UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Ontwikkeling en evaluatie van een raamwerk voor vereisten in Machine Learning projecten: een literatuurstudie en methodologische verkenning

Michelle Watteel

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

dr. Frank VANHOENSHOVEN

2023
2024

UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Ontwikkeling en evaluatie van een raamwerk voor vereisten in Machine Learning projecten: een literatuurstudie en methodologische verkenning

Michelle Watteel

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

dr. Frank VANHOENSHOVEN

Ontwikkeling en evaluatie van een raamwerk voor vereisten in Machine Learning projecten: een literatuurstudie en methodologische verkenning

Michelle Watteel

Abstract In klassieke softwareontwikkeling wordt logica geschreven door de ontwikkelaar. Bij de ontwikkeling van Machine Learning (ML) projecten ligt dat anders. De logica wordt in dat geval geleerd of zelfs gecreëerd door de computer. Traditionele methoden voor het capteren en documenteren van vereisten zijn afgestemd op die eerste manier van softwareontwikkeling. Huidig onderzoek wijst uit dat er een gebrek is aan gevestigde methodologieën voor het capteren en documenteren van vereisten binnen ML-projecten. Omgaan met vereisten is echter een cruciale stap in het softwareontwikkelingsproces. Om zo'n methodologie op te stellen is het van belang dat er in kaart gebracht wordt welke vereisten relevant zijn in ML-projecten. Aan de hand van een kritische literatuurstudie en validatie door middel van interviews, tracht dit onderzoek een overzicht te bieden van die relevante vereisten om een succesvol ML-systeem te ontwikkelen. Daarenboven worden huidige methodologieën voor het capteren en documenteren van vereisten zowel in de academische literatuur als de industrie verkend. Er werd een raamwerk opgesteld van 32 ML-relevante vereisten, dewelke zijn gevalideerd door middel van interviews met domeinexperten. Uit de interviews kwam naar voren dat er geen frequent toegepaste methoden zijn voor het capteren en documenteren van vereisten in ML-projecten. Dit onderzoek kan de basis vormen voor een methodologie waarbij er rekening gehouden wordt met de belangrijkste ML-vereisten.

Inhoudsopgave

1 Introductie

Het gebruik van zelflerende systemen of componenten is alsmaar meer populair, denk bijvoorbeeld aan ChatGPT of zelfrijdende wagens. Die zelflerende systemen worden mogelijk gemaakt door Machine Learning (ML) technologie, een variant van Artificiële Intelligentie (AI).

Zelflerendheid impliceert dat die systemen bepaald gedrag aanleren en vertonen dat niet expliciet geprogrammeerd werd. Dat wil ook zeggen dat het moeilijk is om een ML-systeem gedrag te laten vertonen dat vooraf gespecificeerd werd in de vorm van vereisten (Heyn e.a. 2021; Hu e.a. 2020; Rahimi e.a. 2019). Daarnaast is een ML-systeem vaak meer complex dan andere softwaresystemen, waardoor het nieuwe vereisten met zich meebrengt (Ahmad, Abdelrazek e.a. 2023; Kuwajima en Ishikawa 2019). Door die complexiteit is het eveneens moeilijker te bepalen en te begrijpen hoe het systeem beslissingen maakt, waardoor vereisten zoals eerlijkheid en verklaarbaarheid nodig zijn (Kuwajima en Ishikawa 2019).

Rekening houden met vereisten impliceert de nood aan het capteren ervan. Vereisten verzamelen is een cruciaal element in de ontwikkelingscyclus om projectsucces te stimuleren, evenals de noden van de klant te vervullen (Khan e.a. 2022; Ranawana en Karunananda 2021). Vereisten worden onder andere verzameld door het raadplegen van relevante stakeholders en vormen een communicatiemiddel tussen die stakeholders en de ontwikkelaars van het ML-systeem (Ahmad, Bano e.a. 2021; IBM g.d.). Het is de grondlaag waarop een project bouwt, de eerste stap in het ontwikkelingsproces (Shao en Wang 2022).

De nood om die vereisten op een gestructureerde manier vast te leggen, is in verschillende sectoren erg hoog. Er wordt veelvuldig gebruik gemaakt van ML-componenten of systemen in toepassingen waarbij veiligheid zeer belangrijk is, denk bijvoorbeeld aan zelfrijdende wagens (Al Alamin en Uddin 2021; Hu e.a. 2020). Voor zulke toepassingen is het zeer belangrijk dat het systeem zich binnen de vooropgestelde beperkingen (vereisten) gedraagt.

Huidig onderzoek geeft aan dat er nood is aan meer kennis omtrent het in kaart brengen van de vereisten die een AI-systeem met zich meebrengt (Yoshioka e.a. 2021). Daarnaast wordt er aangegeven dat klassieke methoden om vereisten te capteren en documenteren niet altijd werken voor ML-projecten wegens de hoge complexiteit en nieuwe vereisten die in acht genomen moeten worden (Al Alamin en Uddin 2021; Kuwajima en Ishikawa 2019). Daarbovenop is er een gebrek aan een gestandaardiseerd proces of methodologie voor het ontwikkelen van ML-systemen (Shao en Wang 2022).

Dit onderzoek tracht in kaart te brengen welke, al dan niet nieuwe, vereisten in acht genomen kunnen of moeten worden tijdens het ontwikkelen of implementeren van een succesvol ML-systeem of component. Wat betekenen de nieuwe vereisten die in de literatuur voorkomen?

