van deze cyclus afgebroken wordt, wanneer het $J(\theta)$-verschil tussen twee iteraties verwaarloosbaar klein is.

Uit de bovenstaande differentialen van formule 16 en de afgeleiden volgt de matrixfunctie [31, 32]:

$$[\theta]^{(t+1)} = [\theta]^{(t)} - [\mathrm{H}^{-1}\nabla_\theta J]^{(t)} \tag{10}$$

waarbij:

$$\mathrm{H} = \frac{1}{m}\sum_{i=1}^{m}\left[h_\theta\left(x^{(i)}\right)\left(1 - h_\theta\left(x^{(i)}\right)\right) x^{(i)}\left(x^{(i)}\right)^T\right] \tag{11}$$

$$\nabla_\theta J = \frac{1}{m}\sum_{i=1}^{m}\left[\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x^{(i)}\right] \tag{12}$$

$$h_\theta\left(x^{(i)}\right) = \frac{1}{1 + e^{-\theta^T x^{(i)}}} \tag{13}$$

Hier wordt H de Hessian-matrix genoemd en $\nabla_\theta J$ de gradient-vector. De bovenstaande formules worden tijdens elke iteratie opnieuw berekend om de fout, $J(\theta)$, te minimaliseren.
Bij het starten van de cyclus geldt voor de initialisatie van θ dat $\theta = \vec{0}$.

4.2.5. Voor -en nadelen van het convexe gradient-descent-algoritme en Newton's logistische regressiemethode

Het gebruik van het convexe gradient-descent-algoritme en Newton's logistische regressiemethode voor logistische regressie hebben elk hun voor -en nadelen, die we als volgt kunnen samenvatten [33]:

Convexe gradient-descent-algoritme:

1. Is in de regel makkelijker uit te programmeren;
2. α dienen we zelf te bepalen;
3. Gebruikt vele iteraties;
4. Elke iteratie kent een complexiteit $O(n)$;

5. Is in de regel sneller, rond de duizend toegepaste features of meer.

Newton's logistische regressiemethode:

1. Is in de regel moeilijker uit te programmeren;
2. α behoeft niet meer gekozen te worden;
3. Gebruikt veel minder iteraties;
4. In elke iteratie worden matrixberekeningen uitgevoerd;
5. Is in de regel sneller, bij minder dan duizend toegepaste features of meer.

Doordat we op een smartphone met een beperkt aantal features werken, maken we gebruik van Newton's logistische regressiemethode, doordat deze minder iteraties gebruikt en sneller tot een resultaat leidt.

4.2.6. Regressie en de Gaussian kernel-functie

In plaats van een lineaire lagere ordefunctie is het in theorie mogelijk het voorschrift van een andere algoritme te gebruiken, waarmee $\theta^T x^{(i)}$ uiteindelijk beschreven wordt. Daardoor is het mogelijk een hypothesefunctie op te stellen, waardoor een machine-learning-toepassing een lagere bias kent, maar dit komt tegen een prijs, in de vorm van de benodigde processing-power op een smartphone.

Wanneer een machine-learning-toepassing een lagere bias heeft, zal de decision-boundary de scheiding tussen de mogelijke training-examples beter beschrijven ten opzichte van hun classificaties. Om een lagere bias te bereiken zijn de euclidische afstanden tussen de verschillende training-examples, met gebruikmaking van een Gaussian kernel-functie, van belang. Dit levert een nieuwe training-set, de *kernel-training-set*, op. Hieraan worden vervolgens de kernel-training-set-example-classificaties toegevoegd, om de hypothesefunctie met een regressiemethode te bepalen. Deze contextclassificaties worden uit de oorspronkelijke training-set overgenomen, die door een gebruiker geconfigureerd wordt, zoals we later zullen zien (§ 4.2.7.).

Wanneer alle training-examples uit de originele training-set, die door een gebruiker geconfigureerd wordt, vergeleken worden met de gehele training-set, inclusief de training-example zelf, dan is er in de verzameling van mogelijke vergelijkingsuitkomsten een element waarvoor geldt dat deze uitkomst identiek is. De overeenkomst wordt hier onder meer bepaald door de euclidische afstand, $\left\|x^{(a)} - l^{(i)}\right\|$, tussen twee elementen binnen een n-dimensionale ruimte, volgens een kernel-functie (K). De elementen uit de training-set, waartegen vergeleken wordt, worden binnen deze methode landmarks ($l^{(i)}$) genoemd. Volgens de Gaussian kernel-functie kunnen we dit samenvatten tot de formule:

$$\mathrm{K}\left(x^{(a)}, l^{(i)}\right) = exp\left(-\frac{\left\|x^{(a)} - l^{(i)}\right\|^2}{2\sigma^2}\right) = exp\left(-\frac{\sum_{j=1}^{n}\left(x_j^{(a)} - l_j^{(i)}\right)^2}{2\sigma^2}\right) \qquad (14)$$

Hierbij dient opgemerkt te worden dat x_0 in de vergelijking uitgezonderd is, doordat deze met θ_0 vermenigvuldigd moet worden en $x_0 = 1$ niet van belang is om de euclidische afstand te bepalen.

Volgens de bovenstaande formule geldt dat $\mathrm{K}\left(x^{(a)}, l^{(i)}\right)$ ongeveer gelijk aan één zal zijn, op het moment dat de euclidische afstand ongeveer gelijk aan nul is. Evenzo geldt dat $\mathrm{K}\left(x^{(a)}, l^{(i)}\right)$ ongeveer gelijk aan nul zal zijn op het moment dat de euclidische afstand zeer groot is.

Met behulp van de bovenstaande formule is het mogelijk om voor elke $x^{(a)}$ uit de training-set een feature-matrix op te stellen, waarvan het aantal elementen gelijk is aan het aantal training-examples

uit de training-set. Verder geldt dat het aantal feature-matrices die opgesteld worden eveneens gelijk is aan het aantal training-examples uit de training-set. Zo is het mogelijk de geoptimaliseerde θ_j's te bepalen. Dit impliceert dat het aantal θ_j's ook gelijk zal zijn aan het aantal training-examples uit een training-set plus één, vanwege $x_0^{(i)} = 1$, die aan de nieuwe kernel-training-set wordt toegevoegd. Zo geldt dat voor elke nieuwe classificatie de sensormeetwaarden aan de kernel-functie moeten worden aangeboden, zodat met behulp van de daarmee bepaalde feature-sensormeet-waardenmatrix en de gevonden hypothesefunctie deze classificatie gemaakt kan worden.

De volledige vorm van de kernel-training-set komt in de volgende paragraaf (§ 4.2.7.) aan bod, doordat de contextclassificaties hierin nog ontbreken. Zo kunnen we voor de kernel-training-set (T_K) voor nu noteren:

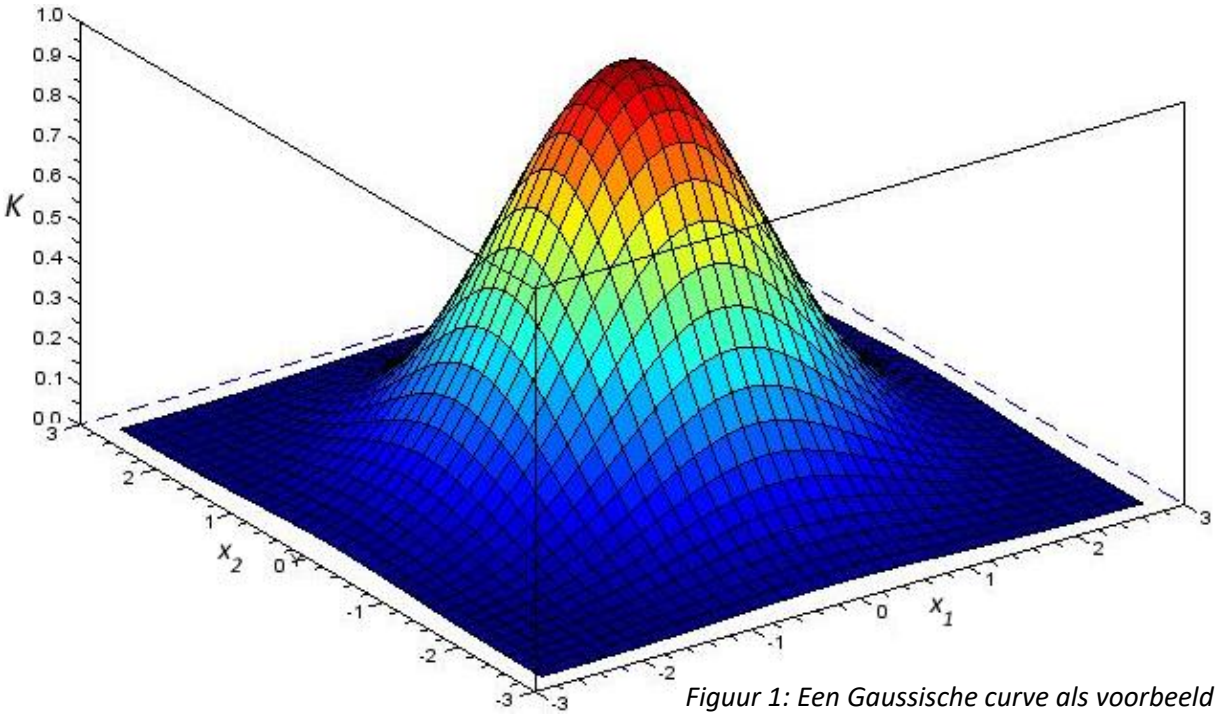
$$\mathrm{T}_{K\,onvolledig} = \begin{bmatrix} x_0^{(1)} & \mathrm{K}\left(x^{(1)}, l^{(1)}\right) & \cdots & \mathrm{K}\left(x^{(1)}, l^{(n)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & \mathrm{K}\left(x^{(n)}, l^{(1)}\right) & \cdots & \mathrm{K}\left(x^{(n)}, l^{(n)}\right) \end{bmatrix}$$

waarbij:

- n = het aantal training-examples uit de oorspronkelijke training-set;
- $x_0^{(1)}, \ldots, x_0^{(n)} = 1$

4.2.7. Regressie met de Gaussian kernel-functie en zijn variantie

In de statistiek wordt de Gaussische kansdichtheidsfunctie de standaarddeviatie (σ) genoemd en het kwadraat daarvan de variantie (σ^2). Daarentegen beschouwen we de Gaussische kansdichtheids-functie hier als een zelf in te stellen schaal, waarlangs de kernel-functie-uitkomst tussen een training-example en een landmark wordt bepaald. Hoe kleiner σ^2, hoe groter de kans dat de kernel-functie-uitkomst kleiner is. Dit wordt met een Gaussische curve als volgt inzichtelijk gemaakt voor een voorbeeld-trainings-example met slechts twee features, welke beide nul zijn, uitgezet tegen alle mogelijke landmarks:



Figuur 1: Een Gaussische curve als voorbeeld

Maken we σ^2 kleiner dan wordt de parabolische voorstelling smaller en de kans groter dat de kernel-functie-uitkomst kleiner wordt. Daardoor zal er minder snel een gelijkenis gevonden worden tussen een training-example en een landmark. Deze gelijkenis is van belang voor een kansberekening met behulp van een regressiemethode over de kernel-training-set en een contextclassificatie-uitkomst.

Daartoe wordt Newton's logistische regressiemethode (§ 4.2.4.) gebruikt en worden uit de oorspronkelijke training-set de contextclassificatiewaarden gemapt naar de overeenkomstige training-examples van de kernel-training-set. Dit geschiedt door aan de kernel-training-set een kolom toe te voegen. Daardoor krijgt de kernel-training-set (T_K) de volgende volledige vorm:

$$\mathrm{T}_K = \begin{bmatrix} x_0^{(1)} & \mathrm{K}\left(x^{(1)}, l^{(1)}\right) & \cdots & \mathrm{K}\left(x^{(1)}, l^{(n)}\right) & (aan\ of\ uit)^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_0^{(n)} & \mathrm{K}\left(x^{(n)}, l^{(1)}\right) & \cdots & \mathrm{K}\left(x^{(n)}, l^{(n)}\right) & (aan\ of\ uit)^{(n)} \end{bmatrix}$$

waarin:

- n = het aantal training-examples uit de oorspronkelijke training-set;
- $x_0^{(1)}, \ldots, x_0^{(n)} = 1$ features van de kernel-training-set zijn;
- $\mathrm{K}\left(x^{(1)}, l^{(1)}\right), \ldots, \mathrm{K}\left(x^{(n)}, l^{(n)}\right)$, de kernel-functie-features zijn van deze training-set;
- de laatste kolom de contextclassificatiewaarden zijn,
  één voor WiFi-module aan of nul voor WiFi-module uit.

Voor de variantie (σ^2) mogen we zelf een waarde kiezen, doordat σ^2 een vrije parameter is.
Dit houdt in dat we de variantie variëren, op het moment dat er veel false-positives of false-negatives plaatsvinden, gedurende de trainingsperiode van een machine-learning-oplossing, om te bezien of het daarmee beter gaat. Zo starten we het onderzoek met $\sigma^2 = 1$, gedurende de trainingsperiode van een machine-learning-oplossing.

4.2.8. Contextclassificatie met behulp van een kernel-training-set

Nadat met behulp van een regressiemethode de hypothese, $h_\theta\left(x^{(i)}\right)$, over een kernel-training-set is opgesteld, is het mogelijk een nieuwe contextclassificatie te classificeren. Zo worden de sensormeetwaarden van een context gebruikt om de kernel-functie-uitkomsten te bepalen (§ 4.2.6.), ten opzichte van alle training-examples uit de oorspronkelijke training-set. Met behulp van de variantie (§ 4.2.7.) is dit mogelijk. Om de kans op een positieve of een negatieve classificatie te berekenen wordt de hypothese, $h_\theta\left(x^{(i)}\right)$, over de kernel-training-set gebruikt, waarbij voor de te classificeren context-feature $\theta_0 x_0$ geldt dat $x_0 = 1$.

Door de toepassing van kernel-functies zal het aantal features evenredig toenemen met het aantal training-examples, die door een gebruiker worden geconfigureerd. Daardoor neemt de benodigde processing-power toe, maar door onder meer Newton's logistische regressiemethode te gebruiken wordt dit zo goed mogelijk tegengegaan.

5. Traceability en Connectivity

In navolging van paragraaf 2.4., over de bruikbaarheid, werken we in dit hoofdstuk uit wat we onder de begrippen connectivity en traceability van een toepassing verstaan en hoe we deze volgens meetbare schalen concreet kunnen maken. We nemen de connectivity en traceability als maten voor de bruikbaarheid en de effectiviteit van een toepassing, waarmee passive WiFi-tracking wordt tegengaan. Zo gaat de traceability over de mate waarop we WiFi-tracking tegengaan, terwijl de connectivity onder meer gaat over de vraag hoe snel we in een vertrouwde omgeving met een vertrouwde access-point verbinding maken. Dit impliceert dat de connectivity tevens gaat over de mate waarop vertrouwde WiFi-access-points beschikbaar zijn.

Om WiFi-tracking tegen te gaan zal een toepassing, die van contextclassificatie gebruik maakt, de WiFi-module van een smartphone uitschakelen. Een passive-scanning-polling-app zal dit echter niet doen. Een passive-scanning-polling-app tracht daarentegen een ongewenste dataverbinding met een niet-vertrouwde WiFi-access-point op een andere wijze tegen te gaan. Doordat een passive-scanning-polling-app geen gebruik maakt van probe-requests kan WiFi-tracking in een bepaalde mate worden tegengegaan. Dit geldt met name wanneer WiFi-trackingnetwerken niet in staat zijn de frames te analyseren van dataverbindingen die met passive-scanning door een smartphone tot stand worden gebracht.

5.1. De traceability

Het effectief tegengaan van WiFi-tracking wordt bepaald door de momenten waarop er geen frames naar een WiFi-access-point worden verzonden, in de situaties waarin passive WiFi-tracking kan plaatsvinden. Dit impliceert dat de traceability in deze situaties tevens bepaald wordt door de mate waarop dit wel gebeurt. In dit laatste geval wordt de traceability negatief beïnvloed door de ratio waarop een WiFi-module aan staat of de ratio waarop een passive-scanning-polling-app een verbinding tracht te maken met een willekeurige WiFi-access-point. Om de traceability te bepalen, waarmee passive WiFi-tracking kan worden tegengegaan, verdelen we de traceability onder als:

1. de *werkelijke tracebilty* in paragraaf 5.1.2., welke de mate verwoordt waarop een WiFi-module uit staat op de momenten waarop hij uit moet staan, gezien de aanname over de omgevingssituatie;

2. de *bruikbare traceability* in paragraaf 5.2., waarmee tevens de connectivity wordt bepaald, doordat clandestiene passive WiFi-tracking-netwerken niet te detecteren zijn.

Met behulp van aannamen over een omgevingscontext wordt een machine-learning-toepassing getraind. Dit houdt in dat dit onderzoek een positieve uitkomst heeft bereikt wanneer de werkelijke traceability zo laag mogelijk is, na de trainingsperiode van een machine-learning-toepassing. Wanneer dit niet het geval is moeten we ons afvragen hoe we dit kunnen verbeteren.

5.1.1. De contextclassificaties

- Onder het aantal fouten vinden we de classificaties die niet correct zijn uitgevoerd.

    1. Zo gaat de traceability over de mate waarin WiFi-tracking-netwerken gebruikers kunnen tracken, doordat er frames op de ongewenste, niet-vertrouwde momenten naar een willekeurige WiFi-access-point worden verzonden.

Een dergelijke situatie definiëren we als een false-positive (FP).

2. Evenzo gaat de traceability over de mate waarop er geen frames naar een WiFi-access-point kunnen worden verzonden, terwijl dit wel moet gebeuren, zodra dit mogelijk is, doordat de gebruiker zich in een situatie bevindt die hij uitdrukkelijk vertrouwt. Een dergelijke situatie definiëren we als een false-negative (FN).

- Het aantal classificaties, die als vertrouwd of als onvertrouwd correct zijn uitgevoerd, kunnen we eveneens omschrijven.

    1. Zo gaat de traceability ook over de momenten waarop een gebruiker de omgeving niet vertrouwd en er geen frames naar een willekeurige WiFi-access-point worden verzonden, waardoor een gebruiker niet vatbaar is voor WiFi-tracking. Een dergelijke situatie definiëren we als een true-negative (TN).

    2. Evenzo gaat de traceability over de momenten waarop een gebruiker de omgeving expliciet vertrouwd en er frames naar een willekeurige WiFi-access-point worden verzonden, zodra dit mogelijk is. Een dergelijke situatie definiëren we als een true-positive (TP).

Deze paragraaf kunnen we met behulp van een tabel als volgt samenvatten:

	Vertrouwde omgeving	Niet-vertrouwde omgeving
WiFi-module aan of frames naar een willekeurige WiFi-access-point, zodra dit moet of plaats vindt.	TP	FP
WiFi-module uit of geen frames naar een willekeurige WiFi-access-point, zodra dit moet of plaats vindt.	FN	TN

Tabel 1: Samenvatting begrippen TP, FP, FN en TN

5.1.2. De werkelijke traceability om passive WiFi-tracking tegen te gaan

Clandestiene passive WiFi-tracking-netwerken zijn niet te detecteren, terwijl we het risico op passive WiFi-tracking zo klein mogelijk maken. Wanneer we in geen geval een aanname over de omgeving hanteren om WiFi-tracking tegen te gaan doen we dit door de WiFi-module van een smartphone uit te schakelen. Daardoor zal de connectivity van een smartphone tot een dieptepunt dalen.

We hanteren daarentegen de aanname dat er alleen binnen niet-expliciet vertrouwde contexten WiFi-tracking plaatsvindt. De werkelijke traceability om WiFi-tracking tegen te gaan wordt daardoor verwoord door de ratio tussen de false-positives (FP) in verhouding tot het totale aantal positives (TP + FP), volgens een machine-learning-toepassing. Wanneer we dit procentueel willen herschrijven drukken we dit uit met behulp van de false discovery rate (FDR) en de positive predictive value (PVV):

$$FDR(TP, FP) = \frac{FP}{TP + FP} \bullet 100\% \qquad (15)$$

$$FDR(TP, FP) = (1 - PVV(TP, FP)) \bullet 100\% \quad (16)$$

$$PVV(TP, FP) = \frac{TP}{TP + FP} \quad (4)$$

De positive predictive value (PVV) wordt ook wel de precision genoemd.

Een smartphone, waarvan een WiFi-module aan staat, maar nooit verplaatst wordt binnen een vertrouwde omgeving, is een bijzondere situatie. In dit geval is de traceability, waarmee WiFi-tracking mogelijk gemaakt wordt, nul procent.