De relevantie van die set aan vereisten wordt vervolgens afgetoetst aan de hand van interviews met experts in de industrie. Welke methoden er reeds bestaan om de vereisten te capteren en documenteren voor ML-projecten zal eveneens onderzocht worden.

Aan de hand van dit onderzoek zullen volgende onderzoeksvragen beantwoord worden:

- Welke vereisten zijn relevant in ML-projecten?
- Welke methoden om die vereisten in kaart te brengen, worden reeds toegepast?

De gebruikte methodologie in dit onderzoek wordt in de volgende sectie uiteengezet. Daarna wordt er getracht een overzicht te geven van de literatuur omtrent vereisten die relevant zijn voor het succesvol ontwikkelen en implementeren van een ML-systeem. In sectie 3.2 wordt er een analyse gemaakt van bestaande methoden om vereisten te capteren en documenteren, en hoe ze kunnen toegepast worden op ML-projecten. Dat wordt gevolgd door een meta-analyse van de literatuur en de definities die dit onderzoek naar voren brengt voor de verschillende vereisten. De validatie van het ontwikkelde raamwerk van vereisten en de courante gebruiken in verband met het capteren en documenteren van vereisten wordt voorgesteld in sectie 4.4. In sectie vijf worden de onderzoeksvragen beantwoord in de vorm van een discussie. Daaropvolgend worden de validiteitsrisico's uiteengezet. Aanbevelingen voor toekomstig onderzoek worden eveneens in die sectie toegelicht, gevolg door de conclusie in sectie zes.

2 Methodologie

2.1 Onderzoeksmethode

Het doel van dit onderzoek is de huidige kennis omtrent ML-specifieke vereisten binnen zowel de academische wereld als de industrie te beschrijven. Er werd daarom door de auteur gekozen om een kritische literatuurstudie uit te voeren, en daarna de onderzoeksresultaten te valideren door middel van semi-gestructureerde interviews met verschillende profielen uit de industrie. De kritische literatuurstudie draagt bij aan het consolideren van bestaand onderzoek omtrent vereisten en methoden om die vereisten te capteren en documenteren in Machine Learning projecten. Het overzicht aan vereisten dat voortvloeit uit die literatuurstudie en de daaropvolgende meta-analyse, is het onderwerp van de validatie-interviews.

2.2 Zoekstrategie

De informatiebronnen die geraadpleegd zullen worden voor dit onderzoek zijn zowel wetenschappelijke literatuur (artikels uit wetenschappelijke tijdschriften en conference proceedings) als *grey literature* zoals thesissen en industriestandaarden. De keuze om *grey literature* te betrekken is een gevolg van de actualiteit van het onderzoeksonderwerp. Het is daarbij belangrijk dat kennis uit de industrie opgenomen wordt in dit onderzoek.

Literatuur werd verzameld uit de volgende databases:

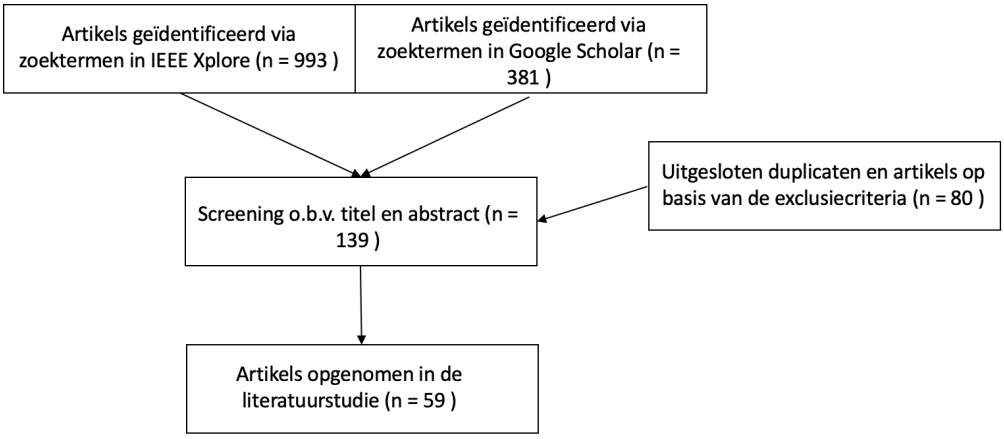
- Google Scholar
- IEEE Xplore

- Medium

IEEE Xplore biedt een uitgebreide selectie aan peer-reviewed literatuur, gericht op technologie en recente ontwikkelingen binnen het veld van machine learning. Aanvullend werd Google Scholar gebruikt om een bredere selectie aan wetenschappelijke tijdschriften te bereiken. Medium is een online platform waarop artikels gedeeld kunnen worden over een breed scala aan onderwerpen. De literatuur van Medium die geselecteerd en gebruikt werd, werd geverifieerd op validiteit door de auteur. Die validiteit betekent in dit onderzoek dat de inhoud door meerdere personen (in de industrie) geverifieerd werd.

Na selectie van de literatuur zullen duplicaten verwijderd worden. Daarnaast zal citation chaining toegepast worden om volledigheid van het onderzoek te garanderen.

Het aantal bronnen dat uiteindelijk werd weerhouden, wordt weergegeven in figuur 1.



Figuur 1: PRISMA-flow diagram (Page e.a. 2021)

2.2.1 Zoektermen

De zoektermen die gebruikt zullen worden in deze literatuurstudie zijn de volgende:

- ”requirements” AND (”AI” OR ”ML” OR ”artificial intelligence” OR ”machine learning”)
- ”requirements capturing” AND (”AI” OR ”ML” OR ”artificial intelligence” OR ”machine learning”)
- ”conceptual modelling” AND (”AI” OR ”ML” OR ”artificial intelligence” OR ”machine learning”)
- ”requirements engineering” AND (”AI” OR ”ML” OR ”artificial intelligence” OR ”machine learning”)

- ”requirements” AND ((”gathering”) OR (”capturing”) OR (”elicitation”)) AND (”AI” OR ”ML” OR ”artificial intelligence” OR ”machine learning”)

Naast bovenstaande zoektermen, werd er ook gezocht naar bronnen met betrekking tot verschillende documentatiemethoden die voorkomen in de geselecteerde literatuur:

- ”GORE” AND ”requirements”
- ”Use cases” AND ”requirements”
- ”User stories” AND ”requirements”
- ”UML” OR ”Unified Modeling Language” AND ”requirements”
- ”Video” OR ”audio” AND ”requirements”
- ”Practical Action Research Design Method”

Let wel: bovenstaande zoektermen werden gebruikt om een korte theoretische toelichting te bieden over de methoden, niet om een volledige literatuurstudie uit te voeren.

Na een verkennende literatuurstudie werd er besloten om bronnen te beperken in de tijd. Bronnen mogen maximaal 24 jaar oud zijn, en dus maximaal dateren uit het jaar 2000.

2.2.2 In- en exclusiecriteria

Door middel van deze criteria wordt er bepaald of literatuur al dan niet opgenomen wordt in dit onderzoek. Volgende criteria zijn van toepassing:

- IN1: De bron heeft betrekking op (het capteren of documenteren van) vereisten in een AI- of ML-context.
- EX1: De bron is niet in het Nederlands of Engels geschreven.
- EX2: De bron werd niet onderworpen aan enige vorm van validatie of peer-review.
- EX3: De bron heeft betrekking op ML- of AI-technieken om vereisten in traditionele softwareprojecten te capteren (Yoshioka e.a. 2021).

2.2.3 Analysedimensies

Om beide onderzoeksvragen te kunnen beantwoorden, werd de literatuur onderverdeeld in twee analysedimensies:

- D1: Vereisten in ML-projecten
- D2: Methoden om vereisten in ML-projecten te capteren

2.3 Interviewmethode

ML is een actueel onderwerp en wordt momenteel veel toegepast in de bedrijfswereld (Ahmad, Abdelrazek e.a. 2023; Shao en Wang 2022). Het is daarom belangrijk om dat perspectief op te nemen in dit onderzoek. Om de relevantie van het opgestelde raamwerk aan vereisten te valideren, werden er interviews uitgevoerd met respondenten die ervaren zijn met het ontwikkelen van ML-systemen of het leiden van projecten waarin ML-systemen ontwikkeld worden. Onderstaande tabel (tabel 1) geeft een overzicht van de verschillende respondenten. Er werd gekozen om de validatie beperkt te houden omdat het als ondersteunend onderdeel moet dienen voor het grondige onderzoek dat eraan vooraf ging. Het doel is om het volledige raamwerk te kunnen valideren, verspreid over de vier respondenten. Sommige vereisten konden meerdere keren bevraagd worden wegens het natuurlijke verloop van het interview, of wanneer de respondent een reeds gevalideerde vereiste zelf aanhaalde.

Tabel 1: Respondenten

Respondenten	
No.	Profiel
1	Team Lead AI innovation
2	Data Scientist
3	Co-founder en CTO ML-startup
4	Onderzoeker AI en data

Alle respondenten hebben ervaring met het ontwikkelen van ML-systemen. Respondent 1 en 2 zijn actief in de klantgerichte sector waarin ze ML-projecten uitvoeren voor klanten. Respondent 3 heeft het ML-algoritme voor diens eigen bedrijf ontwikkeld en respondent 4 voert onderzoek uit in samenwerking met en in dienst van bedrijven die willen innoveren met AI en ML.

Wegens het reeds opgestelde theoretisch raamwerk werd er gekozen voor een semi-gestructureerde interviewmethode volgens de richtlijnen van Kallio e.a. 2016. Daarbij wordt er eerst een diepgaand begrip van het te onderzoeken onderwerp ontwikkeld, wat in dit geval gebeurde door de uitgebreide literatuurstudie. Daarna werd er een eerste versie van het interview opgesteld en gevalideerd met de promotor van dit onderzoek en tevens expert in het vakgebied. De laatste stap in het proces is het uitvoeren van de interviews.

Na het uitvoeren van de interviews werden ze deductief gecodeerd volgens de methode van Pearse 2019. Deductief coderen is een gepaste methode voor dit onderzoek dankzij het vooraf opgestelde codeboek aan de hand van de literatuurstudie. De interviewresultaten worden reeds geïntegreerd in het finale raamwerk in sectie 4.3, en expliciet benoemd in sectie 4.4. Zowel de interviewvragen als het codeboek kunnen teruggevonden worden in appendix A.1.

3 Literatuurstudie

Onderstaande sectie is het resultaat van de diepgaande literatuurstudie, gemotiveerd door beide onderzoeksvragen. Eerst zal de in dit onderzoek gehanteerde definitie van vereisten toegelicht worden. Daarna wordt er een overzicht geboden van de vereisten zoals ze zich voordoen in de literatuur, verdeeld in de categorieën niet-functionele en functionele vereisten. Tot slot worden de captatie- en documentatiemethoden van vereisten in ML-projecten toegelicht.