Een uitzonderlijke situatie treedt op wanneer een WiFi-module permanent uitgeschakeld is binnen één of meerdere niet-vertrouwde omgevingen. In dit laatste geval is de traceability om WiFi-tracking tegen te gaan niet met de bovenstaande formule vast te stellen, door het gebrek aan verschillende meetgegevens, die bij een experiment behoren. Dit komt doordat in deze uitzonderlijke situatie er geen sprake is van true -en false-positives en delen door nul niet mogelijk is. Bij de FDR, de PVV, de true positive rate en de false negative rate, waarvan de laatste twee behandeld worden in de volgende paragraaf, kan het voor komen dat de uitkomst niet wiskundig is vast te stellen, doordat delen door nul niet mogelijk is.

5.2. De connectivity en de bruikbare traceability

Een vertrouwde omgeving is een omgeving waarvan een gebruiker aanneemt dat zich daarbinnen geen WiFi-tracking-access-point bevindt; hetgeen we automatiseren, doordat passive WiFi-tracking-netwerken niet te detecteren zijn. Zo is een WiFi-tracking-access-point een WiFi-access-point die we niet vertrouwen, waardoor de reikwijdte daarvan een niet-vertrouwde omgeving oplevert. Dergelijke access-points worden uitsluitend voor de ontvangst van probe-requests door een passive WiFi-tracking-netwerk ingezet of ze bieden tevens de mogelijkheid frames van een WiFi-datanetwerkverbinding, die tot stand wordt gebracht, uit te luisteren.

Zowel de bruikbare als de werkelijke traceability helpen ons inzicht te verschaffen in de connectivity van een machine-learning-oplossing. De connectivity verwoordt onder meer wanneer een software-oplossing gebruik maakt van vertrouwde wireless acces-points, die binnen het bereik zijn van een vertrouwde omgeving en in welke mate. Dit wordt onder meer met de bruikbare traceabillity verwoord, door de true positive rate (TPR) en/of de false negative rate (FNR):

$$TPR(TP, FN) = \frac{TP}{TP + FN} \bullet 100\% \quad (17)$$

$$TPR(TP, FN) = (1 - FNR(TP, FN)) \bullet 100\% \quad (18)$$

$$FNR(TP, FN) = \frac{FN}{TP + FN} \quad (19)$$

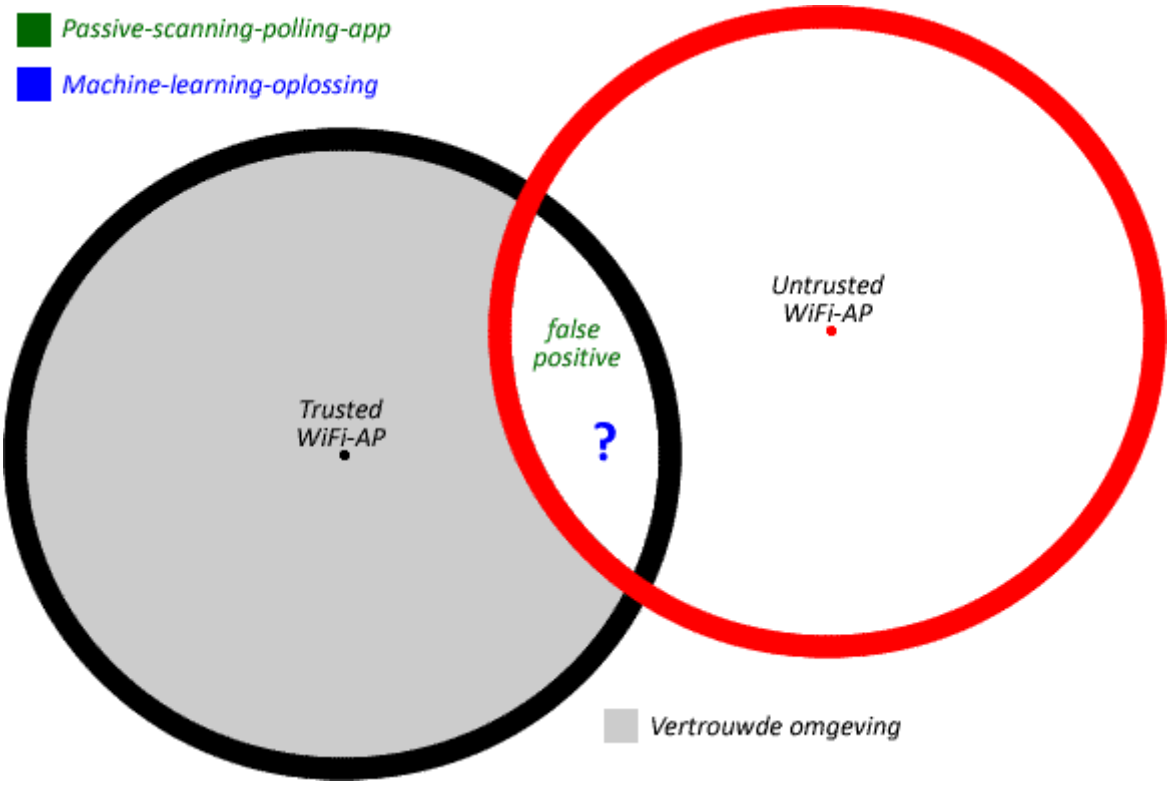
De de true positive rate (TPR) wordt ook wel de recall genoemd.

De bruikbare traceability hangt ook samen met de momenten waarop WiFi-tracking-netwerken een gebruiker kunnen tracken. Onder de bruikbare traceability vallen alle situaties waarin we de omgeving niet expliciet vertrouwen en waarbinnen een WiFi-verbinding ongeoorloofd tot stand wordt gebracht door een willekeurige toepassing. Daartoe maken we gebruik van de false discovery

rate (FDR) (§ 5.1.2.). Zo is het noodzakelijk een goede balans te vinden tussen de bruikbare traceability en de connectivity; hetgeen afhangt van hoe een toepassing geoptimaliseerd wordt.

Afhankelijk van de situatie en de context kan het moment waarop WiFi-module moet worden uitgeschakeld in het bereik van een vertrouwde wireless acces-point liggen. Voor een machine-learning-oplossing is dit het geval wanneer de omgevingssituatie niet vertrouwd wordt, doordat een smartphone zich tevens in het bereik van een niet-vertrouwde wireless access-point bevindt. In deze situatie willen we tijdens ons onderzoek bewust op een true-negative aansturen voor een machine-learning-oplossing. Voor een passive-scanning-polling-app geldt dit niet. Dit heeft een impact op de connectivity.

Dit kunnen we met behulp van een theoretische voorstelling over een situatie die we alleen in het open veld kunnen vinden als volgt weergeven:



Figuur 2: Een theoretische voorstelling over een situatie, die alleen in het open veld aangetroffen wordt, waarin WiFI-AP staat voor een WiFi-access-point. Binnen de voorstelling overlappen de reikwijdten van de twee WiFI-access-points, een vertrouwde en een niet-vertrouwde, elkaar. Het gebied dat niet binnen een vertrouwde omgeving valt behoort daardoor tot een niet-vertrouwde omgeving.

De bruikbare traceability wordt hier uitgedrukt in de mate waarop een accuraat resultaat wordt neergezet, volgens bepaalde doelstellingen: Het voorkomen dat er frames naar een willekeurige WiFi-access-points worden verzonden, door het uitschakelen van een WiFi-module in het geval dat de omgeving als niet-vertrouwd wordt aangemerkt en het omgekeerd evenredige wanneer dit wel het geval is.

Dit uitschakelen van een WiFi-module geldt alleen voor een machine-learning-oplossing. Een passive-scanning-polling-app schakelt een WiFi-module niet uit, maar voor beide toepassingen geldt met betrekking tot het tegengaan van passive WiFi-tracking:

- Dat een false-positives (FP) een situatie is waarin er frames naar een WiFi-access-point worden verzonden of een situatie waarin een WiFi-module aangeschakeld staat, terwijl dit niet mag;

- Dat een false-negative (FN) een situatie is waarin er geen frames naar een WiFi-access-point worden verzonden of een situatie waarin een WiFi-module uitgeschakeld staat, terwijl dit niet mag;

- Dat een true-positive (TP) een situatie is waarin er frames naar een WiFi-access-point worden verzonden of een situatie waarin een WiFi-module aangeschakeld staat, op het moment waarop dit moet;

- Dat een true-negative (TN) een situatie is waarin er geen frames naar een WiFi-access-point worden verzonden of een situatie waarin een WiFi-module uitgeschakeld staat, op het moment waarop dit moet.

Zo verwoordt de connectivity in welke mate een gebruiker gebruik maakt van vertrouwde wireless netwerkverbindingen, zodra dit mogelijk is.

Voor de nauwkeurigheid hanteren we hier de ratio tussen de true-positives plus de true-negatives, in verhouding tot alle negatives plus alle positives, oftewel de accuracy volgens de formule:

$$Accuracy(TP, TN, FP, FN) = \frac{TP + TN}{TP + TN + FP + FN} \bullet 100\% \qquad (20)$$

Een machine-learning-oplossing zal enige tijd nodig hebben om met de nodige beredenering tot een contextclassificatie te komen. Zo is het aantal seconden dat nodig is om een classificatie te maken op de momenten waarop dit moet, waardoor een WiFi-module aan –of uitgeschakeld wordt, een maat voor één van de connectivity-attributen.

Connectivity verwoordt echter meer dan alleen de vraag over hoe snel de functionaliteiten geleverd worden. Dit is één van aspecten die we van de connectivity kunnen vinden. Een ander attribuut van de connectivity beantwoordt de vraag hoeveel energie een toepassing vergt.

- Zo gebruikt een stand-alone passive-scanning-polling-app veel energie op de momenten waarop er een nieuwe vertrouwde access-point gezocht moet worden [24]. Daardoor kunnen we ons afvragen in welke mate een machine-learning-toepassing meer of minder energiezuinig zal zijn. Dit kunnen we meten met behulp van een toepassing die de verbruikte energie van deze twee mogelijkheden meet.

- Het tijdsinterval waarop een contextclassificatie door een machine-learning-oplossing gemaakt wordt kan worden ingesteld (hoofdstuk vier). Dit betekent dat een lagere frequentie hier zal resulteren in een lager energieverbruik.

Een hogere frequentie, waarop een machine-learning-oplossing een contextclassificatie maakt, zal resulteren in een hoger energieverbruik en een kortere reactietijd om deze classificatie te maken, op de momenten waarop dit moet. Om onze onderzoekshypothese te bewijzen dienen we daarom een machine-learning-oplossing te optimaliseren met de optimale frequentie, waarop een contextclassificatie gemaakt wordt.

Met betrekking tot de connectivity kunnen we daardoor twee dingen meten, namelijk: doet een software-oplossing het goed en doet hij het snel. Zo is het mogelijk dat ene toepassing het goed doet, terwijl de andere toepassing het snel doet. Wanneer daaruit blijkt dat ene toepassing door een gebruiker beter te gebruiken is dan de andere, terwijl voor een andere gebruiker dit andersom geldt, dan kunnen we ons afvragen welke toepassing geoptimaliseerd moet worden.

Echter, na de training van een machine-learning-toepassing vindt de classificatie van de context om een WiFi-module aan of uit te schakelen volgens een bepaalde frequentie plaats (hoofdstuk vier). Deze frequentie is in te stellen en afhankelijk van de gehanteerde smartphone. Daardoor wordt de variatie in deze parameters en de variatie in de snelheid, waarmee een WiFi-module aan of uit wordt geschakeld, te groot om te onderzoeken. In plaats daarvan willen we met dit verkennend onderzoek bezien of een machine-learning-toepassing een zinvolle manier is om passive WiFi-tracking tegen te gaan. Dat onderzoeken we door de true en de false-positives/negatives bij te houden, nadat een willekeurige classificatie heeft plaatsgevonden of zou moeten plaatsvinden. Daarmee brengen we de werkelijke en de bruikbare traceability van een machine-learning-oplossing en een passive-scanning-polling-app in kaart, waardoor het mogelijk is beide toepassingen met elkaar te vergelijken. Daarbij speelt de contextbegrenzing, waarbinnen een WiFi-module aan moet staan en daarbuiten uit, een belangrijke rol om de false discovery rate, de true positive rate en de accuracy te bepalen.

Zo worden de exploratieve onderzoeksmethoden in hoofdstuk acht beschreven. De onderzoeks-resultaten, die daarmee gevonden worden, vinden we in hoofdstuk negen. Vervolgens beschrijft hoofdstuk tien de conclusies en de aanbevelingen, die naar aanleiding van de onderzoeksresultaten behaald worden.

6. Smartphone-sensoren

In de literatuur zijn sensoren onder te verdelen in de categorieën fysieke, -virtuele -en logische sensoren [5]. Virtuele sensoren maken gebruik van verschillende soorten fysieke sensoren om tot hun geaggregeerde meetwaarden te komen. Logische sensoren gebruiken daarentegen als aanvulling hierop externe databronnen, zoals bijvoorbeeld databases, om tot hun geaggregeerde meetwaarden te komen. Fysieke sensoren worden ook wel raw-sensors genoemd, terwijl virtuele en de hier genoemde logische sensoren ook wel met synthetic sensors worden aangeduid [7]. Dit dient niet verward te worden met binary sensors, die alleen twee waarden kunnen aangeven, zoals waar of onwaar.

Volgens S.W. Loke et al. [5] kunnen we voor een oplossing, die gebruik maakt van contextclassificatie, van deze sensoren gebruik maken binnen een self-supported of een infrastructure-supported systeem. Wanneer een context-aware systeem volledig op een smartphone geïntegreerd is en de context waargenomen wordt met behulp van smartphone-sensoren dan kunnen we volgens hem spreken van een *self-supported-context-aware systeem*. Hierbij maakt hij het onderscheid met *infrastructure-supported-context-awareness*, waarbij het context-aware systeem gedistribueerd is over een hardware-netwerk.

Deze abstracte indeling hanteren we ook voor sensoren. *Infrastructurele sensoren* zijn sensoren die van externe datacommunicatie gebruik maken en daardoor tot een infrastructure-supported-context-aware systeem leiden, wanneer ze geïmplementeerd worden. Daardoor bestaat het risico dat ze gespooft kunnen worden, terwijl het gebruik van WiFi moet worden geminimaliseerd, om WiFi-tracking tegen te gaan. Zo beschreef Nils Ole Tippenhauer et al. [6] hoe hij het GPS-signaal voor civiele en militaire GPS-ontvangers vervalst. Voorbeelden van infrastructurele sensoren zijn bijvoorbeeld de sensoren die onder de standaard locatie-providers te scharen zijn van het Android-platform: de GSM-locatie-provider, de WiFi-locatie-provider en de GPS-locatie-provider [7], die alleen buitenshuis goed werkt. Een goed voorbeeld van een service, die van deze providers gebruik maakt, is de Google-location-service.

Self-supported sensoren zijn sensoren waarvan de implementatie niet leidt tot een infrastructure-supported-context-aware systeem. Zo spreken we van een self-supported-context-aware systeem wanneer alle sensoren die daarin geïmplementeerd zijn niet tot een infrastructure-supported-context-aware systeem leiden. Voorbeelden van sensoren die in een self-supported-context-aware systeem aangetroffen worden zijn: de accelerometer, de gyroscoop, de barometer, de hygrometer, de thermometer, de lichtmeter, de toesteloriëntatiemeter en de magnetometer of een kompas. Deze sensoren noemen we *self-supported sensoren*. We vinden ze op een “Samsung Galaxy S4”-smartphone en een “Samsung Galaxy S9”-smartphone, welke beide voor het onderzoek gehanteerd worden. Deze maken gebruik van het Android-platform.

Zo beschrijven we in dit hoofdstuk welke self-supported sensoren we tijdens het onderzoek wel of niet gebruiken. Tot slot sluiten we dit hoofdstuk af met een gestructureerd overzicht van de toegepaste sensoren tijdens dit onderzoek.

6.1. Sensor-over -en underfitting

Hypothese-under –of overfitting treedt op wanneer we geen goede hypothese kunnen vinden naar aanleiding van een trainings-set, waarmee redelijke voorspellingen gedaan kunnen worden. Dit kan veroorzaakt worden doordat een smartphone over te weinig of teveel sensoren beschikt, die door een machine-learning-toepassing gebruikt worden.

6.1.1. Sensor-underfitting

Doordat niet alle smartphone-typen dezelfde en hetzelfde aantal sensoren gebruiken, impliceert dit dat we in gelijke situaties enigszins andere uitkomsten kunnen verwachten, waarmee de probleemstelling getackeld wordt. Niet alle smartphones beschikken bijvoorbeeld over een temperatuursensor. Andere smartphones beschikken over te weinig sensoren, waardoor een machine-learning-toepassing niet erg effectief is. Zo is dit een verkennend onderzoek, waarmee machine-learning op een telefoon mogelijk is; hetgeen wil zeggen dat dit niet op alle mogelijke telefoons mogelijk is door sensor-underfitting. Als er sprake is van sensor-underfitting dan dient dit verder onderzocht te worden.

Deze problemen kunnen we in voorkomende gevallen met nader onderzoek trachten tegen te gaan. Dit kunnen we bijvoorbeeld doen door een time-stamp-waarde op te nemen in de verschillende trainings-examples, die we bij de totstandkoming van een hypothese nodig hebben. Daardoor zal een machine-learning-toepassing in voorkomende gevallen beter in staat zijn van het dagelijks gebruikersgedrag te leren.

We hanteren onder meer de volgende smartphone-sensoren, waarmee sensor-underfitting wordt voorkomen: De proximity-sensor, de lichtmeter (lux), de RGB -en IR-lichtsensor (candela), de oriëntatiesensor (gyroscoop), de temperatuur en een zelf samengestelde sensor, die uit een stappensensor en een lineaire acceleratiemeter bestaat. Zonder deze sensoren is het de verwachting dat sensor-underfitting op kan treden. Deze sensoren worden in paragraaf 6.4. uitgelicht.

Het kan dat het aantal gehanteerde sensoren niet voldoende is op een willekeurige smartphone. In dat geval dient nader onderzocht te worden welke sensoren nog meer in aanmerking komen. Dit komt doordat we de kans klein achten dat er een geslaagde machine-learning-oplossing uit komt met deze extra sensoren.

6.1.2. Hypothese-overfitting en feature -of sensor-overfitting

Hypothese-overfitting treedt op wanneer het functieverloop van een hypothese, $h_\theta(x)$, of een descision-boundary grillig is, terwijl deze zo goed mogelijk beschreven wordt, naar aanleiding van de classificaties van de training-examples in een training-set. Door het grillig verloop van een hypothese of een descision-boundary is het twijfelachtig of nieuwe classificaties in het veld juist geclassificeerd worden, wanneer deze niet tot de contexten, die in een training-set voor komen, behoren. Wanneer dit in deze gevallen classificatieproblemen geeft spreken we daardoor van hypothese-overfitting en stellen we dat het machine-learning-model onvoldoende generaliseert.

Feature -of sensor-overfitting treedt op wanneer we gebruik maken van vele sensoren om tot een hypothese-uitkomst te komen, terwijl we over te weinig training-examples beschikken. In dergelijke gevallen is het mogelijk dat dit in een hypothese, binnen een ruimte met vele dimensies, resulteert, die niet goed generaliseert. Sensor-overfitting kan worden tegengegaan door een model-selection-algoritme te gebruiken of zelf de belangrijkste features te selecteren, uit een verzameling gehanteerde features of sensor-meetwaarden, die mogelijk aan elkaar te relateren zijn. Zo is het mogelijk sensor-overfitting tegen te gaan door de θ_j-waarden, die de gewichten bepalen van sensor-meetwaarden, aan te passen. Dit laatste geschiedt met behulp van een regularisatiemethode. Een regularisatiemethode is beter toe te passen, wanneer we te maken hebben met veel features of sensoren, waarvan er een aantal in een ongeveer evenredige mate bijdraagt aan een hypothese-uitkomst. We verwachten echter niet dat we sensor-overfitting behoeven tegen te gaan, doordat we een bescheiden aantal sensoren gebruiken, die niet aan elkaar te relateren zijn.

6.2. Nauwkeurigheid Android-sensoren

Op het Android-platform zijn de verschillende onderdelen, zoals de verschillende sensoren, aan te spreken. Dit maakt dat vrijwel alle sensoren op een willekeurig smartphone-type gebruikt kunnen worden, ondanks dat niet elke smartphone over dezelfde en hetzelfde aantal sensoren beschikt. Doordat niet alle smartphone-typen dezelfde en hetzelfde aantal sensoren gebruiken impliceert dit dat we in gelijke situaties enigszins andere uitkomsten kunnen verwachten. Met deze uitkomsten kunnen we proberen de probleemstelling te tackelen, afhankelijk van de gebruikte smartphone.