3.1 Wat zijn vereisten?

Vereisten zijn een communicatiemiddel tussen de gebruikers en de ontwikkelaars van een project (IBM g.d.). Ze specifiëren wat de software moet doen, hoe het dat moet doen en hoe het dat niet moet doen (Westfall 2005). Er zijn meerdere categorieën van vereisten, de twee grootste zijn functionele en niet-functionele vereisten. Functionele vereisten beschrijven het systeem, het doel van het systeem, de functies die het moet hebben en de positie van het systeem tegenover de maatschappij (IBM g.d.). Niet-functionele vereisten zijn beperkingen die de oplossing opgelegd worden, zoals de veiligheid en de performantie van het systeem of kwaliteiten die een gebruiker wenst van het systeem (Abran en Moore 2001; IBM g.d.). Voorgaande definities zullen doorheen dit onderzoek toegepast worden om de individuele ML-vereisten te categoriseren.

3.2 Relevante vereisten in ML-projecten

3.2.1 Niet-functionele vereisten

In deze sectie wordt een overzicht geboden van niet-functionele vereisten van ML-systemen die voorkomen in de bestudeerde literatuur. De termen worden gecategoriseerd en gedefinieerd op basis van de literatuur. Het is daarom mogelijk dat er enkele overlappingen voorkomen tussen verschillende vereisten. In de sectie *meta-analyse van niet-functionele vereisten* wordt die overlap expliciet gemaakt door een vergelijkende studie in de vorm van een matrix.

Transparency, transparantie Het verklaren van de resultaten of voorspellingen van een ML-systeem is transparantie volgens Horkoff 2019 en Zhang e.a. 2021. Fagbola en Thakur 2019 geven dezelfde definitie, maar dat onderzoek voegt eraan toe dat het systeem begrijpbaar moet zijn voor mensen, onafhankelijk van de expertise. Transparantie kan op drie niveau's toegepast worden, volgens Fagbola en Thakur 2019 en Gjorgjevikj e.a. 2023: op het niveau van het volledige model, het niveau van de componenten of het niveau van het trainingsalgoritme.

Accountability, verantwoordelijkheid Verantwoordelijkheid van een ML-systeem wordt gedefinieerd als volgt: alle componenten en acties van een ML-systeem zijn traceerbaar en verantwoordelijk (*accountable*) (Zhang e.a. 2021). Bijvoorbeeld: een zelfrijdende auto

moet een log bijhouden van alle acties uitgevoerd door zowel de bestuurder als het autonome systeem (Zhang e.a. 2021).

Assessment, evaluatie Om het vertrouwen in ML-systemen te verhogen is er nood aan verschillende evaluatietechnieken van die systemen (Zhang e.a. 2021). Die evaluatie kan volgens Zhang e.a. 2021 op twee niveau's plaatsvinden. Ten eerste is er zelfevaluatie, waarbij een component van het AI-systeem zichzelf zal testen en evalueren op *bias* (Zhang e.a. 2021). Daarnaast spreekt men in Zhang e.a. 2021 van onafhankelijke externe evaluaties. Volgens Wan e.a. 2020 betekent het evalueren van het ML-systeem, het evalueren van gedefinieerde evaluatieparameters aan de hand van testdata.

Robustness, robuustheid Robuustheid van het ML-systeem betreft de mogelijkheid ervan om met onverwachte, ongeziene data en andere omgevingsomstandigheden om te gaan (Gjorgjevikj e.a. 2023; Heck en Schouten 2023; Hu e.a. 2020). De auteurs in Hu e.a. 2020 gaan hier nog een stapje verder in door te zeggen dat robuuste ML-systemen als uitgangspunt menselijke perceptie moeten gebruiken. Het systeem moet robuust zijn tegen alle soorten input waarbij menselijke perceptie niet zou wijzigen (Hu e.a. 2020). Een voorbeeld: een ML-systeem moet in staat zijn een voetganger te detecteren, ook wanneer er op de achtergrond ruis aanwezig is, of wanneer de afbeelding geroteerd wordt (Hu e.a. 2020). Deze vereiste impliceert de nood aan grotere en meer complexe datasets om die algoritmes te trainen (Gjorgjevikj e.a. 2023). Daarnaast spreken Gjorgjevikj e.a. 2023 van een *trade-off* tussen robuustheid en accuraatheid van het ML-systeem.

Zhang e.a. 2021 hanteren een andere definitie voor robuustheid, het zou betekenen dat de output van het AI-systeem eerlijk, veilig, accuraat en betrouwbaar moet zijn.

In Kästner 2022 wordt de volgende definitie gehanteerd: robuustheid geeft aan in welke mate de voorspellingen van een model stabiel blijven wanneer de input licht gewijzigd wordt (Kästner 2022). Robuustheid is volgens dat artikel ook een nuttig inzicht om in te schatten of een model aangevallen wordt met kwaadwillige input (Kästner 2022).

Fairness, eerlijkheid Volgens Horkoff 2019 werd er reeds menig onderzoek gevoerd naar een correcte definitie van eerlijkheid. Daaruit bleek dat eerlijkheid afhangt van hoe het gedefinieerd en gemeten wordt in het betreffende project (Horkoff 2019). Ook Gjorgjevikj e.a. 2023 benadrukt de verschillende pogingen tot het definiëren van eerlijkheid. Eén van de definities is: 'eerlijkheid kan gedefinieerd worden als de afwezigheid van vooroordeel of favoritisme ten opzichte van individuen of groepen, gebaseerd op bepaalde inherente of verkregen karakteristieken' (Gjorgjevikj e.a. 2023). In het werk van Fagbola en Thakur 2019 wordt eerlijkheid als synoniem van gelijkheid en onpartijdigheid (*unbiasedness*) gedefinieerd.

Accuracy, accuraatheid In Horkoff 2019 meet accuraatheid hoe correct de output van het ML-systeem is ten opzichte van de werkelijkheid. Accuraatheid in ML-projecten kan onder andere gemeten worden door middel van de parameters *precision* en *recall* (Horkoff 2019). [Er bestaan meerdere maatstaven om accuraatheid te bepalen, zoals lift (Vogelsang en Borg 2019)]

Unbiasedness, zonder vooroordelen Volgens verschillende onderzoeken is er sprake van *bias* van het ML-systeem wanneer de output ervan oneerlijk is of discriminerend is tegenover iemand op sociaal of economisch gebied (Heck en Schouten 2023; Zhang e.a. 2021). Die vereiste wordt dus gerelateerd aan de eerlijkheidsvereiste (Heck en Schouten 2023; Zhang e.a. 2021). In Laato e.a. 2022 neemt men enkele andere factoren mee in rekening wanneer ze spreken over *bias*: er is ook sprake van *bias* wanneer de output van het systeem afwijkt van het normale. Een voorbeeld van bias in een ML-systeem wordt gegeven in Zhang e.a. 2021: een systeem dat moest voorspellen of iemand crimineel gedrag zou vertonen, bleek *biased* te zijn tegenover zwarte deelnemers. Zij zouden volgens het algoritme een hogere kans hebben op het begaan van misdaden in de toekomst (Zhang e.a. 2021).

Data requirements, datavereisten In een zelflerend systeem speelt data een cruciale rol, er is namelijk een grote hoeveelheid aan kwalitatieve data nodig om een werkend ML-systeem te ontwikkelen (Zhang e.a. 2021). Het is belangrijk om rekening te houden met de beschikbaarheid, kwaliteit, selectie en het testen van de data. Om het niveau van die datavereisten te verzekeren, is het belangrijk om ze te documenteren (Ahmad, Bano e.a. 2021). Het definiëren van vereisten omtrent data is dus essentieel voor het verzekeren van niet alleen de kwaliteit van de data, maar van het gehele ML-systeem (Dey 2022; Heyn e.a. 2021). Daarnaast moet verzamelde data eveneens opgeslagen worden, door de grote hoeveelheden is het belangrijk om dit selectief te doen om kosten te besparen (Kinkiri en Melis 2016). Het is dus belangrijk om te bepalen welke data opgeslagen moet worden, en welke niet, afhankelijk van het gebruikte algoritme (Kinkiri en Melis 2016).

Data versioning In sommige ML-projecten worden meerdere modellen getraind (Laato e.a. 2022). In die gevallen is het belangrijk om bij te houden welk model met welke dataset getraind werd om eventuele problemen te kunnen traceren (Laato e.a. 2022).

Data transparency, datatransparantie Zhang e.a. 2021 definiëren het delen van kwalitatieve datasets als een succesfactor om *biased* ML-modellen en resultaten te verhelpen. Data is vaak eigendom van een kleine fractie van grote bedrijven, het delen ervan zou het vertrouwen in ML-systemen kunnen verhogen (Zhang e.a. 2021).

Data consistency, dataconsistentie Een consistente instroom van kwalitatieve data is een vereiste voor een goed-werkend ML-systeem (Ranawana en Karunananda 2021). Wanneer een systeem grote hoeveelheden manueel gelabelde data vereist, is dat moeilijk te verzekeren en begint men met een hoeveelheid waarmee het ML-systeem kan convergeren (Ranawana en Karunananda 2021). Volgens Vogelsang en Borg 2019 is dataconsistentie gerelateerd aan het format en de representatie van de data die hetzelfde moet zijn voor de gehele dataset.

Data correctness, correctheid van de data Het is belangrijk dat de gebruikte data ook klopt, en dat kan gerelateerd zijn aan de manier waarop de data verzameld werd (Vogelsang en Borg 2019). Publieke datasets zijn volgens Vogelsang en Borg 2019 niet altijd betrouwbaar, omdat er niemand betaald wordt om zo'n grote hoeveelheid aan data te onderhouden.

Data labeling Er wordt een label toegekend dat voor elk dataobject aanduidt wat het is (Wan e.a. 2020). Wanneer data gelabeld wordt door mensen, kan je er bijna zeker van zijn dat

de data bevooroordeeld is (Vogelsang en Borg 2019). In een domein zoals gezondheidszorg vergt het labelen van data een expert (Pareek e.a. 2022).

Protected attributes, beschermde attributen Beschermde attributen zijn attributen in de dataset die niet gebruikt zullen worden tijdens de classificatie die het ML-systeem uitvoert (Vogelsang en Borg 2019). Wanneer het belangrijk is om in het project *bias* te elimineren, moet er nagedacht worden over de attributen die beschermd zullen worden (Vogelsang en Borg 2019).

Model tuning, afstemmen van het model De parameters die het model gebruikt, worden aangepast en eventuele problemen in het model worden geïdentificeerd (Wan e.a. 2020).