Zo kan tevens gesteld worden dat het aantal verschillende sensortypen op een smartphone en de zuiverheid van de metingen, die daarmee gedaan worden, van belang zijn voor een mogelijke uitkomst. De zuiverheid waarmee sensormetingen gedaan worden verschillen per uitgebracht sensortype en type smartphone. Dit wordt verwoord door de maximale meetafwijking, die we voor elk geïmplementeerd sensortype op een smartphone kunnen inzien, met behulp van enkele JAVA-statements.

6.2.1. Mean-normalization & Feature-scaling

Numerieke representaties van verschillende sensorenmeetwaarden kunnen binnen dezelfde omgevingscontext op ongeveer exact hetzelfde moment sterk uiteenlopen; hetgeen een negatieve impact heeft op het verloop van een gradient-descent-algoritme om de θ_j's (hoofdstuk vier) te bepalen. Dit komt doordat er voor de verschillende sensormeetwaarden er verschillende schaalindelingen gebruikt worden. Om dit te ondervangen wordt er van mean-normalization en feature-scaling gebruik gemaakt. Daardoor is het mogelijk de numerieke representaties van de verschillende continue sensormeetwaarden ten opzichte van elkaar te normaliseren, op basis van de verschillende meetschalen die we bij deze sensoren vinden. Dit zonder dat het een negatieve invloed heeft op het verloop van een gradient-descent-algoritme om de θ_j's te optimaliseren bij het gebruik van lineaire regressie. Voor het gebruik van logistische regressie, waaronder Newton's logistische regressiemethode, wordt dit echter niet aangeraden en daardoor niet in dit onderzoek toegepast. Mean-normalization wordt bereikt door gebruik te maken van feature-scaling op de beschikbare sensormeetwaarden, volgens de formule:

$$scaling(x_j^{(i)}) = \frac{x_j^{(i)} - \frac{1}{m}\sum_{i=1}^{m} x_j^{(m)}}{max_{(x_j)} - min_{(x_j)}} \tag{22}$$

waarbij:

$x_j^{(i)}$ = de sensorwaarde van de j^{ste} feature van de i^{ste} training-example;

m = het aantal training-examples in een training-set;

$max_{(x_j)}$ = de maximale sensorwaarde van de feature *j* in een training-set;

$min_{(x_j)}$ = de minimale sensorwaarde van de feature *j* in een training-set.

Door van deze formule gebruik te maken zal de impact van een eventuele afwijking van de sensormeetwaarden verder afnemen. Door van deze formule gebruik te maken zullen alle waarden in een training-set tussen de min één en de één komen te liggen of op één van deze waarden.

6.2.2. Continuous sensors & moving median

Continuous sensors zijn sensoren die een constante stroom aan meetgegevens leveren, ongeacht de

vraag of het besturingssysteem deze data nodig heeft. Voor het Android-platform geldt dat deze stroom sneller kan gaan dan dat het operating-systeem kan verwerken, afhankelijk van het gebruikte smartphone-type [7]. Dit probleem is te ondervangen door van een aantal opeenvolgende sensormeetwaarden, welke op grootte geordend zijn in een set, steeds het middelste element te kiezen. Daardoor zal dit gekozen meetwaarde-element tevens het dichtst in de buurt komen van een reële waarde, waardoor de afwijking van een continuous sensor wordt verkleind. De methode waarmee dit gerealiseerd wordt heet de moving-median-methode. Voorbeelden van continuous sensoren zijn de magnetic sensor of het kompas, de gyroscoop, de barometer, de gravity sensor, de rotatievectormeter en de accelerometer [7, 34].

6.2.3. On-change sensors & moving median

On-change sensoren zijn sensoren die een meetwaarde geven op het moment dat deze verandert. In vrijwel alle gevallen wordt er geen moving-median-methode gebruikt om een meetwaarde van een on-change sensor te verkrijgen. Een uitzondering hierop is bijvoorbeeld de on-change lichtsensor, waarvan de meetwaarde zo vaak verandert dat het raadzaam is hierop de moving-median-methode toe te passen.

6.2.4. moving median & het verzamelen van onderzoeksdata

Continuous sensoren genereren in enkele microseconden zoveel data dat het in sommige gevallen niet mogelijk is deze hoeveelheid te verwerken, afhankelijk van het gebruikte Android-smartphone-type. Dit geldt ook voor sommige on-change sensoren. Het vastleggen van de sensordata tijdens het gebruik en de training van een machine-learning-oplossing dient daardoor gemiddeld te worden, voordat deze data voor verwerking of opslag kan worden aangeboden. Dit probleem is te ondervangen door van een aantal opeenvolgende sensormeetwaarden, welke op grootte geordend zijn in een set, steeds het middelste element te kiezen. Dit wordt de moving-median-methode genoemd. Zo is voor de uitvoering van dit onderzoek voor elk zesde element gekozen uit een set van elf meetwaarden.

6.3. Synthetische sensoren

Virtuele sensoren maken gebruik van fysieke sensoren om tot hun geaggregeerde meetwaarden te komen. Logische sensoren gebruiken als aanvulling hierop externe databronnen, zoals bijvoorbeeld databases en netwerkinformatie, om tot hun geaggregeerde meetwaarden te komen. Daardoor is het mogelijk nieuwe sensoren samen te stellen en te gebruiken in een machine-learning-toepassing. Een machine-learning-toepassing moet daarom universeel en modulair uit te breiden zijn. Deze paragraaf behandelt enkele voorbeelden van virtuele en logische sensoren, waarmee een machine-learning-toepassing kan worden uitgebreid. Deze sensoren worden ook wel synthetische sensoren genoemd.

6.3.1. Barometer

Een alternatief voor de hoogtemeter, die we op een GPS vinden, is de barometer. Met de barometer, die we op een smartphone vinden, is de hoogte bij benadering te bepalen aan de hand van een referentiepunt, waarvan de hoogte is vastgesteld en de luchtdruk kort geleden bepaald en opgeslagen is. Dit komt doordat de luchtdruk afhankelijk is van de weersinvloeden, die binnen een uur drastisch kunnen veranderen [7]. Daardoor is deze oplossingsrichting niet praktisch of bruikbaar gebleken om toe te passen.

Wanneer een betrouwbare luchtdrukwaarde van een referentiepunt niet voorhanden is, kan er in plaats daarvan gebruik gemaakt worden van de luchtdruk op zeeniveau. Deze dient echter via

internet of met het gebruik van een WiFi-module verkregen te worden. Daardoor neemt het risico op WiFi-tracking toe, waardoor ook deze oplossingsrichting niet wordt toegepast tijdens het onderzoek.

In gebieden die ver landinwaarts liggen zal de luchtdruk op zeeniveau of de luchtdruk van een ander betrouwbaar referentiepunt niet altijd voorhanden zijn, bijvoorbeeld in de Himalaya of de Sahara. Dit maakt onder meer dat de barometer geen geschikt instrument is om de hoogte te bepalen voor een machine-learning-toepassing.

6.3.2. Geluidsterkte

Onregelmatige trillingen kunnen gedurende een periode gemeten, vastgelegd en gemiddeld worden binnen verschillende logische sensoren. Daardoor kan de meetwaarde-uitslag van een synthetische sensor aan een machine-learning-oplossing worden aangeboden.

De maximale sterkte van geluidstrillingen (L) wordt gemeten met een virtuele sensor binnen een bepaalde periode. Deze metingen worden niet continu aan het Android-smartphone-besturingssysteem aangeboden. Deze sensor is niet continuous en maakt geen gebruik van de on-change-methode. De sample-frequentie, de frequentie waarop het geluidsniveau gemeten wordt, is derhalve zo hoog mogelijk binnen een bepaalde periode ingesteld.

Dit is bereikt met behulp van een for-next-loop binnen een thread. Daardoor wordt het verrichten van geluidsmetingen afgebroken, wanneer deze thread getopt wordt. Doordat de sample-frequentie zo hoog mogelijk is en het enige tijd duurt, voordat de bijbehorende platformfaciliteiten beschikbaar zijn, worden voorkomende nulwaarden gefilterd. Daardoor wordt de maximale geluidssterkte in decibel (dB) gemeten over de periode waarin gemeten wordt, met behulp van de formule:

$$L = 20 \log_{10}(amplitude) \tag{23}$$

Zo is het ook mogelijk het gemiddelde geluidsniveau en de mediaan vast te stellen van de gemeten geluidsniveaus binnen een bepaalde periode. Doordat het geluidsniveau binnen een willekeurige omgevingssituatie snel kan veranderen, waarvan elke vastgestelde waarde tot een willekeurige classificatie leidt, is het twijfelachtig of deze sensor geschikt is. Zo zijn er relatief zeer veel training-examples nodig om deze sensor binnen een machine-learning-oplossing te doen slagen (§ 6.4.3.).

De meetwaarden van een geluidsensor worden tijdens dit onderzoek bijgehouden, waardoor het achteraf mogelijk is de toegevoegde waarde van een geluidsensor vast te stellen. Dit komt doordat we niet zonder meer aannemen dat een geluidssensor voor een machine-learning-oplossing geschikt is. Zo is het mogelijk een machine-learning-oplossing met de bijgehouden meetwaarden van een geluidssensor achteraf te trainen, om hier uitsluitsel over te geven.

6.3.3. De magnetometer

De magnetometer is, net als een accelerometer, een continuous sensor. Met een magnetometer is het mogelijk de sterkte van een magnetisch veld (B) te meten in microtesla (μT). De sterkte hiervan wordt onder meer berekend met behulp van drie meetwaarden, die over de drie assen (x, y en z) van het driedimensionaal cartesisch coördinatenstelsel van een smartphone gemeten worden. Dit geschiedt volgens de formule:

$$B_{smartphone} = \sqrt{{B_x}^2 + {B_y}^2 + {B_z}^2} \tag{24}$$

Doordat elke smartphone in elke willekeurige positie gehouden kan worden betekent dit dat de maximale meetwaarde van het magnetisch veld gemeten wordt in een bepaalde richting van het magnetisch veld.

Door de drie meetwaarden, B_x, B_y en B_z, volgens het coördinatenstelsel van een smartphone, om te rekenen naar meetwaarden, volgens het coördinatenstelsel van het aardse universum, wordt dit ondervangen. Daartoe hebben we de rotatievector van de smartphone binnen het aardse universum benodigd. Deze wordt met enkele JAVA-statements verkregen, waartoe de waarden van een accelerometer (a_x, a_y en a_z), B_x, B_y en B_z benodigd zijn. Zo is de accelerometer, die hierbij gebruikt wordt, een accelerometer waaruit de werking van de zwaartekracht niet gefilterd is.

Door $\vec{B}$, welke uit B_x, B_y en B_z bestaat en door een smartphone verkregen is, te vermenigvuldigen met de rotatievector ($\vec{R}$) is het mogelijk het magnetisch veld van een bepaalde locatie te verkrijgen, volgens de formule:

$$\overrightarrow{B_{res}} = \vec{R}\ \overrightarrow{B_{x,y,z}} \tag{25}$$

waarbij:

$\overrightarrow{B_{res}}$ = De vector, die de resulterende B_x, B_y -en B_z-waarden volgens het coördinatenstelsel binnen het aardse universum bevat;

$\vec{R}$ = De rotatievector;

$\overrightarrow{B_{x,y,z}}$ = De vector, die de B_x, B_y -en B_z-waarden volgens het coördinatenstelsel van een smartphone bevat.

Met behulp van de B_x, B_y -en B_z-waarden uit $\overrightarrow{B_{res}}$ is het mogelijk het magnetisch veld te berekenen, welke onafhankelijk is van de positie waarin een smartphone gehouden wordt, volgens de formule:

$$B = \sqrt{B_{x_{res}}^2 + B_{y_{res}}^2 + B_{z_{res}}^2} \tag{26}$$

Hiermee is in theorie het aardmagnetisch veld te bepalen. Fluctuaties van het gemeten magnetisch veld kunnen daarentegen veroorzaakt worden door elektronische apparatuur in de omgeving, waardoor deze fluctuaties per locatie verschillen. Doordat er geen elektronische apparatuur in de omgeving gevonden is, die op het aardmagnetisch veld kan inwerken, is het mogelijk dat het aardmagnetisch veld dagelijks geleidelijk fluctueert. Een andere mogelijkheid is dat de nauwkeurigheid van de gehanteerde sensoren fluctueert (§ 6.4.4.), doordat het aardmagnetisch veld redelijk constant behoort te zijn [7].

Doordat de gemeten veldsterkten binnen een willekeurige omgevingssituatie veranderen, waarvan elke vastgestelde waarde tot een willekeurige classificatie leidt, is het twijfelachtig of deze sensor geschikt is (§ 6.4.4.).

6.3.4. Gebruikersactiviteit en patroonherkenning

Wanneer bepaalde gebruikersactiviteiten vastgelegd worden is het mogelijk hier patronen in te ontdekken, binnen een logische sensor en daar een bepaalde waarde aan te verbinden, zo stelt Watanabe et al. [21]. Deze waarde kan als input aangeboden worden aan een machine-learning-oplossing. Patroonherkenning kent daarentegen zijn eigen machine-learning-oplossingen en draagt niet bij tot het maken van een contextclassificatie tijdens het exploratieve onderzoek. Dit om een WiFi-module aan of uit te schakelen op een self-supported context-aware systeem, op een smartphone. Zo wordt patroonherkenning niet in dit onderzoek toegepast.

6.3.5. Netwerk en afgelegde routeinformatie

Naast de laatste netwerkinformatie is het mogelijk om de luchtdruk en de bochten van een afgelegde

route te bemonsteren, die te voet, per auto, per trein of per vliegtuig wordt afgelegd. Daarmee kan de afgelegde route zo goed mogelijk bepaald worden tussen twee punten, door onder meer gebruik te maken van een accelerometer, zoals beschreven is door Mesenia et al. [20]. Door aan het resultaat van de gebruikersactiviteit een waarde te verbinden kunnen we deze aanbieden aan een machine-learning-oplossing, om de context zo precies mogelijk te bepalen. Daartoe maken Mesenia et al. [20] gebruik van een infrastructure-supported machine-learning-oplossing. Dit komt doordat van data uit onder meer plattegronden, de actuele weerberichten, de luchtdruk en de actuele trein -en vliegdienstregelingen gebruik moet worden gemaakt.

Hiermee wordt het risico op WiFi-tracking vergroot, tenzij we dit probleem kunnen ondervangen, door op een vertrouwde wijze deze data in een database op te slaan en we geen WiFi gebruiken om de actuele luchtdruk te verkrijgen. Zo is de actuele luchtdruk van een betrouwbaar referentiepunt nodig om de hoogte te bepalen, terwijl 3G/4G-netwerken hun eigen privacy -en security-issues kennen. Doordat dienstregelingen onderhevig zijn aan veranderingen en niet alle dienstregelingen verzameld en bijgehouden kunnen worden is er niet voor deze oplossingsrichting gekozen. Ook de beperkte opslagcapaciteit op een smartphone draagt daartoe bij.

6.4. Validatie van sensoren

Voordat een willekeurige sensor binnen een machine-learning-oplossing kan worden toegepast moet worden vastgesteld of deze sensor een onderscheid kan maken tussen verschillende contexten. Dit is mogelijk door een machine-learning-oplossing met de meetwaarden van één sensor, die gevalideerd moet worden, te trainen en daarmee steekproeven uit te voeren. Bij een testuitslag van meer dan 50% hebben we de hoop dat dit zich laat generaliseren buiten onze experimenten. Bij minder dan 50% hebben we de verwachting dat dit misschien door specifieke omstandigheden van de experimenten komt, doordat we bijvoorbeeld ons in de zomer bevinden, waardoor een temperatuursensor een slecht onderscheid kan maken tussen verschillende contexten. Daardoor kiezen we ervoor een sensor met een testuitslag van 50% of minder niet voor een machine-learning-toepassing te gebruiken.

Een sensor waarvan we zeker weten dat deze in alle gevallen een machine-learning-uitslag heeft van minder dan 50% kan geïnverteerd worden om tot een bruikbaar resultaat te komen, als we alle mogelijke voorkomende contexten testen. Hiervan is echter geen sprake, doordat het aantal mogelijke contexten die in de praktijk kunnen voorkomen niet vast te stellen is, voor een toepassing die iedereen kan gebruiken.

Een sensor, die tijdens onze steekproeven het onderscheid tussen minimaal twee contexten bewerkstelligt, met een slagingspercentage van meer dan 50%, volstaat om in een machine-learning-toepassing te worden opgenomen, na nader onderzoek. Voor een lichtsensor betekent dit bijvoorbeeld dat een machine-learning-toepassing minimaal het verschil moet kunnen zien tussen kunstlicht van een bepaalde sterkte en geen licht in meer dan de helft van de gevallen.

Nadat deze eerste rudimentaire testen voltooid zijn worden de sensoren, die aan dit criterium voldoen, nader onderzocht, door de sensorwaarden binnen verschillende andere contexten vast te leggen en samen te vatten in grafieken. Hiermee wordt bewezen dat het maken van een onderscheid tussen verschillende contexten mogelijk is. Dit wordt uiteengezet in de laatste paragraaf van 6.4.

6.4.1. Het onderscheidend vermogen van verschillende sensortypen

Door een machine-learning-oplossing met de meetwaarden van één sensor, die gevalideerd moet worden, te trainen is het mogelijk in te schatten of deze sensor een onderscheid kan maken tussen

twee verschillende contexten. Dit is gedaan met een Samsung Galaxy S9. De twee verschillende contexten zijn een donker kamertje op huiskamertemperatuur en een druk kruispunt in een woonwijk met alleen autoverkeer, tijdens de wintermaanden.

Nadat aannemelijk gemaakt is welke sensoren een onderscheid maken tussen de verschillende contexten volgt in deze paragraaf een gestructureerd samenvattend overzicht van deze sensoren. Voor elke sensor zijn daartoe een zestal testen uitgevoerd tussen twee verschillende contexten, drie van de ene context naar de andere en andersom. Daarbij geldt dat een voorspelling correct is uitgevoerd wanneer een machine-learning-toepassing het verschil tussen de twee contexten opmerkt.

Sensor	Aantal training-examples	Correcte voorspellingen	Slagingspercentage
Licht (Lux)	3	6	100 %
Licht (candela)	4	6	100 %
Magnetometer	2	6	100 %
Rotation Vector *	2	6	100 %
Proximity	2	6	100 %
Battery Temperature	2	6	100 %
Gravity	2	0	0 %
Linear Acceleration	2	0	0 %

Tabel 2: Testuitslagen van onderzochte sensoren op hun geschiktheid, in een toepassing om WiFi-tracking tegen te gaan, uitgevoerd met een machine-learning-toepassing.

*) De rotation-vector-sensor is een logische sensor die gebruik maakt van een gyroscoop. Met behulp van een rotation-vector-sensor is het daardoor mogelijk de positie van een smartphone te verkrijgen na de draaiing om zijn assen. Doordat een smartphone-gyroscoop terug naar nul loopt wanneer een draaiing heeft plaatsgevonden is dit met een smartphone-gyroscoop alleen niet mogelijk.

Een gravity-sensor is te gebruiken wanneer deze zeer precies en erg gevoelig is. Dit is op een smartphone niet het geval, waardoor het niet mogelijk is een onderscheid te maken tussen verschillende contexten met een gravity-sensor bij normaal gebruik.

Voor de rotation-vectorsensor geldt dat deze een onderscheid kan maken tussen de twee contexten. Dit onder de voorwaarde dat een gebruiker een smartphone binnenshuis plat neerlegt, terwijl hij deze smartphone buitenshuis verticaal meedraagt, bijvoorbeeld in zijn borstzak.

De proximity-sensor is een binaire sensor. Voor een proximity-sensor geldt eveneens dat deze een onderscheid kan maken tussen de context binnenshuis en buitenshuis. Dit onder de voorwaarde dat een gebruiker een smartphone binnenshuis plat neerlegt, terwijl hij deze smartphone buitenshuis opgeborgen meedraagt, bijvoorbeeld in een etui of in een broekzak.

6.4.2. Het onderscheidend sensorvermogen op een Samsung Galaxy S9

Het onderzoek is voortgezet met een Samsung Galaxy S9, een opvolger van de Samsung Galaxy S4. De Samsung Galaxy S9 bezit grotendeels dezelfde sensoren als een Samsung Galaxy S4. Als aanvulling hierop bezit de Samsung Galaxy S9 diverse andere sensoren, waarvan een enkele geschikt is om een omgevingscontext in kaart te brengen. Hiermee worden de lichtintensiteiten van de lichtkleuren rood, infrarood, groen en blauw in candela gemeten. Dit in tegenstelling tot de lichtsensor, die de verlichtingssterkte in lux meet van verschillende lichtkleuren tezamen.