Ethics, ethiek De vele mogelijkheden die een ML-systeem met zich meebrengt, kunnen een positieve maar ook een negatieve invloed hebben op de samenleving (Thinyane en Goldkind 2020). Wanneer men rekening houdt met ethiek tijdens het ontwikkelen, wordt er bijgedragen aan het globale welzijn van de mensheid (Thinyane en Goldkind 2020). In Ahmad, Bano e.a. 2021 worden de ethische richtlijnen van de Europese Commissie aangehaald. Daarin wordt ethiek gekoppeld aan respect voor de mensheid, het voorkomen van schade, eerlijkheid en verklaarbaarheid (Ahmad 2021). Rekening houden met deze vereiste doet men best vanaf het begin van het project, de trainingdata en validatiedata kan namelijk reeds onethische output veroorzaken (Fagbola en Thakur 2019).

Interpretability, interpreteerbaarheid Er is discussie in de huidige literatuur over de term interpreteerbaarheid. Deze vereiste wordt vaak als synoniem voor *explainability* (verklaarbaarheid) gebruikt (Gjorgjevikj e.a. 2023; Habiba e.a. 2022). Interpreteerbaarheid wordt in Gjorgjevikj e.a. 2023 gedefinieerd als de mogelijkheid om iets uit te leggen op een manier die begrijpbaar is voor een persoon. Ander onderzoek zegt dat de term subjectief is, afhankelijk van de probleemcontext en de doelgroep (Gjorgjevikj e.a. 2023). In Fagbola en Thakur 2019 wordt volgende betekenis aan interpreteerbaarheid gegeven: de mate waarin mensen kunnen begrijpen hoe een model werkt. Gegeven enkele parameters die het algoritme gebruikt, kan een mens mogelijke output voorspellen (Fagbola en Thakur 2019). Habiba e.a. 2022 hanteren voor hun onderzoek de definitie dat de stakeholder de oorzaak-gevolgrelatie tussen de in- en output van het model kan begrijpen. In Kästner 2022 wordt interpreteerbaarheid gedefinieerd als de mate waarin een mens de interne werking van het model kan begrijpen, met als doel *debugging* of het auditeren van het model.

Privacy Het gebruik van persoonlijke data tijdens het trainen van een ML-model kan ervoor zorgen dat die gevoelige informatie gestolen wordt (Gjorgjevikj e.a. 2023; Horkoff 2019; Kästner 2022). In het onderzoek van Heck en Schouten 2023 werd er daarom gekozen voor het niet gebruiken van privacygevoelige data in het ML-systeem. Het opslaan van bijvoorbeeld metadata van foto's die een gebruiker maakt, kan aantonen dat er een dure camera gebruikt werd en waar die persoon zich vaak bevindt (Heck en Schouten 2023).

Safety, veiligheid Het gebruik van ML-systemen in verschillende domeinen, brengt nieuwe veiligheidsrisico's met zich mee (Gjorgjevikj e.a. 2023). Enkele voorbeelden zijn volgens Gjorgjevikj e.a. 2023: de assumptie dat trainingdata afkomstig is van de operationele

distributie, de lage kansdichtheid van de operationele distributie in bepaalde regio's en de onzekerheid afkomstig van de manier waarop de testset werd geconcretiseerd. Veiligheid wordt ook gedefinieerd als de minimalisatie van risico en onzekerheid gerelateerd aan schadelijke gebeurtenissen (Gjorgjevikj e.a. 2023). Volgens Fagbola en Thakur 2019 wordt veiligheid gebaseerd op een veilig gebruik voor mensen, verficatie en validatie. In Kuwajima en Ishikawa 2019 wordt veiligheid aan technische robuustheid, weerstand tegen aanvallen, accuraatheid, betrouwbaarheid en reproduceerbaarheid gerelateerd. Ook in het onderzoek van Zhang e.a. 2021 wordt het veiligheidsrisico van externe aanvallen aangehaald, zoals *data poisoning*. De term die in dat onderzoek gebruikt wordt is *security* (Zhang e.a. 2021).

Traceable paths, traceerbare paden In Rahimi e.a. 2019 wordt het traceerbaar pad als vereiste voorgesteld. Dat wil zeggen dat het mogelijk is om de overeenstemming tussen de code en de vooropgestelde design- en coderichtlijnen op te volgen (Rahimi e.a. 2019). Het kan de veiligheid van het systeem bevorderen door het traceren van de inperking van gedefinieerde risico's (Rahimi e.a. 2019).

Generalizability, generaliseerbaarheid Het onderzoek van Heck en Schouten 2023 benadrukt generaliseerbaarheid als vereiste. Generaliseerbaarheid betekent dat de regels die het ML-model leert uit de trainingdata, ook gelden voor nieuwe data (Heck en Schouten 2023). In dat onderzoek wordt er een systeem ontwikkeld waarbij men bloemen kan fotograferen en vervolgens opladen in een database, waardoor de populatie van de bloemensoorten in kaart gebracht kan worden (Heck en Schouten 2023). Het is hierbij cruciaal dat foto's genomen met verschillende soorten toestellen en op verschillende plaatsen eveneens herkend kunnen worden door het systeem (Heck en Schouten 2023).

Model correctness, nauwkeurigheid van het model Ook in het onderzoek van Heck en Schouten 2023 is de nauwkeurigheid van het model belangrijk, dat wordt gerealiseerd door de vereiste dat het systeem correcte regels leert uit de trainingdata. Het werd in dat onderzoek geïmplementeerd door bijvoorbeeld een minimale zekerheid van 75 % te vereisen voor het herkennen van gekende bloemensoorten (Heck en Schouten 2023).

Controlability, controleerbaarheid Controleerbaarheid van het ML-systeem wordt gedefinieerd als de mogelijkheid van de gebruiker om de beslissing van het systeem te controleren (overschrijven) (Heck en Schouten 2023). Het onderzoek van Heck en Schouten 2023 past het bijvoorbeeld toe door de gebruiker de categorie van de bloem die het systeem herkent, te laten aanpassen.

Cloud vendor Een manier om data op te slaan is via een cloud vendor (Laato e.a. 2022). Volgens een respondent in dat onderzoek, is de rol van een cloud vendor even belangrijk als die van een elektriciteitsleverancier (Laato e.a. 2022). Volgens andere respondenten is het gebruik van een cloud vendor niet in alle gevallen wenselijk, het kan namelijk dataveiligheidsproblemen met zich meebrengen (Laato e.a. 2022).

Explainability, verklaarbaarheid Verklaarbaarheid heeft betrekking op de interne werking van het ML-systeem volgens Fagbola en Thakur 2019. Het ondersteunt de interpreteerbaarheid ervan (Fagbola en Thakur 2019). Het impliceert daarnaast ook dat er uitleg

gegeven wordt aan de gebruiker over hoe het systeem tot een beslissing is gekomen, waardoor het vertrouwen van de gebruiker stijgt (Fagbola en Thakur 2019; Habiba e.a. 2022; Heck en Schouten 2023; Li en Han 2023). Ook volgens Kästner 2022 is verklaarbaarheid de mate waarin het model bruikbare verklaringen voor zijn voorspellingen (gegeven een bepaalde input) kan geven. In dat onderzoek wordt eveneens aangehaald dat een verhoogde verklaarbaarheid kan zorgen voor meer vertrouwen in het model (Kästner 2022). Volgens Li en Han 2023 helpt het verklaarbaar maken van een ML-systeem ook bij het identificeren en verminderen van *bias*. In Habiba e.a. 2022 wordt de onenigheid over de exacte definitie van zowel interpreteerbaarheid als verklaarbaarheid in het onderzoekslandschap benadrukt. Ook in Li en Han 2023 wordt er gesproken over hoe verklaarbaarheid een andere definitie kan hebben, afhankelijk van de stakeholder.

Collaboration effectiveness, effectiviteit in het samenwerken Een andere vereiste uit het onderzoek van Heck en Schouten 2023 is de effectiviteit in het samenwerken tussen het systeem en de gebruiker. Het wordt gedefinieerd als volgt: het systeem zal effectief samenwerken met de gebruiker om de vereiste taken uit te voeren (Heck en Schouten 2023). Een voorbeeld van die effectieve samenwerking is bijvoorbeeld het feit dat het systeem de correcties van de gebruiker direct meeneemt in de hertraining (Heck en Schouten 2023).

Human autonomy, menselijke autonomie Tot slot definiëert Heck en Schouten 2023 menselijke autonomie als vereiste. Dat wil zeggen dat het systeem de gebruikers niet zal bedreigen in hun autonomie (Heck en Schouten 2023). Het wordt geïllustreerd door het voorbeeld dat biologen, door de komst van automatisatie door ML-systemen, zich meer kunnen focussen op innovatieve projecten in plaats van repetitieve taken (Heck en Schouten 2023) .

Regulatory compliance, naleven van de regelgeving Regelgeving is een belangrijke vereiste om rekening mee te houden voor veel AI-systemen, denk bijvoorbeeld aan GDPR die een beperking legt op het hanteren van data (Laato e.a. 2022). Deze vereiste heeft niet enkel betrekking op wetgeving, maar ook op andere regulering en beleid, waardoor het belangrijk is om de regelgeving in de toepasselijke projectcontext goed te begrijpen (Laato e.a. 2022).

Trustworthiness, betrouwbaarheid Betrouwbaarheid wordt volgens Ronanki 2023 gedefinieerd als het gebruik en de implementatie van een ML-systeem dat de opgelegde regulering volgt en rekening houdt met ethische principes. Ook volgens Zhang e.a. 2021 is het rekening houden met ethische principes een eigenschap van betrouwbaarheid. Betrouwbaarheid betekent dat de output van het ML-systeem vertrouwd kan worden (Zhang e.a. 2021). Wat daarvoor nodig is, is volgens Zhang e.a. 2021 menselijke interventie, robuustheid, veiligheid, data governance en privacy, transparantie en interpreteerbaarheid.

In Ahmad, Bano e.a. 2021 wordt betrouwbaarheid samengenomen met verklaarbaarheid (explainability). Verklaarbaarheid zou betrouwbaarheid in veel gevallen bevorderen, omdat er bijvoorbeeld verklaard kan worden waarom het ML-systeem een bepaalde voorspelling maakte. Een gelijkaardige mening delen de auteurs van Fagbola en Thakur 2019, die vinden

dat het begrijpen van de redenering achter voorspellingen helpt bij het inschatten van de betrouwbaarheid ervan.

Inference latency Deze vereiste meet de duurtijd van het maken van één predictie (Kästner 2022). Stel dat je een ML-systeem bouwt dat als taak heeft om frauduleuze transacties van creditcards te detecteren (Kästner 2022). In dat geval is het belangrijk dat de latency onder enkele seconden blijft (Kästner 2022).

Inference throughput Een maatstaf die aanduidt hoeveel predicties er binnen een bepaalde tijd gemaakt kunnen worden (Kästner 2022). In het voorbeeld van de fraudedetectie op creditcardtransacties is throughput erg belangrijk omdat er vaak grote volumes aan transacties zijn (Kästner 2022).

Training latency Deze vereiste geeft aan hoe lang het duurt om een model te (her)trainen (Kästner 2022).