Met sommige Samsung-smartphones is het mogelijk ook het ultraviolette licht te meten. De sensor waarmee dit mogelijk is, wordt gebruikt om de ultraviolette straling in het directe zonlicht te meten [37]. Doordat een sensor om het ultraviolette licht te meten zich achterop een smartphone bevindt en vele contexten niet onderhevig zijn aan het directe zonlicht wordt deze sensor niet toegepast. Een sensor om de relative humidity te meten ontbreekt daarentegen op een Samsung Galaxy S9. Dit leidt er toe dat deze sensoren niet tijdens dit onderzoek worden toegepast.

6.4.3. Het onderscheidend vermogen van een geluidsensor

Voor de geluidsensor was het niet inherent duidelijk dat er een verschil tussen twee contexten moet bestaan. Tevens was het voor deze sensor niet inherent duidelijk dat er nooit een verschil tussen twee context kan bestaan. Dit komt doordat het geluidsniveau binnen een willekeurige omgevingssituatie snel kan veranderen en veel verschillende waarden kan aannemen, waarvan elke vastgestelde waarde tot een willekeurige classificatie leidt. Zelfs wanneer er sprake is van dezelfde verschillende geluiden, die zich binnen een bepaalde omgevingssituatie voordoen, kan de mate waarop ze zich voordoen en het geluidsniveau ervan variëren. Dit maakt het geluidsniveau, die door een geluidsensor (§ 6.3.2.) gemeten wordt, twijfelachtig om toe te passen tijdens dit onderzoek. Zo zijn er relatief zeer veel training-examples nodig om een sensor, die alleen het geluidsniveau meet, misschien te doen slagen, binnen een machine-learning-oplossing.

Echter, het is in theorie mogelijk bepaalde patronen te ontdekken van verschillende geluiden. Deze geluiden zijn binnen een synthetische sensor te analyseren, waardoor er een bepaalde waarde aan te verbinden is. Deze waarde kan als input aangeboden worden aan een machine-learning-oplossing. Patroonherkenning kent daarentegen zijn eigen machine-learning-oplossingen en draagt niet bij tot het maken van een contextclassificatie tijdens dit exploratieve onderzoek. Dit om een WiFi-module aan of uit te schakelen op een self-supported context-aware systeem, op een smartphone. Zo wordt patroonherkenning niet in dit onderzoek toegepast.

Doordat geluiden onderhevig zijn aan veranderingen en niet alle geluiden verzameld en bijgehouden kunnen worden is er niet voor patroonherkenning gekozen. De beperkte opslagcapaciteit op een smartphone draagt daartoe bij. We verwachten dat het toepassen van patroonherkenning of herkenning van het gebruikersgedrag leidt tot een te grote prijs in de benodigde processing-power en opslagruimte op een smartphone. Zo verwachten we dat voor het onderscheidend vermogen van een synthetische geluidsensor met behulp van diepe neurale netwerken (§ 3.6. en § 3.7.) een kans van slagen heeft, wanneer we over meer computer-power beschikken. Dit laatste is echter niet het geval op een smartphone. Daardoor wordt patroonherkenning niet in dit exploratieve onderzoek toegepast, in relatie tot de beschikbare projectdoorlooptijd.

6.4.4. Het onderscheidend vermogen van de magnetometer

Voor de magnetometer was het niet inherent duidelijk dat er een verschil tussen twee contexten moet bestaan. Tevens was het voor deze sensor niet inherent duidelijk dat er nooit een verschil tussen twee context kan bestaan. Dit komt doordat het magnetisch veld binnen een willekeurige omgevingssituatie geleidelijk lijkt te veranderen en veel verschillende waarden kan aannemen,

gedurende een dag. Dit terwijl het aardmagnetisch veld redelijk constant behoort te zijn en er geen elektronische apparatuur in de meetomgeving te vinden is, die op dit magnetisch veld inwerkt.

Daardoor is er gekeken naar de nauwkeurigheid van de gehanteerde sensoren (§ 6.3.3.), waarmee het magnetisch veld wordt bepaald. Volgens Greg Milette en Adam Stroud [7] is de nauwkeurigheid van een magnetometer op een smartphone een probleem, door de geïmplementeerde hardware. Ze stellen daarentegen dat de nauwkeurigheid van een magnetometer gemakkelijk vast te stellen en gemakkelijk te kalibreren is. Dit houdt in dat we voor het vastleggen van het magnetisch veld de nauwkeurigheid van de magnetometer moeten testen en deze eventueel moeten kalibreren.

Met een magnetometer is het magnetisch veld te meten. Echter, veel smartphonehoesjes hebben een magnetische sluiting, die deze sensor beïnvloeden en de resultaten daarvan onbetrouwbaar maken. Zo kunnen ook andere materialen van metaal het magnetisch veld in de nabijheid verstoren. Een bureaublad van metaal of een sleutelbos zijn hiervan voorbeelden. Hoe dit in de praktijk uitpakt is moeilijk te voorspellen. We zien daartoe enkele mogelijkheden:

- Verstoorde sensordata verzamelen, waarna we alle negatieve invloeden hieruit filteren;
- Mogelijke verstoringen in de verkregen sensordata accepteren.

In het laatste geval zal de verkregen sensordata soms vervuild zijn, bijvoorbeeld door een sleutelbos, waardoor deze data in het algemeen niet erg betrouwbaar is. Wanneer de verstoring hier op alle dagen een constant gegeven is, dan kunnen we dit verschijnsel filteren of extrapoleren, maar dit is niet het geval. Daarbij merken we op dat een constante verstoring van het magnetisch veld, tijdens het dagelijks gebruik van een smartphone in verschillende situaties, niet realistisch is.

Dit maakt het magnetisch veld, die door een magnetometer gemeten wordt, twijfelachtig om toe te passen tijdens dit onderzoek. Zo worden de meetwaarden van een magnetometer tijdens dit onderzoek bijgehouden, waardoor het achteraf mogelijk is de toegevoegde waarde van een magnetometer vast te stellen. Dit komt doordat we niet zonder meer kunnen aannemen dat een magnetometer voor een machine-learning-oplossing geschikt is.

6.4.5. De synthetische bewegingssensor

Een context is niet per definitie gebonden aan een fysieke omgeving (§ 2.3.1.). Een context kan ook een bepaalde situatie, een bepaalde gebeurtenis of een actie zijn die zich in een willekeurige omgeving voordoet, zoals bijvoorbeeld wandelen, fietsen of brommerrijden. Tijdens het wandelen, fietsen of brommerrijden is er gemeten met de lineaire acceleratiemeter, die op een smartphone te vinden is.

Door de lineaire acceleratiemeter met een voetstappensensor samen te vatten in een synthetische sensor is het daardoor mogelijk een onderscheid te maken tussen de volgende vier situaties of contexten:

1) Geen beweging (0 m/s^2), gedetecteerd met de lineaire acceleratiemeter;
2) Lopen op verschillende snelheden, gedetecteerd met een stappensensor;
3) Fietsen en brommerrijden, het gemechaniseerd vervoer, gedetecteerd met de lineaire acceleratiemeter;
4) Geen van de drie voorgaande contexten.

Lopen wordt gedetecteerd met een stappensensor, die we op een smartphone vinden. Dit is voor het onderscheid noodzakelijk, doordat de gemiddelde acceleratie tijdens het lopen bij benadering ongeveer 1¼ meter per seconde in het kwadraat bedraagt, hetzelfde als voor het fietsen geldt.

Uiteraard hangt een gemiddelde af van het gebruikersgedrag, want niet elke gebruiker zal hetzelfde gedrag vertonen. Om een machine-learning-oplossing te trainen is het nodig om het aantal meters per seconde in het kwadraat in een training-example op te nemen binnen een feature. Dit laatste geldt echter niet voor het wandelen en het rennen, want een stappensensor geeft geen uitslag in meters per seconde in het kwadraat. De stappensensor is een binaire sensor of de stappensensor is een logische sensor, die het aantal afgelegde stappen teruggeeft sinds de smartphone is aangezet.

Beide soorten stappensensoren hebben echter iets gemeen: Ze geven een event op het moment dat een gebruiker een stap verzet, terwijl hij aan het lopen is. Daardoor is het mogelijk een vaste waarde binnen een feature in een training-example op te nemen voor de contexten wandelen en rennen, bijvoorbeeld 3,75. Dit binnen dezelfde feature die door de lineaire acceleratiemeter gebruikt wordt, op het moment dat er van wandelen geen sprake is. Daardoor is een machine-learning-toepassing in staat het onderscheid te maken tussen de verschillende contexten, die aan beweging gerelateerd worden.

In theorie geldt dit niet alleen voor wandelen en rennen, fietsen en brommerrijden, maar mogelijk ook voor andere vormen van vervoer, waarover verder onderzoek mogelijk is. Zo is het onderzoek verder gezet met een Samsung Galaxy S9. Deze opvolger van de Samsung Galaxy S4 bezit een motion-sensor, die met de Samsung Motion SDK [38] en de daarvoor ontwikkelde API te gebruiken is; hetgeen op een Samsung Galaxy S4 ontbreekt. Voor het detecteren van beweging kennen de oplossingen met een Samsung Galaxy S4 en een Samsung Galaxy S9 elk hun voor -en nadelen. Deze sommen we als volgt op:

Samsung Galaxy S4:

1. Toepassing is ongeveer 50 milliseconden sneller, in vergelijking met een Samsung Galaxy S9 tijdens het wandelen;
2. Maakte één fout in 65 vastgelegde meet-samples tijdens het wandelen;
3. Kan de contexten geen beweging, lopen, fietsen en brommerrijden detecteren;
4. Geeft alleen voor lopen een vaste waarde terug.

Samsung Galaxy S9:

1. Motion-sensor-toepassing is ongeveer 50 milliseconden langzamer, in vergelijking met een Samsung Galaxy S4 tijdens het wandelen;
2. Maakte ongeveer 6 fouten in 65 vastgelegde meet-samples tijdens het wandelen;
3. Kan alleen de contexten geen beweging, wandelen, rennen en op of in een voertuig detecteren;
4. Geeft voor elk van de voortbewegingsmogelijkheden een unieke discrete waarde terug, waarbij de contexten wandelen en rennen verder onderverdeeld kunnen worden. Het lopen kan horizontaal en

in beide verticale richtingen onderverdeeld worden. Deze laatste onderverdeling geschiedt aan de hand van unieke discrete waarden, die bij deze onderverdeling horen.

De conclusies bij punten één en twee van beide oplossingen werden getrokken na een vergelijkend onderzoek en de daarbij verzamelde onderzoeksdata, een dataset van 65 on-change-events of vastgelegde meet-samples tijdens het wandelen.

Zo kan er een keuze gemaakt worden tussen de zelf samengestelde synthetische bewegingssensor en de Samsung-bewegingssensor, die we op een Samsung Galaxy S9 vinden. Daarin speelt het korte verschil in de onderlinge sensorreactietijd van ongeveer 50 milliseconden geen beduidende rol, maar het aantal sensormeetfouten wel. Tijdens ongeveer 18 minuten brommerrijden bleek het aantal fouten van de Samsung-bewegingssensor op een Samsung Galaxy S9 erg hoog te zijn. Naar schatting werden 68% van de meet-sampels door deze sensor verkeerd geclassificeerd tijdens het brommerrijden.

Zo is de Samsung-bewegingssensor op een Samsung Galaxy S9 niet geschikt gebleken voor dit onderzoek en is er voor de toepassing van de zelf samengestelde synthetische bewegingssensor gekozen. De zelf samengestelde synthetische bewegingssensor maakte tijdens hetzelfde testgeval, het brommerrijden of het gemechaniseerde vervoer, waaronder het brommerrijden valt, geen enkele classificatiefout binnen het continue bereik van een acceleratiemeter.

6.4.6. Metingen met succesvolle sensortypen

Met behulp van de in deze paragraaf behandelde sensoren is het mogelijk het onderscheid tussen verschillende contexten maken. Daartoe werden binnen verschillende contexten metingen verricht. Deze contexten waren het grasveld, in een park tijdens een zonnige dag, de lokale supermarkt of de lokale Albert Heijn, de studeerkamer, brommerrijden, fietsen, wandelen en geen beweging. De meetgegevens, die bij de uitslagen van tabel vier behoren, zijn in de bijlagen van dit onderzoek te vinden. Daaronder vinden we de gegenereerde grafieken en de tijdstippen, waarop deze metingen uitgevoerd zijn.

Sensor	Albert Heijn	Grasveld	Studeerkamer	Opmerkingen
Geluid	80 dB	64 dB	53 dB	Gemiddelde
Licht (Lux)	209 Lux	47.000 Lux	142 Lux	Gemiddelde
Magnetometer	84,07 μT	46,65 μT	84,55 μT	Gemiddelde van het totale aardmagnetisch veld.
Licht (Candela) * - Lichtkleur A -	710 cd	Oververzadiging Sensor	264 cd	Gemiddelde van één van de lichtkleuren
Licht (Candela) * - Lichtkleur B -	341 cd	2.853 cd	120 cd	Gemiddelde van één van de lichtkleuren

Licht (Candela) * - Lichtkleur C -	446 cd	4.012 cd	138 cd	Gemiddelde van één van de lichtkleuren
Licht (Candela) * - Lichtkleur D -	385 cd	4.525 cd	138 cd	Gemiddelde van één van de lichtkleuren
Battery Temperature	23,25 °C	42,89 °C	27,22 °C	Gemiddelde

Tabel 3: Uitslagen van sensoren met een relevant onderscheidend vermogen.

*) Van de TMD4906-sensor, waarmee de verschillende lichtkleuren in candela gemeten worden op een Samsung Galaxy S9, is praktisch niet te achterhalen welke van de lichtkleuren de lichtkleuren rood, infrarood, groen en blauw betreffen. Dit komt doordat meetopstellingen om dit te verifiëren ontbreken. Het is wel duidelijk dat de vier lichtkleuren A tot en met D elk één verschillende lichtkleur van de vier lichtkleuren rood, infrarood, groen en blauw meten.

In tabel vier wordt van een geschat gemiddelde gesproken. Voor het geluid, het licht (lux) en de magnetometer is dit gemiddelde berekend met Plotly [47], te vinden op https://plot.ly/create/#/ Plotly is een on-line tool, waarmee het mogelijk is tevens grafieken te genereren, op basis van de daartoe beschikbaar gestelde meetgegevens. Deze meetgegevens zijn voor het geluid, het licht (lux) en de magnetometer verkregen met de Physics Toolbox Pro [48], waarmee de gegevens te exporteren zijn naar een CSV-bestand, geschikt voor de on-line Plotly-tool.

De overige sensorwaarden, uit tabel vier van verschillende sensoren, zijn verkregen met de Falcon SQL Client [49], een open-source SQL client. Deze tool biedt dezelfde vergelijkbare functionaliteiten als de on-line Plotly-tool. De Falcon SQL Client wordt gebruikt om SQLite-dataverzamelingen te visualiseren, waarbij het gemiddelde te berekenen is met behulp van SQLite. Deze SQLite-dataverzamelingen bevatten sensormeetwaarden, die niet met Physics Toolbox Pro te verkrijgen zijn. Dit houdt in dat we een onderzoekstoepassing op een smartphone gebruiken, om deze SQLite-dataverzamelingen te verkrijgen. Zo is deze onderzoekstoepassing een voorloper van een toepassing, die we in het verdere verloop van dit exploratieve onderzoek uitbouwen, om een WiFi-module aan of uit te schakelen.

Met behulp van een logische bewegingssensor (§ 6.4.5.) bleek voor de contexten brommerrijden, fietsen, wandelen en geen beweging dat het slagingspercentage al snel boven de 50% ligt. Dit bleek na het aanmaken van enkele training-examples, voor elk van deze contexten en de daaropvolgende contextclassificatie. Zo geldt voor de context waarbinnen geen beweging plaatsvindt dat het slagingspercentage rond de 100% ligt. Hetzelfde geldt voor de context wandelen, terwijl dit voor de contexten fietsen en brommerrijden iets lager ligt, afhankelijk van het rijgedrag en het gekozen vervoersmiddel.

De rotation-vector-sensor is een logische sensor die gebruik maakt van een gyroscoop. Met behulp van een rotation-vector-sensor is het daardoor mogelijk de positie van een smart-phone te verkrijgen na de draaiing om zijn assen. Doordat een smartphone-gyroscoop terug naar nul loopt wanneer een draaiing heeft plaatsgevonden is dit met een smartphone-gyroscoop alleen niet mogelijk (§ 6.4.1.). Zo geldt voor de rotation-vectorsensor dat deze een onderscheid kan maken tussen verschillende contexten. Dit onder de voorwaarde dat een gebruiker een smartphone binnenshuis plat neerlegt, terwijl hij deze smartphone buitenshuis verticaal meedraagt, bijvoorbeeld in zijn borstzak.

De proximity-sensor is een binaire sensor. Voor een proximity-sensor geldt eveneens dat deze een onderscheid kan maken tussen verschillende contexten. Dit onder de voorwaarde dat een gebruiker een smartphone binnenshuis plat neerlegt, terwijl hij deze smartphone buitenshuis opgeborgen meedraagt, bijvoorbeeld in een etui of in een broekzak (§ 6.4.1.).

Nadat we bepaald hebben welke sensoren geschikt zijn voor het onderzoek worden de exploratieve onderzoeksmethoden beschreven in hoofdstuk acht, waarna de resultaten worden besproken hoofdstuk negen. Om dit mogelijk te maken wordt in het volgende hoofdstuk beknopt de architectuur van de gehanteerde machine-learning-toepassing besproken.

7. De Architectuur

In dit hoofdstuk vatten we de architectuur van de machine-learning-toepassing op een Samsung Galaxy S9 kort samen. Deze toepassing wordt voor het Android-platform uitgevoerd en in drie lagen opgebouwd:

1. De presentatielaag;
2. De logic-layer;
3. De datalaag.

Binnen dit hoofdstuk bespreken we de verschillende lagen in verschillende paragrafen, in chronologische volgorde. Vanwege de eenvoudige architectuur van een passive-scanning-app wordt deze niet behandeld. De programmatuur van de machine-learning-toepassing en de passive-scanning-polling-app zijn beide bij de bijlagen van dit onderzoek te vinden.

7.1. De presentatielaag

De presentatielaag bestaat uit de grafische user-interface, die uit verschillende fragmenten is opgebouwd. Elk van deze fragmenten, die geselecteerd kunnen worden, presenteren verschillende functionaliteiten aan een gebruiker of een onderzoeker. Zo onderscheiden we een fragment, waarmee het mogelijk is om nieuwe contexten te classificeren met een bepaalde scanfrequentie, en een fragment waarmee het mogelijk is nieuwe training-examples in een training-set op te nemen.

7.1.1. Het scanfragment

Met behulp van het scanfragment is het mogelijk nieuwe contexten te classificeren, volgens een bepaalde frequentie, binnen een korte periode. Daartoe wordt de logic-layer aangesproken, waarmee de verschillende sensoren gebruikt worden. Zo worden verschillende sensorwaarden in real-time weergegeven binnen het scanfragment in de achtergrond, terwijl de uitslag van een contextclassificatie op de voorgrond gepresenteerd wordt. De classificatie-uitslag vinden we binnen het bereik tussen nul en één, analoog aan het bereik tussen nul en honderd procent.

Wanneer een contextclassificatie gemaakt is, kunnen we deze op de juistheid hiervan beoordelen. Op het moment dat er een contextclassificatie gemaakt wordt, worden aan de onderzoeker twee knoppen gepresenteerd, één OK-knop en één NO-knop. Daarmee is de gebruiker in staat te beoordelen of de contextclassificatie correct is uitgevoerd of niet, waarna deze beoordeling in een dataverzameling opgenomen wordt (§ 8.3.1.).

7.1.2. Het training-example-fragment

Met behulp van het training-example-fragment is het mogelijk een nieuwe training-example aan de training-set toe te voegen. Daartoe wordt een nieuwe context met behulp van verschillende sensoren opgenomen in de logic-layer, gedurende een korte periode, bijvoorbeeld tien seconden. Vervolgens is men in staat een omschrijving van de nieuwe context in te geven, waaronder onder meer: vertrouwde of onvertrouwde context, in een openbare of een niet-openbare omgeving en een eigen vrije tekstuele omschrijving. Deze gegevens worden in een dataverzameling opgenomen.

7.2. De logic-layer

Binnen de logic-layer vinden de processen plaats die verantwoordelijk zijn voor de bewerkingen op de data die verkregen is. Zo worden de sensormeetwaarden gemiddeld of wordt er op deze meetwaarden de moving-median-methode (§ 6.2.2. en § 6.2.3.) toegepast, afhankelijk van het sensortype. Om dit mogelijk te maken wordt er van een ConcurrentHashMap gebruik gemaakt, waardoor de functionaliteit van het scanfragment grotendeels aanwezig blijft, wanneer van het

training-example-fragment gebruik gemaakt wordt. Dit houdt in dat de machine-learning-toepassing in staat blijft een context te beoordelen, tijdens het proces om een nieuwe context aan een training-set toe te voegen.

Met de functionaliteiten, die door het scanfragment en het training-example-fragment geboden worden, wordt er van een zelfde verzameling sensormethoden gebruik gemaakt. Deze verzameling noemen we de *sensor-handler* van de machine-learning-toepassing.

7.2.1. Het scannen voor een nieuwe training-example

Om een nieuwe context als een training-examples in een training-set toe te voegen wordt er van het training-example-fragment (§ 7.1.2.) gebruik gemaakt. Na het toevoegen van een training-set wordt de nieuwe verzameling theta's berekend, over de waarden in een training-set, en weggeschreven naar een dataverzameling. Daarbij wordt van kernel-functies (§ 4.2.6. tot en met § 4.2.8.) gebruik gemaakt, in combinatie met Newton's logistische regressie.

Dit houdt in dat er vele matrixberekening toegepast worden. Daartoe wordt er van een open-source-library, de Universal Java Matrix Package [46], gebruik gemaakt, te verkrijgen via "https://ujmp.de/". Voor het samenstellen van een nieuwe verzameling theta's wordt dit binnen een package in de programmatuur gevat, in de zogenaamde Training-package.

Tijdens het scannen voor een nieuwe training-example wordt er ook data verzameld van sensoren, die in eerste aanleg niet noodzakelijk is voor het uitvoeren van het onderzoek en waarvan geen gebruik gemaakt wordt om de theta's vast te stellen. Dit komt doordat het in theorie mogelijk is dat er correlaties bestaan tussen de meetwaarden van sensoren, waarvan we niet zonder meer aannemen dat deze voor de classificatie van een context van belang zijn, binnen dit onderzoek (§ 8.6.2.).

7.2.2. Het scannen voor WiFi-module aan of uit

Om een WiFi-module aan of uit te schakelen moet een context geclassificeerd worden, door van het scanfragment (§ 7.1.1.) gebruik te maken. Daartoe wordt de verzameling vastgestelde theta's gebruikt.

7.3. De datalaag

De datalaag is verantwoordelijk voor het bijhouden van alle dataverzamelingen. Daartoe wordt een SQLite-database gebruikt, die door een databasemanager in de programmatuur gehanteerd wordt en standaard met het Android-smartphone-platform meegeleverd wordt. Daardoor is het mogelijk sensorgerelateerde data op te slaan, onder vermelding van het sensortypenummer en de daarbij horende onderverdeelde indexnummers, welke alle door het Android-smartphone-platform vastgesteld zijn. Doordat de verwerking op deze wijze mogelijk is, is het mogelijk nieuwe sensoren aan een machine-learning-toepassing toe te voegen of te verwijderen, op een vereenvoudigde wijze, om een WiFi-module aan of uit te schakelen.

8. Exploratieve Onderzoeksmethoden

Om de connectivity en de traceability (hoofdstuk vijf) van een machine-learning-toepassing en een passive-scanning-polling-app te bepalen is het nodig de verschillende methoden, waarmee het onderzoek wordt verricht, vast te stellen. Met behulp van deze methoden moet het mogelijk zijn een betrouwbare uitspraak te doen over de verschillende aspecten die onderzocht worden. Daartoe hebben we een vertrouwde context omschreven als een omgeving, waarvan een gebruiker aanneemt dat zich daarbinnen geen WiFi-tracking-access-point bevindt. Met behulp van deze aanname wordt een machine-learning-toepassing getraind.

Zo wordt de contextbegrenzing door een machine-learning-toepassing geleerd, maar niet door een passive-scanning-polling-app. Daardoor verwachten we dat machine-learning een zinvolle manier is om passive WiFi-tracking tegen te gaan en onder bepaalde condities daarin beter slaagt dan een passive-scanning-polling-app. Deze condities beschrijven de situaties waarin een contextclassificatie gemaakt wordt. De beschrijving en de validatie hiervan beschrijven we in de paragrafen 8.2. en 8.3.

Zo behandelen we in het eerste gedeelte van dit hoofdstuk de verschillende aspecten, waarmee tijdens het trainen van een machine-learning-oplossing rekening gehouden wordt. Het gaat hier over de wijze waarop meetdata, waarmee contexten beschreven worden, wordt verzameld. Vervolgens wordt behandeld welke situaties we onderzoeken in paragraaf 8.2. en hoe de classificaties, na het voltooien van de trainingsperiode, worden vastgelegd in paragraaf 8.3. Daarna bespreken we hoe de classificatiedata van een passive-scanning-polling-app wordt verzameld. In de tweede helft van dit hoofdstuk vinden we vervolgens een gedeelte over de gemeten energie, die wordt verbruikt voor beide toepassingen, en een stukje over de praktische onderzoeks-setup. Uiteindelijk worden de onderzoeksresultaten gepresenteerd in het volgende hoofdstuk, hoofdstuk negen. Tot slot volgen de conclusies en de aanbevelingen naar aanleiding van het onderzoek in hoofdstuk tien.

8.1. Het trainen van een machine-learning-oplossing

8.1.1. Het bijhouden van de contexten

Een machine-learning-oplossing bepaalt de kans waarop een waargenomen context vertrouwd is. Dit impliceert dat hij ook kennis moet hebben van beschreven contexten die niet vertrouwd worden, want anders zal elke waargenomen context als vertrouwd worden geclassificeerd. Deze contexten worden beschreven in training-examples. De training-examples in een training-set bevatten momentopnamen van verschillende sensoren, die elk hun eigen contexten omschrijven. Om de gewenste werking van de machine-learning-oplossing te verifiëren dienen we daarom bij elke nieuw toe te voegen training-example een beschrijving op te nemen. Deze beschrijving dient minimaal een beschrijving te zijn van de context, die als een trainings-example wordt opgenomen met de bijbehorende gewenste classificatie, WiFi-module aan of WiFi-module uit. Dit dient overeen te komen met de situatie die we in de validatiefase willen verifiëren. In de gevallen waarin een fysieke omgeving buitenshuis in een training-example wordt gevat kan de bijbehorende omschrijving worden aangevuld met GPS-coördinaten. Dit komt doordat GPS alleen buitenshuis op een open vlakte optimaal werkt.

Om het trainen van een machine-learning-oplossing te vergemakkelijken of te versnellen hanteren we de vuistregel dat op openbare plekken de context niet vertrouwd wordt. Dit geldt niet wanneer bij de omschrijving in een training-set uitvoerig is opgenomen dat dit voor bepaalde openbare plekken wel geldt. Op openbare plekken dragen we de smartphone in onze borst -of broekzak of we bergen hem op in een etui, terwijl dit voor niet-openbare plekken in de regel niet geldt. Voor

uitzonderlijke situaties, waarin een smartphone in een openbare omgeving gebruikt wordt, dienen er één of meerdere training-examples met een duidelijke beschrijving te worden gemaakt.

8.1.2. Tegenstrijdige training-examples

Een training-set kan tegenstrijdige informatie bevatten, in verschillende training-examples, om een willekeurige context te classificeren. Deze tegenstrijdigheid treedt op wanneer met gebruikmaking van dezelfde feature-sensorwaarden in de ene training-example de gewenste classificatie WiFi-module aan wordt genoteerd, terwijl voor de andere training-example deze classificatie niet geldt. Om problemen, die uit tegenstrijdige training-example-informatie voortvloeien, zoveel mogelijk te beperken wordt er gebruik gemaakt van machine-learning. Daartoe worden alle training-examples bijgehouden. Deze zijn als bijlage bij dit onderzoeksverslag te vinden.

8.1.3. Leren omgaan met veranderingen

Een machine-learning-oplossing, die van zuivere logistische regressie gebruik maakt moet gedurende een geruime tijd getraind worden, om zo goed mogelijk met veranderende contexten om te leren gaan. Zo is het bijvoorbeeld mogelijk dat een machine-learning-oplossing moet leren dat een WiFi-module in de studeerkamer aan moet staan, ook wanneer er geen licht in deze kamer aanwezig is. Dit is het geval wanneer de rol die aan het licht toegekend is te onderscheidend is, volgens een hypothese (hoofdstuk vier), waardoor de WiFi-module in de studeerkamer uit is op het moment dat er geen licht binnenvalt.

Tijdens de trainingsperiode van een machine.learning-oplossing, waarin trainings-examples worden verzameld, is dit te ondervangen door één of meerdere nieuwe training-examples in de training-set op te nemen. Dit doen we met training-examples die bij de context “studeerkamer & geen licht” horen, totdat een machine-learning-oplossing correct reageert en de WiFi-module aan zet. Dit doet hij met behulp van een nieuw opgestelde hypothesefunctie, waarmee de kans wordt berekend op het aanzetten van de WiFi-module over de beschikbare training-examples. Wanneer een machine-learning-oplossing geen fouten maakt en hij voor alle expliciet vertrouwde contexten de WiFi-module aan zet, betekent dit dat de uitslag van de kansberekening binnen deze vertrouwde contexten altijd meer dan 50% bedraagt. Daarbuiten geldt dit niet (hoofdstuk vier). Zo worden er na het voltooien van de trainingsperiode er geen nieuwe training-examples toegevoegd en volgt de validatiefase van de machine-learning-oplossing. Zo weet een machine-learning-oplossing, na een trainingsperiode, zelf met veranderingen om te gaan.

8.1.4. Besluiteloosheid van de machine-learning-oplossing

De kans op WiFi-module aan of WiFi-module uit is in theorie exact 50% wanneer de machine-learning-oplossing geen enkel idee heeft of de WiFi-module aan of uit moet worden geschakeld. Dit impliceert dat de machine-learning-oplossing over onvoldoende relevante training-examples beschikt om een zinnige inschatting te maken. In dit geval wordt de waargenomen context niet expliciet vertrouwd, waardoor de WiFi-module dient te worden uitgeschakeld. Dit probleem van besluiteloosheid is te ondervangen door één of meerdere nieuwe training-examples over de context in kwestie in de training-set op te nemen, tijdens de trainingsperiode van een machine-learning-oplossing, daarbuiten niet. Zo volgt de validatie na de trainingsperiode, waarin een machine-learning-oplossing zelf zijn keuzes maakt.

Na training blijkt, vanwege afrondingen, die ver achter de komma plaatsvinden, bijvoorbeeld door het gebruik van Newton’s logistische regressiemethode (§ 4.2.4.), dat de theoretische kans van exact 50% (praktisch) nooit gehaald wordt. Daardoor is er besloten tijdens de context-classificatie, WiFi-module aan of WiFi-module uit, het resultaat van de kansberekening tot op één cijfer nauwkeurig achter de komma af te ronden. Dit betekent een afronding tot op een tiende procent.

8.1.5. De variantie van de machine-learning-oplossing

Voor de variantie (σ^2) mogen we zelf een waarde kiezen, doordat σ^2 een vrije parameter is (§ 4.2.7.). Dit houdt in dat we de variantie variëren op het moment dat er veel false-positives of false-negatives plaatsvinden, gedurende de trainingsperiode van een machine-learning-oplossing, om te bezien of het daarmee beter gaat. Zo starten we het onderzoek met $\sigma^2 = 1$, gedurende de trainingsperiode van een machine-learning-oplossing. Het resultaat van de ingestelde variantie vermelden we in hoofdstuk negen bij de verzamelde onderzoeksdata tijdens de validatiefase.

8.1.6. Het voltooien van de trainingsperiode

Een machine-learning-toepassing is nooit uitgeleerd, doordat er altijd nieuwe training-examples aan de training-set kunnen worden toegevoegd. Dit houdt in dat er een plafond aan het aantal te verzamelen training-examples, die tijdens de trainingsperiode verzameld worden, moet worden aangebracht. Deze werd bepaald door de tijd die we redelijkerwijs voor het voltooien van het onderzoek ter beschikking hebben. Zo hebben we deze grens gelegd bij honderd verschillende contexten en hun training-examples.

8.2. De onderzoekssituaties

Gezien de beperkte middelen en de projectdoorlooptijd die ons ter beschikking staan, kiezen we enkele situaties voor ons verkennend onderzoek. Daarmee proberen we aan te tonen dat machine-learning een levensvatbare manier is om passive WiFi-tracking tegen te gaan. Omgekeerd proberen we aan te tonen dat machine-learning niet zinvol is om passive WiFi-tracking tegen te gaan, wanneer een passive-scanning-polling-app in deze situaties hier beter in slaagt.

In de volgende paragrafen beschrijven we enkele situaties waarbinnen er sprake is van een vertrouwde context en de achtergronden daarvan. Doordat we bezig zijn met een verkennend onderzoek, waarmee we proberen aan te tonen dat machine-learning een levensvatbare manier is om passive WiFi-tracking tegen te gaan, onderzoeken we deze situaties eerst elk apart. Dit doen we door voor elke situatie een eigen training-set op te stellen, gedurende een trainingsperiode.

Om een meer algemener beeld te krijgen voegen we de training-sets, die bij deze onderzoeks-situaties horen, daarna samen, waarna een nieuwe trainingsperiode volgt. Daardoor verwachten we dat eventuele tegenstrijdigheden, die niet tot het juist functioneren van een machine-learning-toepassing leiden, worden opgelost of tegengegaan. De genoemde training-sets zijn als bijlage bij dit onderzoeksverslag te vinden.

Zo verwachten we bijvoorbeeld dat verschillende lichtspectra onderscheidend zijn voor de verschillende contexten, die we bij de verschillende onderzoekssituaties vinden. Daartoe gebruiken we verschillende sensoren, waaronder bijvoorbeeld de lichtmeter in Lux en de RGB -en IR-lichtsensor in candela (hoofdstuk zes).

8.2.1. Een eengezinsrijtjeshuis, de eigen woning

Binnenshuis is de context vertrouwd. Dit geldt ook in de voor -en achtertuin van een rijtjeshuis. Op openbare plekken dragen we de smartphone in onze borst -of broekzak of we bergen hem op in een etui, terwijl dit voor niet-openbare plekken in de regel niet geldt (§ 8.1.1.). Naar verwachting zal het onderscheid met de openbare ruimte daardoor te maken zijn door een machine-learning-toepassing. Zo worden de situaties buiten de vertrouwde context van het eigen woonperceel niet vertrouwd.

Sensoren zullen in een bepaalde mate bijdragen aan het onderscheid en de classificaties die gemaakt worden, naar aanleiding van het geleerde door een machine-learning-toepassing (hoofdstukken vijf en zes). Zo kunnen we bijvoorbeeld aan de lichtspectra denken, die op verschillende locaties anders

is. In welke mate een sensor precies bijdraagt aan het onderscheid is daarentegen niet te voorspellen. Dit hangt onder meer af van een machine-learning-hyothese, die door het trainen van een toepassing tot stand wordt gebracht (hoofdstuk vier). Zo verwachten we bijvoorbeeld dat de verschillende lichtspectra van verschillende contexten bijdragen aan het onderscheidend vermogen tussen deze contexten.

8.2.2. Binnen één bepaalde andere woning

Voor een woning op één bepaalde andere locatie geldt dat binnenshuis als een niet-vertrouwde context wordt aangemerkt, binnen een niet-openbare omgeving. Dit is de tweede onderzoeks-situatie. Dit houdt in dat de verschillende ruimten in dit huis eveneens gekenmerkt worden door onder meer het binnenvallende lichtspectrum van zonlicht en kunstverlichting met een bepaalde sterkte van het licht. Zo verwachten we dat de context als niet-vertrouwd wordt geclassificeerd door een machine-learning-toepassing.

Doordat de omgeving binnen en buiten deze onderzoekssituatie niet vertrouwd wordt kunnen we in theorie volstaan met het toevoegen van één enkele training-example aan een lege training-set (§ 8.1.1.). In de praktijk kiezen we er echter voor ongeveer evenveel training-examples aan een lege training-set toe te voegen, als in de voorgaande onderzoekssituatie van de eigen woning. Dit komt doordat we voor de vierde onderzoekssituatie de verkregen training-sets van de voorgaande drie onderzoekssituaties samenvoegen.

Zo komen in de training-set van de tweede onderzoekssituatie er geen training-examples voor, waarvan de contextclassificatie vertrouwd is opgenomen. Zo blijkt voor de tweede onderzoeks-situatie dat de machine-learning-oplossing geen fouten maakt.

In de voor -en achtertuin, op het woonperceel van de woning binnen de tweede onderzoekssituatie, dragen we onze smartphone in onze borst -of broekzak of we bergen hem op in een etui. Zo verwachten we dat op het woonperceel van deze woning de context als niet-vertrouwd wordt geclassificeerd.

8.2.3. Binnen een WiFi-netwerk dat van dezelfde SSID's gebruik maakt

Wanneer het mogelijk is het onderscheid tussen vertrouwde en niet-vertrouwde contexten binnen -en buitenshuis te maken, zal dit naar verwachting ook binnen een WiFi-netwerk mogelijk zijn met access-points die dezelfde SSID uitzenden. Daarmee gaan we het risico op WiFi-tracking tegen op de momenten dat we de context niet vertrouwen, doordat we menen dat er binnen de dekking van dit WiFi-netwerk er één of meerdere niet-vertrouwde WiFi-tracking-access-points te vinden zijn (hoofdstuk vijf).

Zo gebruiken we voor het exploratieve onderzoek een universitair ziekenhuis in Rotterdam, het Erasmus MC. Dit is een openbare gelegenheid, met uitzondering van de gedeelten waarin bekend mag verondersteld dat bezoekers daar niet zonder meer mogen komen. Daardoor dragen we onze smartphone doorgaans in onze borst -of broekzak mee, tenzij we de omgevingscontext expliciet vertrouwen.

Het Erasmus MC beschikt over een WiFi-netwerk, die voor bezoekers op een gemakkelijke manier toegankelijk is. Het Erasmus MC bestaat uit vele gebouwen, die onderling verbonden zijn en door dit WiFi-netwerk gedekt worden. Om dit mogelijk te maken bestaat dit netwerk uit verschillende WiFi-access-points, die dezelfde SSID uitzenden. Daardoor is het mogelijk dat binnen de dekking van dit WiFi-netwerk er een clandestiene WiFi-tracking-access-point opgesteld staat.

Voor ons exploratieve onderzoek nemen we aan dat de volgende openbare ruimten in het Erasmus MC expliciet vertrouwd worden:

- Restaurant KADE 80;
- Restaurant Garden;
- Starbucks;
- Albert Heijn;
- AKO, boekhandel & snuisterijen.

Andere openbare ruimten binnen het Erasmus MC worden niet vertrouwd en niet-openbare ruimten binnen het Erasmus MC worden niet in dit exploratieve onderzoek meegenomen. Dit houdt in dat er over niet-openbare ruimten er geen training-examples aan een training-set worden toegevoegd. Zo worden onder meer de wandelgangen aangeduid als niet-vertrouwde openbare ruimten.

8.3. Het verzamelen en vastleggen van machine-learning-onderzoeksdata

In deze paragraaf beschrijven we welke data wordt verzameld en hoe deze wordt vastgelegd, nadat de trainingsperiode van een machine-learning-toepassing is afgerond. Nadat de trainingsperiode is afgerond dient de machine-learning-oplossing te worden gevalideerd. Dit houdt in dat bekeken wordt in welke mate een machine-learning-oplossing zelfstandig de juiste contextclassificaties maakt. Er moet worden bepaald hoe goed een machine-learning-toepassing is in het herkennen van de situaties met de bijbehorende classificaties, die in de training-set voor komen. Dit is de training-set-score. De-training-set-score wordt door de werkelijke traceability, de bruikbare traceability en de connectivity (§ 5.1.2. en § 5.2.) van een machine-learning-oplossing uitgedrukt.

8.3.1. De mogelijke contextclassificaties

Met het geleerde uit de verschillende training-examples maakt een machine-learning-oplossing verschillende contextclassificaties. Dit ook in de situaties die niet expliciet in een training-example beschreven zijn met behulp van inschattingen, naar aanleiding van het geleerde uit een training-set. Tijdens het gebruik zijn deze situaties de ingeschatte contexten, de true-positives, de true-negatives en de false-positives/negatives. Dit impliceert dat we op elk moment tijdens de validatiefase van een machine-learning-oplossing goed bijhouden wat de door deze oplossing gemaakte context-classificaties zijn.

Met betrekking tot het tegengaan van passive WiFi-tracking geldt voor een machine-learning-oplossing:

- Dat een false-positives (FP) een situatie is waarin een WiFi-module aangeschakeld staat, terwijl dit niet mag;

- Dat een false-negative (FN) een situatie is waarin een WiFi-module uitgeschakeld staat, terwijl dit niet mag;

- Dat een true-positive (TP) een situatie is waarin een WiFi-module aangeschakeld staat, op het moment waarop dit moet;

- Dat een true-negative (TN) een situatie is waarin een WiFi-module uitgeschakeld staat, op het moment waarop dit moet.

Dit kunnen we voor elke context, die onderweg aangedaan wordt door een getrainde machine-laerning-oplossing, beoordelen en opnemen in een dataverzameling. Daardoor is het mogelijk om tot een training-set-score te komen.

Om te voorkomen dat we een contextclassificatie missen, die we op zijn waarde moeten beoordelen, is de machine-learning-oplossing voorzien van een trilfunctie. Dit houdt in dat de gehanteerde smartphone enkele malen kort trilt op het moment dat er een contextclassificatie gemaakt wordt. Op het moment dat er een contextclassificatie gemaakt wordt, worden aan de onderzoeker twee knoppen gepresenteerd, één OK-knop en één NO-knop. Daarmee is de gebruiker in staat te beoordelen of de contextclassificatie correct is uitgevoerd of niet, waarna deze beoordeling in een dataverzameling opgenomen wordt. Wanneer een beoordeling over een contextclassificatie gemist wordt, wordt dit eveneens in deze dataverzameling opgenomen. Daardoor kunnen we uiteindelijk iets zeggen over de kwaliteit van de verzamelde onderzoeksgegevens. Heeft een gebruiker eenmaal een contextclassificatie beoordeeld, dan verdwijnen de OK-knop en de NO-knop tot de volgende contextclassificatie.

8.3.2. Beoordeling en vastlegging contextclassificaties

Tijdens de validatiefase van een machine-learning-oplossing trilt de smartphone enkele malen, op het moment dat een WiFi-module aan of uit gezet wordt door een machine-learning-oplossing. Daardoor is het mogelijk over een contextclassificatie te boordelen of deze correct gemaakt is of niet, waarna dit vastgelegd wordt. Zo houden we op de momenten waarop dit mogelijk is de grafische user-interface van de toepassing scherp in de gaten, om het aantal gemiste beoordelingen tegen te gaan.

Het valideren geschiedt aan de hand van de omschrijvingen tussen vertrouwde en niet-vertrouwde contexten (§ 8.1.1. en § 8.2.), die de begrenzingen van de contexten in een training-set afbakenen. Daardoor is het mogelijk vast te stellen of een contextclassificatie een true-positive/negative of een false-positive/negative betreft. Deze vaststelling wordt zo goed mogelijk ingeschat en weggeschreven naar een dataverzameling. Deze dataverzameling is als bijlage bij dit onderzoeks-verslag te vinden. Daardoor zijn de false discovery rate (FDR), de true positive rate (TPR) en de accuracy vast te stellen. Deze maken het onder meer mogelijk een vergelijking met een passive-scanning-polling-app te maken, door de testscores van de verschillende toepassingen te vergelijken.

8.3.3. Herhaalbaarheid van het onderzoek op de onderzoeksdata

Van elke contextclassificatie, die met een bepaalde frequentie gemaakt wordt, wordt bijgehouden wat de sensorwaarden zijn, om tot een classificatie te komen. Van elke contextclassificatie wordt tijdens de trainingsperiode en de validatiefase van de machine-learning-oplossing tevens bijgehouden wat de gehanteerde training-examples waren om tot een hypothese te komen. Daardoor moet het mogelijk zijn het onderzoek te verifiëren met behulp van de sensorwaarden van de training-examples uit de training-set.

Wanneer met de validatiefase van de machine-learning-oplossing begonnen wordt, zal van elke contextclassificatie, die met een bepaalde frequentie gemaakt wordt, beoordeeld worden of deze correct gemaakt is of niet (§ 8.3.1.). Zo is het achteraf na te gaan met hoeveel training-examples en met welke hypothese begonnen wordt aan de validatiefase, doordat deze informatie wordt weggeschreven naar een dataverzameling. Dit wordt mogelijk gemaakt door van elke mogelijke contextclassificatiebeoordeling, die in deze dataverzameling opgeslagen wordt, bij te houden uit hoeveel opeenvolgende training-examples de hypothese is opgebouwd. Dit aantal dient daardoor voor alle mogelijke beoordelingen hetzelfde te zijn, wanneer we ons onderzoek zo goed mogelijk

uitvoeren.

8.4. Het vastleggen van passive-scanning-polling-app-onderzoeksdata

Voor het verzamelen van onderzoeksdata met een passive-scanning-polling-app is er gebruik gemaakt van een Android-app, die speciaal hiervoor ontwikkeld is. Nadat de onderzoeksresultaten van een machine-learning-oplossing verzameld en vastgelegd zijn, is het mogelijk deze te verifiëren met een passive-scanning-polling-app.

8.4.1. De mogelijke contextclassificaties

De gewenste contextclassificaties, die door een passive-scanning-polling-app gemaakt worden, zijn te herleiden uit de vastgelegde omschrijvingen van contexten, die tevens de begrenzingen tussen vertrouwde en niet-vertrouwde contexten afbakenen (§ 8.1.1. en § 8.2.). Deze worden gebruikt voor het trainen en het valideren van een machine-learning-oplossing. Wanneer de gewenste classificatie volgens de contextbegrenzingen een true-positive is, wordt er aangenomen dat er een vertrouwde WiFi-access-point aanwezig is binnen deze context.

Met betrekking tot het tegengaan van passive WiFi-tracking geldt voor een passive-scanning-polling-app:

- Dat een false-positives (FP) een situatie is waarin er frames naar een WiFi-access-point worden verzonden, terwijl dit niet mag;
- Dat een false-negative (FN) een situatie is waarin er geen frames naar een WiFi-access-point worden verzonden, terwijl dit niet mag;
- Dat een true-positive (TP) een situatie is waarin er frames naar een WiFi-access-point worden verzonden, op het moment waarop dit moet;
- Dat een true-negative (TN) een situatie is waarin er geen frames naar een WiFi-access-point worden verzonden, op het moment waarop dit moet.

Doordat een passive-scanning-polling-app een WiFi-module niet uitschakelt, zoals een machine-learning-toepassing dit wel doet, verwachten we false-positives op het moment dat we ons buiten het woonperceel van de eigen woning bevinden. Dit is tijdens het gebruik van een machine-learning-toepassing niet het geval. Daardoor is het aantal false-positives van een passive-scanning-polling-app afhankelijk van hoe vaak, hoe lang en waar we ons buiten het eigen woonperceel bevinden.

8.4.2. Mogelijk falen van WiFi-access-points

Wanneer een vertrouwde WiFi-access-point niet aanwezig is binnen een vertrouwde context, dan wordt deze hierbinnen geplaatst. Daardoor is het mogelijk dat de classificatie over een naastgelegen context een false-positive oplevert, volgens de passive-scanning-polling-app, ten opzichte van de gewenste resultaten (§ 8.2.). Zo wordt er een mobiele WiFi-access-point, een “Huawei Mobile WiFi E5330”, gebruikt. In de praktijk verwachten we echter niet dat het plaatsen van een mobiele WiFi-access-point nodig is.

Wanneer de gewenste classificatie een true-positive is en er een false-negative plaatsvindt, tijdens het gebruik van een passive-scanning-polling-app, dan moeten we dit onderzoeken. Mogelijke oorzaken zijn het falen van de hardware of een verkeerde instelling, waardoor een WiFi-access-point verborgen is en hij geen beacon-requests uitzendt (hoofdstuk twee).

8.4.3. Beoordeling en vastlegging contextclassificaties

Tijdens de validatiefase van een passive-scanning-polling-app wordt er een trilsignaal gegeven op het moment dat een WiFi-verbinding aan of uit gezet wordt door deze toepassing. Daardoor is het mogelijk bij elke veranderende classificatie te boordelen of deze correct gemaakt is of niet; hetgeen vastgelegd wordt. Dit geschiedt op dezelfde wijze als voor een machine-learning-oplossing geldt (§ 8.3.1.). Daardoor is het mogelijk de false discovery rate (FDR) de true positive rate (TPR) en de accuracy over alle classificaties tezamen vast te stellen voor een passive-scanning-polling-app. Daarmee is het mogelijk een vergelijking tussen beide toepassingen te maken.

8.5. Het energieverbruik van beide toepassingen

Tijdens het valideren van een passive-scanning-polling-app en een machine-learning-toepassing wordt het energieverbruik bijgehouden voor elk van deze toepassingen. Het energieverbruik is één van de attributen waarmee inzicht in de connectivity kan worden verkregen naast de traceability.

8.5.1. Het bijhouden van het energieverbruik

Tijdens de validatiefase wordt gebruik gemaakt van een app, waardoor het mogelijk is inzicht te verkrijgen over de apps die actief zijn op een smartphone en hoeveel energie deze verbruiken. Zo is het noodzakelijk alle niet-noodzakelijke toepassingen af te sluiten, waardoor het energieverbruik van een machine-learning-oplossing of een passive-scanning-polling-app zo nauwkeurig mogelijk wordt bijgehouden.

Voordat met de validatie wordt begonnen wordt de tijd en de actuele energievoorraad genoteerd. Ditzelfde geldt wanneer met de validatie wordt geëindigd. Daardoor is het mogelijk een reële inschatting te maken van het energieverbruik van de machine-learning-toepassing, ten opzichte van een passive-scanning-polling-app.

8.5.2. Scanfrequentie & energieverbruik machine-learning-toepassing

Een machine-learning-toepassing leest met een bepaalde frequentie de sensorwaarden uit om een contextclassificatie te maken. Deze frequentie is in te stellen. Daardoor wordt geprobeerd deze frequentie zo laag mogelijk te krijgen, zonder dat er veel op de bruikbaarheid wordt ingeboet. Dit houdt in dat het verzamelen en vastleggen van machine-learning-onderzoeksdata minimaal enkele malen geheel of gedeeltelijk wordt uitgevoerd op verschillende frequenties voor één onderzoekssituatie. Hier wordt gekozen voor de eengezinsrijtjeshuis, de eigen woning, eventueel aangevuld met één of twee andere onderzoekssituaties, welke doorlopen worden.

Zo verwachten we dat dit voor de context brommerrijden (§ 6.4.2.) bijvoorbeeld anders zal zijn. Dit komt doordat we verwachten dat op de brommer of in een ander voertuig de contextclassificatie sneller moet plaatsvinden dan dat voor lopen of wandelen het geval is. Gezien de middelen en de projectdoorlooptijd die ons ter beschikking staat, kiezen we minimaal voor een eengezinsrijtjeshuis, welke doorlopen wordt om de frequentie te bepalen, waarmee contextclassificaties gemaakt worden.

8.6. De experimentele hypothese

Vanuit een theoretisch perspectief is het mogelijk een voorspelling te doen over het verloop en het resultaat van een experiment; hetgeen verwoord wordt door een hypothese. Een experimentele hypothese beschrijft onze verwachtingen over een experiment, waartoe een aantal training-examples nodig zijn. Zo is het de verwachting dat een machine-learning-oplossing steeds betere

voorspellingen kan maken, naar mate er meer training-examples beschikbaar worden gesteld. Zo is een machine-learning-oplossing in theorie nooit uitgeleerd.

Dit houdt in dat we vanuit een theoretisch perspectief moeten onderbouwen hoeveel training-examples we minimaal nodig hebben om tot een positief onderzoeksresultaat te komen. Dit onderzoeken we voor de verschillende onderzoekssituaties (§ 8.2.). Daarbij dienen we rekening te houden met het volgende:

- Een contextclassificatie, welke door een machine-learning-toepassing gemaakt wordt, is afhankelijk van een logistische regressiehypothese.

- De uitkomst van een logistische regressiehypothese is afhankelijk van de actuele sensorwaarden, de vastgelegde sensorwaarden uit een training-set en de daarin voorkomende classificaties, welke door een gebruiker ingegeven zijn.

- Sensorwaarden kunnen elk moment verschillen, ook binnen een willekeurige context, terwijl we geen exacte voorkennis hebben over deze sensorwaarden.

- Alleen de contextclassificaties, welke door een onderzoeker of een gebruiker in een training-set ingegeven worden en waarbij we aannemen dat zich daarbij geen beoordelingsfouten voordoen, zijn consequent en voorspelbaar.

Daardoor kunnen we alleen een voorspelling doen over het minimaal benodigde aantal training-examples en de tijd die gemoeid is om deze te verkrijgen, wanneer we met het trainen van een machine-learning-oplossing beginnen. Door tijdens het trainen de behaalde resultaten uit te zetten tegen de verstreken tijd en het aantal toegevoegde training-examples moet het mogelijk zijn de progressie van een machine-learning-oplossing inzichtelijk te maken. Dit doen we door de verhouding tussen het aantal foute contextclassificaties en het aantal beschikbare training-examples op verschillende momenten vast te stellen. Daarmee geraken we steeds dichter bij een punt, waarvan we steeds preciezer kunnen inschatten dat daarmee zoveel mogelijk contexten van de onderzoekssituatie juist worden geclassificeerd.

8.6.1. De progressiefunctie

Wanneer we dagelijks een onderzoekssituatie geheel doorlopen en onderweg terug verschillende training-examples toevoegen is het mogelijk het aantal classificatiefouten, in verhouding tot het aantal training-examples, te zien afnemen. Dit ten opzichte van de voorafgaande dagelijkse trainingssessies, tot een punt waarop een machine-learning-toepassing nagenoeg uitgeleerd is. Idealiter is deze progressie na een aantal trainingen te extrapoleren, op basis van het aantal training-examples in een training-set, tot een punt waarop een machine-learning-toepassing bij benadering is uitgeleerd. Wanneer dit niet zo is dienen we ons af te vragen hoe dit komt.

Doordat we hier van extrapolatie gebruik maken spreken we van een progressiefunctie. Met behulp van een progressiefunctie is het mogelijk een hypothese over een experiment op te stellen en in te schatten hoeveel trainining-examples er nodig zijn. Wanneer we dit op een juiste wijze kunnen inschatten weten we tevens hoeveel tijd hier ongeveer mee gemoeid is. Zo zal het exploratieve onderzoek niet gemakkelijk uit te voeren zijn wanneer bijvoorbeeld blijkt dat we één miljard training-examples nodig hebben om een machine-learning-toepassing correct te laten werken.

8.6.2. De invloed van het aantal sensoren op de progressiefunctie

Gedurende een aantal dagen onderzoeken we de verschillende onderzoekssituaties, waardoor we van deze situaties verschillende progressiefuncties kunnen opstellen. Tijdens het verzamelen van de sensormeetwaarden verzamelen we onder meer de meetwaarden van een geluidssensor (§ 6.3.2. en § 6.4.2.), welke alleen het geluidsniveau meet. Daarbij dienen we op te merken dat er relatief zeer veel training-examples nodig zijn om een sensor, die alleen het geluid meet, misschien te doen slagen binnen een machine-learning-oplossing. Daardoor is het mogelijk achteraf een progressie-functie op te stellen, zonder de meetwaarden van deze geluidssensor. Daartoe worden alle meetwaarden van de verschillende sensoren dagelijks opgeslagen en bijgehouden met behulp van training-set-back-ups.

Tijdens ons onderzoek worden alle sensormeetwaarden van alle behandelde sensoren verzameld, die voor de classificatie van een omgevingscontext relevant zijn en we op een Samsung Galaxy S9 kunnen vinden. Dit zijn sensoren waarvan we aannemen dat deze voor de contextclassificatie van belang zijn. Dit zijn de volgende sensoren: licht (lux), licht (rood, infrarood, groen en blauw in candela), de rotation-vector (gyroscoop), de binaire proximity-sensor, de battery-temperature-sensor en de zelf samengestelde synthetische bewegingssensor (§ 6.4.5.). Hierbij geldt dat het mogelijk is achteraf een progressiefunctie op te stellen, zonder de meetwaarden van één of meerdere sensoren uit deze sensorverzameling. Daartoe worden alle meetwaarden van de verschillende sensoren dagelijks opgeslagen en bijgehouden.

In theorie is het mogelijk dat er correlaties bestaan tussen de meetwaarden van sensoren, waarvan we niet zonder meer aannemen dat deze voor de classificatie van een context van belang zijn in dit onderzoek. De sensoren die de magnetische veldsterkte en het geluidsniveau meten zijn hiervan voorbeelden. Van deze sensoren worden de meetwaarden tijdens de trainingsperiode van een machine-learning-oplossing opgeslagen in een aparte dataverzameling, op het moment dat we een context opnemen in een nieuwe training-example. Daardoor is het mogelijk de sensormeetwaarden, waarvan we niet zonder meer kunnen aannemen dat deze voor de classificatie van een context van belang zijn, achteraf aan de training-set toe te voegen, tijdens dit exploratieve onderzoek. Wanneer hiervan sprake is, doen we dit tijdens de trainingsperiode van een machine-learning.oplossing, want tijdens de validatiefase van deze oplossing dient een machine-learning-oplossing zelfstandig zijn keuzes te maken. Dit betekent dat er opnieuw een progressiefunctie moet worden opgesteld, wanneer hiervan sprake is. Zo bevat een training-set de training-examples met de contextinformatie, waarvan we wel aannemen dat deze van belang zijn voor de classificatie van een context.

Tijdens de trainingsperiode van een machine-learning-oplossing worden sensoren gehanteerd en apart van de training-set opgeslagen, waarvan we niet zonder meer kunnen aannemen dat deze voor de classificatie van een context van belang zijn in dit onderzoek. Dit zijn onder meer de sensoren op een Samsung Galaxy S9, welke we in appendix B kunnen vinden.

8.7. De praktische onderzoeks-setup

Op openbare plekken dragen we de smartphone in de regel in onze borst -of broekzak of we bergen hem op in een etui, terwijl dit voor niet-openbare plekken in de regel niet geldt (§ 8.1.1.). Dit houdt in dat we geen trainings-examples kunnen toevoegen en dat we niet in staat zijn een toepassing te valideren, tenzij we dit problemen ondervangen.

We lossen dit op met een beeldschermverbinding, die met behulp van Airdroid, een software-toepassing, en een usb-verbinding tot stand wordt gebracht. Met behulp van een laptop is het

daardoor mogelijk de grafische user-interface van een Android-smartphone te besturen en af te lezen, ook wanneer deze zich in een etui, borst -of broekzak bevindt.

9. De onderzoeksresultaten

In dit hoofdstuk presenteren we de onderzoeksresultaten, naar aanleiding van de exploratieve onderzoeksmethoden, die in het vorige hoofdstuk behandeld werden. Met behulp van deze onderzoeksmethoden is het mogelijk om tot de testscores te komen van een machine-learning-toepassing en passive-scanning-polling-app, welke ten opzichte van elkaar vergeleken worden. Bij een bepaalde scanfrequentie bestaat een testscore onder meer uit:

- de false discovery rate (FDR) en/of de positive predictive value (PVV) (§ 5.1.2.);
- de true positive rate (TPR) en/of de false negative rate (FNR) (§ 5.2.);
- de nauwkeurigheid of de accuracy (§ 5.2.);
- het energieverbruik.

We hanteren de aanname dat er alleen binnen niet-expliciet vertrouwde contexten WiFi-tracking plaatsvindt. De werkelijke traceability om WiFi-tracking tegen te gaan wordt daardoor verwoord door de ratio tussen de false-positives (FP) in verhouding tot het totale aantal positives (TP + FP), de false discovery rate (FDR), volgens een machine-learning-toepassing. Dit geldt ook voor een passive-scanning-polling-app, die een WiFi-module niet uitschakelt, wanneer een context niet expliciet vertrouwd wordt. In dit geval zal dit daardoor een false-positive opleveren, wanneer er frames naar een willekeurige WiFi-access-point worden verzonden, binnen een context die niet vertrouwd wordt.

We hanteren de aanname dat er binnen vertrouwde contexten er geen WiFi-tracking plaatsvindt. De werkelijke traceability om WiFi-tracking tegen te gaan wordt hier verwoord door de ratio tussen de true-positives (TP) in verhouding tot het totale aantal positives (TP + FP), de positive predictive value (PVV). Zo zijn de PVV en de FDR zijn samen 100%, wanneer het onderzoek foutloos is, voor de machine-learning-toepassing en de passive-scanning-polling-app.

De connectivity verwoordt onder meer in welke mate een software-oplossing gebruik maakt van vertrouwde wireless acces-points, die binnen het bereik zijn van een vertrouwde omgeving. Dit wordt met de bruikbare traceabillity verwoord door de true positive rate (TPR) en/of de false negative rate (FNR). De TPR en de FNR worden beide in verhouding naar het aantal true-positives en false-negatives (TP + FN) uitgedrukt. De TPR en de FNR zijn daardoor samen 100%, wanneer het onderzoek foutloos is, voor de machine-learning-toepassing en de passive-scanning-polling-app.

Voor de nauwkeurigheid of de accuracy hanteren we de ratio tussen de true-positives plus de true-negatives, in verhouding tot alle negatives plus alle positives, volgens formule 20.

De connectivity wordt tevens verwoordt door de hoeveelheid energie die een machine-learning-toepassing en een passive-scanning-polling-app elk verbruiken. Het tijdsinterval, waarop een classificatie gemaakt wordt, kan worden ingesteld (hoofdstuk vier). Dit betekent dat een lagere scanfrequentie van een passive-scanning-polling-app zal resulteren in een lager energieverbruik.

Zo behandelen we in de eerste paragraaf de eerste onderzoekssituatie, waarin met verschillende scanfrequenties wordt gewerkt, het energieverbruik. Daartoe maken we van een Samsung Galaxy S9 gebruik, met een – voor consumenten – niet-uitneembare smartphone-batterij van drie Ampère-uur. Vervolgens presenteren we in paragraaf twee tot en met vier de onderzoeksresultaten van de overige onderzoekssituaties. Door de beperkte projectdoorlooptijd is het energieverbruik op verschillende scanfrequenties hierin niet meegenomen.

9.1. Resultaten eerste onderzoekssituatie van een eengezinsrijtjeshuis, de eigen woning

Binnen de eerste onderzoekssituatie worden alle contexten, binnen het woonperceel van een eengezinsrijtjeshuis, inclusief de bijbehorende voor -en achtertuin, vertrouwd, daarbuiten worden alle contexten niet vertrouwd. Zo verkrijgen we de volgende onderzoeksresultaten, door binnen en buiten het eigen woonperceel metingen te verrichten:

AI-toepassing	TP	FP	TN	FN	Ongecontroleerd	Totaal
Eigen woning	183	1	182	0	0	366

AI-toepassing	FDR	PVV	TPR	FNR	Accuracy	Scanfrequentie	Variantie (σ^2)
Eigen woning	0,54%	99,46%	100%	0%	99,73%	0,1 Hz	1

Tabel 4: Resultaten machine-learning-toepassing, tijdens de eerste onderzoekssituatie.

Deze resultaten zijn verkregen met een training-set die uit 35 training-examples bestaat. Hiervan hebben geen van de training-example-features van de zelf samengestelde synthetische bewegingssensor een vaste waarde voor wandelen en rennen verkregen. Dit komt doordat er tijdens de trainingsperiode niet gelopen wordt, op de momenten waarop een nieuwe training-example wordt aangemaakt, soms met een laptop in de hand.

Voor een passive-scanning-polling-app gelden de bovenstaande uitslagen niet. Dit komt doordat een passive-scanning-polling-app ook buiten het eigen woonperceel de context als vertrouwd classificeert, zolang er een vertrouwde WiFi-access-point binnen bereik is. Voor een passive-scanning-polling-app verwachten we daardoor andere resultaten te behalen, afhankelijk van waar we ons begeven. Deze resultaten zijn verkregen tijdens een steekproef:

Passive-scanning-app	TP	FP	TN	FN	Ongecontroleerd	Totaal
Eigen woning	129	61	91	0	1	282

Passive-scanning-app	FDR	PVV	TPR	FNR	Accuracy	Scanfrequentie
Eigen woning	32,1%	67,9%	100%	0%	78%	0,1 Hz

Tabel 6: Resultaten passive-scanning-polling-app, tijdens de eerste onderzoekssituatie.

Uit tabel vijf en zes concluderen we dat de FDR bij een passive-scanning-polling-app beduidend hoger is, dan bij een machine-learning-toepassing, in verhouding, waardoor deze vatbaarder is voor WiFi-tracking. Voor de machine-learning-toepassing en de passive-scanning-polling-app geldt in beide gevallen dat voor alle contexten, waarin er van een TP sprake moet zijn, een WiFi-verbinding mogelijk is. Dit concluderen we doordat de TPR in beide gevallen 100% is. Een machine-learning-toepassing kent daarentegen een hogere accuracy in deze onderzoekssituatie.

9.1.1. Het energieverbruik van de machine-learning-toepassing, in de eerste onderzoekssituatie

Om het energieverbruik van een machine-learning-toepassing op een smartphone vast te stellen tijdens de eerste onderzoekssituatie worden alle niet-noodzakelijke toepassingen afgesloten. Tijdens de verschillende onderstaande testgevallen houden we het beeldscherm zover mogelijk gedimd, door de smartphone-instellingen te gebruiken. Tijdens de verschillende testgevallen hanteren we dezelfde variantie van de machine-learning-toepassing, namelijk $\sigma^2 = 1$. Bij het starten van de energiemetingen wordt er in alle gevallen begonnen met een volledig gevulde, met de smartphone

meegeleverde standaardbatterij. Zo zijn over het energieverbruik van de machine-learning-toepassing de volgende resultaten vastgesteld op een Samsung Galaxy S9, binnen een zelfde vertrouwde omgeving:

Scanfrequentie	Starttijd en datum	Eindtijd en datum	Batterijverbruik	Gemiddeld batterijverbruik
0,1 Hz	04-03-2019 20:36u	05-03-2019 07:25u	51%	0,078582% per min.
0,2 Hz	05-03-2019 20:24u	06-03-2019 07:31u	52%	0,077961% per min.
0,5 Hz	07-03-2019 20:26u	08-03-2019 07:33u	51%	0,076692% per min.

Tabel 7: Onderzoeksgegevens, waarmee het gemiddelde energie -of batterijverbruik per minuut is te herleiden van een machine-learning-toepassing.

Hier valt op dat het energieverbruik op de verschillende frequenties weinig uiteen loopt. Naar aanleiding van literatuuronderzoek nemen we aan dat dit door de verschillende kenmerken van continuous sensoren en on-change sensoren komt. Zo staan continuous sensoren altijd aan, ongeacht de vraag of een toepassing hiervan gebruik maakt, terwijl on-change sensoren alleen reageren of aan gaan wanneer er een verandering plaats vindt (§ 6.2.).

9.1.2. Het energieverbruik van de passive-scanning-polling-app, in de eerste onderzoekssituatie

Om inzicht te verkrijgen in het energieverbruik van een passive-scanning-polling-app moet de scanfrequentie van een dergelijke toepassing bekend of instelbaar zijn. Zo is er gekozen voor een zelfontwikkelde passive-scanning-polling-app, die op verschillende scanfrequenties kan worden ingeregeld. Zo gelden tijdens de verschillende onderstaande testgevallen dezelfde overige testcriteria, als voor de machine-learning-toepassing, om het energieverbruik vast te stellen, op σ^2 na. Daardoor is het mogelijk de uitslagen over de experimenten met elkaar te vergelijken.

Scanfrequentie	Starttijd en datum	Eindtijd en datum	Batterijverbruik	Gemiddeld batterijverbruik
0,1 Hz	17-03-2019 20:03u	18-03-2019 07:53u	66%	0,092958% per min.
0,2 Hz	19-03-2019 19:59u	20-03-2019 07:39u	83%	0,118571% per min.
0,5 Hz	20-03-2019 20:13u	21-03-2019 07:45u	90%	0,130057% per min.

Tabel 8: Onderzoeksgegevens, waarmee het gemiddelde energie -of batterijverbruik per minuut is te herleiden van een passive-scanning-polling-app.

Hier valt direct op dat het energieverbruik hoger is, dan dat voor een machine-learning-toepassing het geval is. Bij een scanfrequentie van 0,1 Hz betekent dit een toename in het energieverbruik van ongeveer 18,3% per minuut, ten opzichte van een machine-learning-toepassing. Bij 0.2 Hz is dit ongeveer 52,1% per minuut, en bij een scanfrequentie van 0,5 Hz is dit ongeveer 69,6% per minuut. Tevens valt op dat het gemiddeld energie -of batterijverbruik per minuut niet-evenredig toeneemt met de scanfrequentie bij een passive-scanning-polling-app. Waardoor deze niet-evenredige toename plaatsvindt hebben we, vanwege de scope en de beschikbare projectdoorlooptijd van dit exploratieve onderzoek, niet onderzocht.

9.2. Resultaten tweede onderzoekssituatie, een andere woning

Binnen de tweede onderzoekssituatie worden alle contexten niet vertrouwd. Daardoor verwachten we dat FDR, de PVV, de TPR en de FNR alle vier gelijk aan nul procent zijn, terwijl de accuracy ongeveer 100% is. De verkregen resultaten zijn, zonder een vertrouwde omgeving aan te doen:

AI-toepassing	TN	Ongecontroleerd	Totaal
Een woning *	302	0	302

AI-toepassing	Accuracy	Scanfrequentie	Variantie (σ^2)
Een woning *	100%	0,1 Hz	1

Tabel 9: Resultaten machine-learning-toepassing, tijdens de tweede onderzoekssituatie.

*) Delen door nul is wiskundig niet mogelijk om de uitkomsten van de FDR, de PVV, de TPR en de FNR te verkrijgen. We stellen dat de uitkomsten voor de FDR, de PVV, de TPR en de FNR daardoor niet bestaan (§ 5.1.2.), doordat de aantallen TP's, FP's en FN's alle nul zijn in deze onderzoekssituatie.

Deze resultaten zijn verkregen met een training-set die uit 42 training-examples bestaat. Hiervan hebben geen van de training-example-features van de zelf samengestelde synthetische bewegingssensor een vaste waarde voor wandelen en rennen verkregen. Dit komt doordat er tijdens de trainingsperiode niet gelopen wordt op de momenten waarop een nieuwe training-example wordt aangemaakt, soms met een laptop in de hand.

Er is binnen deze onderzoekssituatie geen vertrouwde WiFi-access-point binnen bereik. Daardoor verwachtten we dat de resultaten van een passive-scanning-poling-app ongeveer gelijk zijn aan de resultaten ven een machine-learning-toepassing. De resultaten van de passive-scanning-polling-app zijn binnen de tweede onderzoekssituatie:

Passive-scanning-app	TN	Ongecontroleerd	Totaal
Een woning *	301	0	301

Passive-scanning-app	Accuracy	Scanfrequentie
Een woning *	100%	0,1 Hz

Tabel 10: Resultaten passive-scanning-polling-app, tijdens de tweede onderzoekssituatie.

*) Delen door nul is wiskundig niet mogelijk om de uitkomsten van de FDR, de PVV, de TPR en de FNR te verkrijgen. We stellen dat de uitkomsten voor de FDR, de PVV, de TPR en de FNR daardoor niet bestaan (§ 5.1.2.), doordat de aantallen TP's, FP's en FN's alle nul zijn in deze onderzoekssituatie.

Voor de machine-learning-toepassing en de passive-scanning-polling-app geldt in beide gevallen dat voor alle contexten, waarin er van een TN sprake moet zijn, een WiFi-verbinding niet mogelijk is. Dit concluderen we doordat het totaal aantal true-negatives in beide gevallen gelijk is aan het totale aantal gevalideerde metingen. In deze onderzoekssituatie scoren een machine-learning-toepassing en een passive-scanning-polling-app gelijkwaardig, wanneer het niet om het energieverbruik gaat van beide toepassingen.

9.3. Resultaten derde onderzoekssituatie, het Erasmus MC

Voor ons exploratieve onderzoek nemen we aan dat de volgende openbare ruimten in het Erasmus MC expliciet vertrouwd worden:

- Restaurant KADE 80;
- Restaurant Garden;
- Starbucks;
- Albert Heijn;
- AKO, boekhandel & snuisterijen.

Andere openbare ruimten binnen het Erasmus MC worden niet vertrouwd en niet-openbare ruimten binnen het Erasmus MC worden niet in dit exploratieve onderzoek meegenomen (§ 8.2.3.). Zo verkrijgen we de volgende onderzoeksresultaten, door binnen en buiten de vertrouwde omgevingen metingen te verrichten:

AI-toepassing	TP	FP	TN	FN	Ongecontroleerd	Totaal
Erasmus MC	227	8	316	0	1	552

AI-toepassing	FDR	PVV	TPR	FNR	Accuracy	Scanfrequentie	Variantie (σ^2)
Erasmus MC	3,4%	96,6%	100%	0%	98,4%	0,1 Hz	1

Tabel 11: Resultaten machine-learning-toepassing, tijdens de derde onderzoekssituatie.

Deze resultaten zijn verkregen met een training-set die uit 63 training-examples bestaat. Hiervan heeft één training-example-feature van de zelf samengestelde synthetische bewegingssensor de waarde 3,75 voor wandelen en rennen. Tijdens de trainingsperiode, voorafgaand aan de validatiefase, bleek echter dat een vaste waarde van 0,5 voor lopen (wandelen en rennen) tot betere resultaten leidt. Vermoedelijk komt dit doordat er tijdens de trainingsperiode en de validatiefase langzaam of niet gelopen is tussen twee classificaties, soms met een laptop in de hand. Vanwege de geringe impact op het exploratief onderzoek, in relatie tot de projectdoorlooptijd, hebben we ervoor gekozen de trainingsperiode en de validatiefase niet opnieuw te doorlopen om dit te corrigeren.

Voor een passive-scanning-polling-app gelden de bovenstaande uitslagen niet. Dit komt doordat het Erasmus MC gebruik maakt van een WiFi-netwerk die uit vele access-points bestaat, die alle dezelfde SSID uitzenden. Daardoor worden met een passive-scanning-polling-app alle ruimten binnen het Erasmus MC als vertrouwd aangemerkt, ook waar dit niet de bedoeling is. Voor een passive-scanning-polling-app verwachten we daardoor andere resultaten, binnen het Erasmus MC. Deze resultaten zijn tijdens een steekproef verkegen:

Passive-scanning-app	TP	FP	TN	FN	Ongecontroleerd	Totaal
Erasmus MC	268	218	0	0	1	487

Passive-scanning-app	FDR	PVV	TPR	FNR	Accuracy	Scanfrequentie
Erasmus MC	44,9%	55,1%	100%	0%	55%	0,1 Hz

Tabel 12: Resultaten passive-scanning-polling-app, tijdens de derde onderzoekssituatie.

Uit tabel elf en twaalf concluderen we dat de FDR bij een passive-scanning-polling-app beduidend hoger is, dan bij een machine-learning-toepassing, in verhouding, waardoor deze vatbaarder is voor WiFi-tracking. Voor de machine-learning-toepassing en de passive-scanning-polling-app geldt in beide gevallen dat voor alle contexten, waarin er van een TP sprake moet zijn, een WiFi-verbinding mogelijk is. Dit concluderen we doordat de TPR in beide gevallen 100% is. Een machine-learning-toepassing kent daarentegen een hogere accuracy in deze onderzoekssituatie.

9.4. De vierde onderzoekssituatie, over de voorgaande drie

De vierde onderzoekssituatie bestaat uit de voorgaande drie onderzoekssituaties samen. Dit wil zeggen dat de vierde onderzoekssituatie ontstaat door de training-sets van de voorgaande drie samen te voegen. De vierde onderzoekssituatie bestaat uit de samengevoegde training-sets van de volgende onderzoekssituaties:

- een eengezinsrijtjeshuis, de eigen woning (§ 9.1.);
- één andere willekeurige woning (§ 9.2.);
- het Erasmus MC, een academisch ziekenhuis (§ 9.3.).

Doordat alle contexten van de tweede onderzoekssituatie (§ 9.2.) niet vertrouwd worden en de contexten hiervan lijken op de contexten van de andere onderzoekssituaties kiezen we ervoor eerst de volgende training-sets samen te voegen:

1. die van een eengezinsrijtjeshuis, de eigen woning (§ 9.1.);
2. die van het Erasmus MC, een academisch ziekenhuis (§ 9.3.).

De onderzoeksresultaten over deze twee samengevoegde onderzoekssituaties zetten we in paragraaf 9.4.1. uiteen. In paragraaf 9.4.2. doen we dit voor alle drie de samengevoegde onderzoekssituaties. Daarmee verwachten we een conclusie te kunnen trekken, die in hoofdstuk tien besproken wordt.

9.4.1. De eigen woning en het Erasmus MC samen

Binnen deze onderzoekssituatie, waarin de onderzoekssituaties van de eigen woning en het Erasmus MC samengenomen worden, worden de contexten binnen het eigen woonperceel en bepaalde ruimten binnen het Erasmus MC vertrouwd. Daarbuiten geldt dit niet. Daarbuiten worden alle contexten niet vertrouwd. Zo verkrijgen we de volgende onderzoeksresultaten, door binnen en buiten de vertrouwde omgevingen van deze twee onderzoekssituaties metingen te verrichten:

AI-toepassing	TP	FP	TN	FN	Ongecontroleerd	Totaal
Eigen woning & Erasmus MC	244	7	289	0	1	541

AI-toepassing	FDR	PVV	TPR	FNR	Accuracy	Scanfrequentie	Variantie (σ^2)
Eigen woning & Erasmus MC	2,79%	97,21%	100%	0%	98,52%	0,1 Hz	1

Tabel 13: Resultaten machine-learning-toepassing, tijdens een onderzoekssituatie, waarin de training-sets van de eigen woning en het Erasmus MC zijn samengenomen.

Deze resultaten zijn verkregen met een training-set die uit 100 training-examples bestaat, 35 uit training-set van de onderzoekssituatie thuis (§ 9.1.), 63 van de onderzoekssituatie Erasmus MC (§ 9.3.) en twee aanvullende training-examples. Hiervan hebben geen van de training-example-features van de zelf samengestelde synthetische bewegingssensor een vaste waarde voor wandelen en rennen verkregen. Dit komt doordat er tijdens de trainingsperiode niet of nauwelijks gelopen wordt, op de momenten waarop een nieuwe training-example wordt aangemaakt, soms met een laptop in de hand.

Uit tabel dertien concluderen we dat de FDR bij een machine-learning-toepassing nog steeds beduidend laag is, waardoor deze weinig vatbaarder is voor WiFi-tracking. Voor de machine-learning-toepassing geldt dat voor alle contexten, waarin er van een TP sprake moet zijn, een WiFi-verbinding mogelijk is, doordat er een WiFi-module in deze gevallen aangeschakeld staat. Dit concluderen we doordat de TPR 100% is. Een machine-learning-toepassing kent een hoge accuracy in deze onderzoekssituatie. Deze ligt tussen de machine-learning-accuracy-waarden binnen de eigen woning, of de eerste onderzoekssituatie, en het Erasmus MC, de derde onderzoekssituatie. Van deze situaties is het gedrag van een passive-scanning-polling-app inmiddels bekend (§ 9.1. en § 9.3.).

9.4.2. De eigen woning, het Erasmus MC en een andere woning samen

Binnen deze onderzoekssituatie worden alle training-sets uit de voorgaande drie onderzoekssiuaties samengenomen. Deze zijn:

- een eengezinsrijtjeshuis, de eigen woning (§ 9.1.);
- één andere willekeurige woning (§ 9.2.);
- het Erasmus MC, een academisch ziekenhuis (§ 9.3.).

Nadat de training-sets zijn samengenomen tot één nieuwe training-set volgde er een nieuwe trainingsperiode. Daardoor verwachten we dat eventuele tegenstrijdigheden, die niet tot het juist functioneren van een machine-learning-toepassing leiden, worden opgelost of tegengegaan. Daartoe worden er nieuwe trainings-examples extra aan de samengestelde training-set toegevoegd, om foute classificaties tegen te gaan, voordat met de validatiefase begonnen wordt. Dit komt onder meer doordat alle contexten binnen de willekeurige woning van de tweede onderzoekssituatie niet worden vertrouwd. Zo worden er tijdens de validatiefase er geen trainings-examples aan een training-set toegevoegd.

Binnen deze onderzoekssituatie, waarin de onderzoekssituaties van de eigen woning en het Erasmus MC samengenomen worden, worden de contexten binnen het eigen woonperceel en bepaalde ruimten binnen het Erasmus MC vertrouwd. Hieraan wordt de onderzoekssituatie over het woonperceel van een andere woning toegevoegd, waarbinnen alle contexten niet vertrouwd worden (§ 9.2.). Door binnen en buiten de vertrouwde omgevingen van deze drie onderzoekssituaties metingen te verrichten verkrijgen we de resultaten van de laatste onderzoekssituatie. Zo verkrijgen we de volgende onderzoeksresultaten met 26 extra trainings-examples, die aan de samengestelde training-set van 140 training-examples toegevoegd worden:

AI-toepassing	TP	FP	TN	FN	Ongecontroleerd	Totaal
De onderzoeks-situaties samen	243	69	327	2	1	642

AI-toepassing	FDR	PVV	TPR	FNR	Accuracy	Scanfrequentie	Variantie (σ^2)
De onderzoeks-situaties samen	22,12%	77,88%	99,18%	0,82%	88,79%	0,1 Hz	1

Tabel 14: Resultaten machine-learning-toepassing, tijdens de laatste onderzoekssituatie, waarin de traing-sets van de eerste drie onderzoekssituaties (§ 9.1. tot en met § 9.3.) zijn samengenomen.

Uit tabel veertien concluderen we dat de FDR bij een machine-learning-toepassing nog steeds beduidend laag is, lager dan dat voor een passive-scanning-polling-app geldt in de eerste twee onderzoekssituaties. Daardoor is deze minder vatbaarder voor WiFi-tracking. Voor de machine-learning-toepassing geldt dat voor contexten, waarin er van een TP sprake moet zijn, een WiFi-verbinding haast altijd mogelijk is. Dit concluderen we doordat de TPR 99,18% is. Een machine-learning-toepassing kent een hoge accuracy in deze onderzoekssituatie. Deze ligt onder de machine-learning-accuracy-waarden binnen de eigen woning, of de eerste onderzoekssituatie, en het Erasmus MC, de derde onderzoekssituatie.

10. Conclusies en aanbevelingen

Doordat we hebben aangetoond dat het tegengaan van WiFi-tracking op een bruikbare en effectieve wijze een serieuze probleemstelling is, hebben we hier onderzoek naar gedaan. Dit komt onder meer doordat veel smartphones standaard gebruik maken van active WiFi-scanning om het energie-verbruik te minimaliseren, waardoor deze erg kwetsbaar zijn voor WiFi-tracking. Om dit aan te tonen hebben we de verschillende soorten WiFi-tracking-systemen, die van passive en active WiFi-tracking gebruik maken, uiteen gezet. Deze systemen vormen een bedreiging voor de publieke en de individuele privacy.

Na een exploratief onderzoek blijkt dat machine-learning en bepaalde smartphone-sensoren een belangrijke rol vervullen om deze strijd te winnen. Dit naast het gebruik van reeds andere bestaande tegenmaatregelen, zoals bijvoorbeeld het gebruik van een passive-scanning-polling-app, die onder bepaalde condities kwetsbaarder is. Dit komt onder meer doordat een passive-scanning-polling-app een WiFi-module nooit uit zet, waardoor het uitluisteren van WiFi-access-point-verbindingen mogelijk is.

Met behulp van Newton's logistische regressie en kernel-functies hebben we aangetoond dat machine-learning op een moderne smartphone in bepaalde situaties beter in staat is om WiFi-tracking tegen te gaan. Dit zonder daarbij afhankelijk te zijn van externe systemen, die aangevallen en gespooft kunnen worden of tracking mogelijk maken. Dit met betrekking tot de traceability en de connectivity, als maten van de bruikbaarheid en de effectiviteit.

Dit is niet alleen het geval in grote gebouwen, waar zich veel WiFi-access-points bevinden, die dezelfde SSID uitzenden en een WiFi-netwerk vormen, maar ook in andere situaties die we buitenshuis niet vertrouwen. Deze conclusie hebben we getrokken door de resultaten van een machine-learning-oplossing te vergelijken tegen die van een passive-scanning-polling-app. Daartoe maakt een machine-learning-toepassing gebruik van contextclassificatie en schakelt hij een WiFi-module op de belangrijke momenten aan of uit, terwijl dit voor een passive-scanning-polling-app niet geldt.

Om dit mogelijk te maken hebben we vastgesteld wat context is, om een machine-learning-toepassing te leren welke omgevingscontexten vertrouwd worden. Door van een machine-learning-toepassing gebruik te maken is het mogelijk met veranderlijke contexten om te gaan. Zo'n toepassing zal er tevens toe bijdragen dat de vertrouwelijke contextdata minder gevoelig zal zijn voor mogelijke aanvallers, doordat de harde locatiedata van verschillende omgevingsclassificaties in een database ontbreekt.

Uit de onderzochte onderzoekssituaties, die met een machine-learning-toepassing en een passive-scanning-polling doorlopen zijn, blijkt dat een machine-learning-toepassing op het vlak van de traceability aanmerkelijk beter scoort. Dit geldt ook voor de connectivity, doordat een smartphone gebruik maakt van energiebesparende sensoren. Uit de onderzochte onderzoekssituaties blijkt daardoor dat een machine-learning-toepassing op deze vlakken aanmerkelijk beter scoort dan een passive-scanning-polling-app.

We zijn daardoor van mening dat met de komst van meer processing-power en meer energie-besparende mogelijkheden op smartphones er met dit exploratieve onderzoek er een belangrijke stap is gezet om WiFi-tracking tegen te gaan. Doordat we hebben aangetoond dat het met machine-learning mogelijk is WiFi-tracking tegen te gaan moedigen we meer vervolgonderzoek aan, vooral op het gebied van processing-power en energiebesparende mogelijkheden op smartphones. Zo zal dit laatste het mogelijk maken betere en meer sensoren te gebruiken en/of geavanceerde machine-

learning-algoritmen toe te passen op smartphones, om WiFi-tracking tegen te gaan met een toepassing die iedereen wil gebruiken.

Voordat het zover is, is het mogelijk een passive-scanning-polling-app te gebruiken met een machine-learning-toepassing, tezamen op een smartphone, om WiFi-tracking tegen te gaan; ook in de niet onderzochte situaties. Dit komt doordat een machine-learning-toepassing een WiFi-module aan of uit zal zetten binnen en buiten de vertrouwde contexten, terwijl dit voor een passive-scanning-polling-app niet geldt.

Referenties

[1] College bescherming persoonsgegevens (13 oktober 2015), “Wifi-tracking van mobiele apparaten in en rond winkels door Bluetrace”, laatst geopend op 17 februari 2019: https://autoriteitpersoonsgegevens.nl/sites/default/files/atoms/files/rapport_db_bluetrace.pdf

[2] J. Huijbregts (3 november 2016), “Android en iOS hebben marktaandeel van 99,7 procent op smartphonemarkt”, laatst geopend op 17 februari 2019: https://tweakers.net/nieuws/117475/android-en-ios-hebben-marktaandeel-van-99-komma-7-procent-op-smartphonemarkt.html

[3] P. Flach, “The Art and Science of Algorithms that Make Sense of Data”, Cambridge University Press, The Edinburgh Building, Cambridge, UK, 2012, ISBN: 978-1-107-42222-3

[4] A.K. Dey, “Understanding and Using Context”, Personal and Ubiquitous Computing, vol. 5, no. 1, 2001, pp. 4-7.

[5] S. W. Loke, “Context-aware artifacts: Two development approaches”, Pervasive Comput., vol. 5, no. 2, pp. 48–53, Apr.–Jun. 2006

[6] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, S. Capkun, “On the requirements for successful GPS spoofing attacks” in Proc. ACM CCS 2011, pp. 75-86, Chicago, IL, 2011

[7] A. S. Greg Milette, "Professional Android Sensor Programming", John Wiley & Sons, New York, NY, USA, 2012, ISBN: 978-1-118-18348-9

[8] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, F.L. Wong. "SenSay: A Context-Aware Mobile Phone", Proceedings of the IEEE International Symposium on Wearable Computers, White Plains, NY, USA, 2003.

[9] M. Baldauf, S. Dustdar, F. Rosenberg, "A survey on context-aware systems", International Journal of Ad Hoc and Ubiquitous Computing, 2(4), 263–277, 2007.

[10] T. Chaari, D. Ejigu, F. Laforest and V.M. Scuturici. “A Comprehensive Approach to Model and Use Context for Adapting Applications in Pervasive Environments.”, Journal of Systems and Software, Volume 80, Issue 12, December 2007, pp. 1973–1992.

[12] G. Deak, K. Curran, J. Condell, “Filters for RSSI-based measurements in a Device-free Passive Localisation Scenario.”, International Journal on Image Processing & Communications 15 (2011) 23–34

[13] L. Schauer, M. Werner, P. Marcus, “Estimating crowd densities and pedestrian flows using wi-fi and bluetooth,” in Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 171–177.

[14] A.C. Petre, C. Chilipirea, M. Baratchi, C. Dobre, M. van Steen. “WiFi tracking of pedestrian behavior.” (2017) In Smart Sensors Networks (pp. 309-337). Academic Press.

[15] J. Zhang, M.H. Firooz, N. Patwari, S.K. Kasera, Advancing wireless link signatures for location distinction, in: Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom '08), 2008.

[16] N. Patwari and S. K. Kasera, "Robust location distinction using temporal link signatures," in MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking, 2007, pp. 111–122.

[17] M. Vanhoef, C. Matte, M. Cunche, L. Cardoso, F. Piessens. Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In ACM AsiaCCS, Xi'an, China, May 2016.

[19] IEEE Std 802.11-2012: IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE, New York, NY, USA, 2012.

[20] A. Mosenia, X. Dai, P. Mittal, N.K. Jha. "PinMe: Tracking a Smartphone User around the World", IEEE Transactions on Multi-Scale Computing Systems (2017).

[21] T. Watanabe, M. Akiyama, T. Mori. "RouteDetector: Sensor-based Positioning System That Exploits Spatio-Temporal Regularity of Human Mobility." *WOOT*. 2015.

[22] AndroidTM Benchmarks, Performance Comparison of Android Devices, CPUMark Rating, laatst geopend op 17 februari 2019: https://www.androidbenchmark.net/cpumark_chart.html

[23] P. Bahl, V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system", Microsoft Research, 2000.

[24] J. Lindqvist et al. "Privacy-preserving 802.11 access-point discovery." *Proceedings of the second ACM conference on Wireless network security*. ACM, 2009.

[25] M. Gruteser, D. Grunwald. "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis." *Mobile Networks and Applications* 10.3 (2005): 315-325.

[26] E. Frøkjær, M. Hertzum, K. Hornbæk. "Measuring usability: are effectiveness, efficiency, and satisfaction really correlated?" *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 2000.

[27] F. Nayebi, J.M. Desharnais, A. Abran. "The state of the art of mobile application usability evaluation." *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. IEEE, 2012.

[28] R. Gafni. "Usability issues in mobile-wireless information systems." *Issues in Informing Science & Information Technology*6 (2009).

[29] H.B. Duh, G.C.B. Tan, V.H. Chen. "Usability evaluation for mobile device: a comparison of laboratory and field tests." *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. ACM, 2006.

[30] S. Naafs, “Living laboratories: the Dutch cities amassing data on oblivious residents” The Guardian (01 maart 2018), laatst geopend op 17 februari 2019: https://www.theguardian.com/cities/2018/mar/01/smart-cities-data-privacy-eindhoven-utrecht

[31] A. Ng, Associate Professor, Stanford University, Chief Scientist, on-line OpenClassroom: Machine Learning, Logistic Regression: Newton's Method I, laatst geopend op 17 februari 2019: http://openclassroom.stanford.edu/MainFolder/VideoPage.php?course=MachineLearning&video=04.6-LogisticRegression-NewtonsMethodI, Stanford, USA

[32] A. Ng, Associate Professor, Stanford University, Chief Scientist, on-line OpenClassroom: Machine Learning, Logistic Regression: Newton's Method II, laatst geopend op 17 februari 2019: http://openclassroom.stanford.edu/MainFolder/VideoPage.php?course=MachineLearning&video=04.7-LogisticRegression-NewtonsMethodII, Stanford, USA

[33] A. Ng, Associate Professor, Stanford University, Chief Scientist, on-line OpenClassroom: Machine Learning, Logistic Regression: Gradient Descent vs Newton's Method, laatst geopend op 17 februari 2019: http://openclassroom.stanford.edu/MainFolder/VideoPage.php?course=MachineLearning&video=04.8-LogisticRegression-GradientDescentVsNewtonsMethod, Stanford, USA

[34] The Android Open Source Project, “Sensor Types”, maart 2017, laatst geopend op 17 februari 2019: https://source.android.com/devices/sensors/sensor-types

[35] H. Zou, T. Hastie. "Regularization and variable selection via the elastic net." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67.2 (2005): 301-320.

[36] J. Zhu, T. Hastie. "Kernel logistic regression and the import vector machine." Journal of Computational and Graphical Statistics 14.1 (2005): 185-205.

[37] The Samsung Sensor Extension, laatst geopend op 17 februari 2019: https://developer.samsung.com/galaxy/sensor-extension

[38] The Samsung Motion SDK, laatst geopend op 17 februari 2019: https://developer.samsung.com/galaxy/motion

[39] E. Charniak. "Bayesian networks without tears." AI magazine 12.4 (1991): 50.

[40] A.K. Jain, M.N. Murty, P.J. Flynn. "Data clustering: a review." ACM computing surveys (CSUR) 31.3 (1999): 264-323.

[41] T.M. Kodinariya et al. "Review on determining number of Cluster in K-Means Clustering." International Journal 1.6 of Advance Research in Computer Science and Management Studies (2013): 90-95.

[42] T. Roos et al. "On discriminative Bayesian network classifiers and logistic regression." *Machine Learning* 59.3 (2005): 267-296.

[43] W.S. Noble. "What is a support vector machine?." *Nature biotechnology* 24.12 (2006): 1565.

[44] G. Ifrim. "Statistical learning techniques for text categorization with sparse labeled data." phd-thesis (2009), Faculties of Natural Sciences and Technology of Saarland University.

[45] A.C. Rencher, G.B. Schaalje (2008). "Linear Models in Statistics", Second Edition, John Wiley & Sons, Inc., Hoboken, New Jersey, USA

[46] H. Arndt, M. Bundschus, A. Naegele (2009). “Towards a Next-Generation Matrix Library for Java”, 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC), 2009, Department of Computer Science, Technical University of Munich, Germany, laatst geopend op 16 maart 2019: https://holger-arndt.de/publications/COMPSAC2009-ujmp-draft.pdf

[47] Plotly, Modern Analytic Apps for the Enterprise, Achieve next-level returns on your enterprise data investment, laatst geopend op 14 april 2019: https://plot.ly/ en https://plot.ly/create/#/

[48] Physics Toolbox Pro, Vieyra Software, laatst geopend op 14 april 2019: https://www.vieyrasoftware.net/single-post/2017/03/12/Physics-Toolbox-Pro-Now-Available

[49] Falcon SQL Client, laatst geopend op 14 april 2019: https://plot.ly/free-sql-client-download/

Appendix A

Bij dit onderzoeksverslag vinden we verschillende bijlagen, welke op een gegevensdrager aangeleverd worden. Deze zijn:

1.) Paragraaf 6.4.6. – Metingen met succesvolle sensortypen
Binnen de map "6.4.6. – Metingen met succesvolle sensortypen" is de gevonden data opgenomen, waarover in de tabel van paragraaf 6.4.6. gesproken wordt. Hier wordt de gevonden onderzoeksdata in SQLite-databases en in Excel-csv-formaat gepresenteerd, afhankelijk van de sensortype in kwestie.

2.) De onderzoekssituaties
Binnen de map "Onderzoekssituaties" vinden we de onderzoeksdata, die tijdens de verschillende onderzoekssituaties uit hoofdstuk negen verzameld zijn, onderverdeeld naar elke specifieke onderzoekssituatie. Dit geldt voor:

1. de machine-learning-toepassing;
2. de passive-scanning-polling-app.

De gepresenteerde onderzoeksdata wordt hier gepresenteerd in SQLite-databases.

3.) Presentatiefilmpjes van de onderzoekstoepassingen
Binnen de map "Presentatiefilmjes toepassingen" vinden we twee opeenvolgende presentatiefilmpjes, namelijk van:

1. de machine-learning-toepassing;
2. de passive-scanning-polling-app.

4.) Broncode Onderzoekstoepassingen
Binnen de map "Broncode Onderzoekstoepassingen" vinden we de broncode van de machine-learning-toepassing en van de passive-scanning-polling-app. Deze zijn beide uitgeprogrammeerd met Android Studio 3.4. Daardoor is het mogelijk dat we een rebuild moeten toepassen, wanneer we Android Studio op een ander systeem gebruiken, om de programmatuur werkend te krijgen.

De broncode is ook op een andere wijze gebruiken, doordat deze van beide toepassingen in platte tekst is in te zien. Daartoe navigeren we voor de machine-learning-toepassing en de passive-scanning-polling-app naar respectievelijk de mappen, met hun onderliggende mappen:

1.) Broncode Onderzoekstoepassingen\Machine-learning-toepassing\Abstract_all\app\src\main\java\nl\aksel\abstract_all

2.) Broncode Onderzoekstoepassingen\Passive-scanning-polling-app\ScanWifi\app\src\main\java\nl\aksel\scanwifi

Voor de machine-learning-toepassing en de passive-scanning-polling-app geldt dat de geïmporteerde bibliotheken of packages te vinden zijn in de respectievelijke mappen:

1.) Broncode Onderzoekstoepassingen\Machine-learning-toepassing\Abstract_all\app\libs

2.) Broncode Onderzoekstoepassingen\Passive-scanning-polling-app\ScanWifi\app\libs

Appendix B

Tijdens de trainingsperiode van een machine-learning-oplossing worden sensoren gehanteerd, waarvan we niet zonder meer kunnen aannemen dat deze voor de classificatie van een context van belang zijn in dit onderzoek (§ 8.6.2.). Dit zijn onder meer de volgende sensoren op een Samsung Galaxy S9, waaronder ook synthetische sensoren die zelf samengesteld zijn:

Communicatienetwerken

Sensor:	Grootheid:	Toelichting:
Cell Tower Id (CID)	Natuurlijk getal	In het bereik van nul tot 65.535
Local Area Code (LAC)	Natuurlijk getal	
Gemiddelde GSM Signal Strength	Rationaal getal	Ontvangen zendsterkte GSM-mast
Het MAC-adres van een reguliere WiFi-access-point, waarmee een verbinding onderhouden wordt.	Natuurlijk getal	Een hexadecimale representatie in het decimale bereik van nul tot 256^6.
De Euclidische afstand over de (zes) hexadecimale representaties, waaruit het laatst bekende MAC-adres van een reguliere WiFi-access-point is opgebouwd.	Rationaal getal	Een MAC-adres wordt genoteerd als paren van hexadecimale cijfers, gescheiden door dubbele punten. De Euclidische afstand vervangt dit, maar is daardoor minder exact.

Smartphonedraaiing

Sensor:	Grootheid:	Toelichting:
Oriëntatie	Graden (°)	Azimuth, Pitch en Roll
Smartphone-gyroscoop	Radialen / seconde (rad/s)	Gyroscoop in rotation-vector-sensor, § 6.4.1.
Ongekalibreerde gyroscoop	Radialen / seconde (rad/s)	
Screen Orientation	Natuurlijk getal	Voor elke gedraaide veelvoud van 90°: • Voor 0° geeft deze een nul terug; • Voor 90° geeft deze een één terug; • Voor 180° geeft hij een twee terug; • Voor 270° geeft hij een drie terug.
Interrupt Gyroscope	Radialen / seconde (rad/s)	
Samsung Game Rotation Vector	Rationaal getal	Positie over de drie assen van smartphone's cartesische assenstelsel, onafhankelijk van het aardmagnetisch noorden.

Beweging

Sensor:	Grootheid:	Toelichting:
(Raw) Accelerometer, zonder zwaartekrachtfilter	m/s^2	Gemeten over de drie assen van smartphone's cartesische assenstelsel.
(Raw) Accelerometer, ongekalibreerd	m/s^2	Gemeten over de drie assen van smartphone's cartesische assenstelsel.
Lineaire Accelerometer, tevens in bewegingssensor, § 6.4.5.	m/s^2	Gemeten over de drie assen van smartphone's cartesische assenstelsel.
Significant Motion Sensor	Geen waarde of één, als natuurlijk getal	Deze sensor reageert op nieuwe bewegingen, bijvoorbeeld wandelen, fietsen of het besturen van een auto.
Samsung Tilt Sensor	Geen waarde of één, als natuurlijk getal	Reactie op het schudden van een smartphone.
Samsung Pick Up Gesture	Geen waarde of één, als natuurlijk getal	Reactie op het oppakken van een smartphone.
SX9320 Grip Sensor	Geen waarde of één, als natuurlijk getal	Reactie op het vastpakken van een smartphone.
Samsung Motion Sensor	Nul of een getal per bewegingstype	Zie § 6.4.5. In de training-sample-tijd worden de laatste en de gemiddelde waarden per beweging bijgehouden.

Navigatie

Sensor:	Grootheid:	Toelichting:
Magnetometer	microtesla (μT)	Magnetisch veld (§ 6.3.3. en § 6.4.4.)
Ongekalibreerde magnetometer	microtesla (μT)	Magnetisch veld
GPS: a) Breedtegraad b) Lengtegraad c) Hoogte	a) Graden (°) b) Graden (°) c) Meter (m)	

Omgevingseigenschappen

Sensor:	Grootheid:	Toelichting:
Geluid	Decibel (dB)	Gemeten wordt (§ 6.3.2. en § 6.4.4.): ● Gemiddeld geluidsniveau; ● Mediaan van het geluidsniveau; ● Piekdecibellen; ● Gemiddelde piekdecibellen.
Barometer	Hectopascal (hPa) of millibar (mbar)	1 hPa = 1 mbar
Zwaartekracht	m/s^2	

Overig

Sensor:	Grootheid:	Toelichting:
Timestamp	Tijd	Een string-representatie van een java.util.Date-object
Verstreken minuten na middernacht	Natuurlijk getal	