Model size, grootte van het model De grootte van het model wordt vaak gemeten aan de hand van de bestandsgrootte van het model (Kästner 2022). Het is belangrijk om op te merken dat de grootte van het model ook bepaalt hoeveel gegevens verzonden moeten worden bij een update van het model, maar het bepaalt ook de opslag die het model vereist (Kästner 2022). Het GPT-3 model van OpenAI heeft bijvoorbeeld 700 gigabyte opslagruimte nodig (Kästner 2022).

Energy consumption, energieverbruik In sommige scenario's is het belangrijk om het energieverbruik per predictie zo laag mogelijk te houden (Kästner 2022). Denk bijvoorbeeld aan smartphones, die afhankelijk zijn van de beperkte batterijcapaciteit (Kästner 2022). Daarnaast kost het trainen van ML-systemen veel energie en stoot de hoeveelheid aan nodige rekenkracht eveneens CO2 uit (Kästner 2022). Het trainen van diepe neurale netwerken kan een enorme hoeveelheid aan CO2 uitstoten (Kästner 2022).

Deterministic predictions, deterministische voorspellingen Wanneer een model deterministische voorspellingen toepast, wil dat zeggen dat het altijd dezelfde voorspelling zal doen wanneer dezelfde input gegeven wordt (Kästner 2022).

Scalability, schaalbaarheid Schaalbaarheid meet hoe de verwerkingscapaciteit van het systeem opgeschaald kan worden (Kästner 2022). Dat opschalen wordt vaak gedaan door middel van een cloud infrastructuur (Kästner 2022).

Cost, kost De gebruikte hardware, de grootte van het model, het gebruik van GPU's en hoge latency- en throughputvereisten (Kästner 2022). Het zijn allemaal factoren die de kostprijs van het ML-systeem beïnvloeden (Kästner 2022). Die wordt vaak uitgedrukt in kost per voorspelling (Kästner 2022).

3.2.2 Functionele vereisten

Documenting assumptions, documenteren van assumpties In Gjorgjevikj e.a. 2023 worden enkele vereisten voorgesteld die te maken hebben met het documentatieproces. Ten eerste zouden gemaakte assumpties gedocumenteerd moeten worden (Gjorgjevikj e.a. 2023). Een voorbeeld van een assumptie kan zijn: de assumptie dat de statistische eigenschappen

over de gehele dataset gelijk zijn (Gjorgjevikj e.a. 2023). Andere assumpties kunnen te maken hebben met de trainingdata, de *loss*-functie, het optimalisatiealgoritme... Het identificeren en documenteren van deze assumpties zorgt ervoor dat ze niet over het hoofd gezien worden tijdens het ontwikkelingsproces en dat de effecten ervan gemonitord kunnen worden (Gjorgjevikj e.a. 2023).

Documenting dependencies, documenteren van afhankelijkheden Naast het documenteren van assumpties, is het eveneens belangrijk om afhankelijkheden te documenteren (Gjorgjevikj e.a. 2023). Afhankelijkheden zijn volgens Gjorgjevikj e.a. 2023 externe factoren of componenten waarvan het systeem afhankelijk is, maar waarvan het beheer buiten de controle van het systeem ligt. ML-systemen zijn onder andere afhankelijk van de data die ze gebruiken en van de veelvuldig gebruikte software *libraries* om het model te bouwen (Gjorgjevikj e.a. 2023).

Specification of system architecture, systeemarchitectuur bepalen In het onderzoek van Ranawana en Karunananda 2021 wordt het belang van technische documentatie tijdens het ontwikkelen van een ML-systeem benadrukt. Ten eerste is het belangrijk om te weten in welke systeemarchitectuur het ML-systeem of component zal werken (Ranawana en Karunananda 2021). Ten tweede worden tijdens het ontwikkelen van de softwaremodules de user interface, het ML-framework en de nodige modules gedefinieerd en geïmplementeerd (Ranawana en Karunananda 2021). Tot slot worden ook de inputs van het ML-systeem ontworpen (Ranawana en Karunananda 2021).

Data augmentation, verrijken van data Veel ML algoritmes vereisen een grote hoeveelheid aan data om optimaal te werken (Almajalid e.a. 2018). Vaak is de hoeveelheid data echter beperkt, waardoor ervoor gekozen wordt om de dataset te verrijken (Almajalid e.a. 2018). Dat wil zeggen dat de hoeveelheid dataobjecten vergroot wordt (Almajalid e.a. 2018). Een respondent in het onderzoek van Vogelsang en Borg 2019 verrijkte een dataset bijvoorbeeld door data van andere bronnen toe te voegen.

Retraining frequency, frequentie van hertrainen Het onderhouden van een ML-systeem vereist hertraining (Vogelsang en Borg 2019). Hertraining wordt uitgevoerd zodat het ML-systeem zich kan aanpassen aan recente data (Vogelsang en Borg 2019; Wan e.a. 2020). Het is daarom volgens Vogelsang en Borg 2019 belangrijk dat er wordt bepaald wanneer het systeem hertraind zal worden, evenals hoe dat zal gebeuren. Volgens het onderzoek van Nakamichi e.a. 2020 vereist 34 procent van meer dan 400 onderzochte ML-systemen, hertraining. 77 procent daarvan moet minstens maandelijks hertraind worden (Nakamichi e.a. 2020).

Determine the algorithm, algoritme bepalen Er moet een algoritme gekozen worden dat gebruikt zal worden in het ML-systeem (Polyakov e.a. 2018). Vaak wordt er daarvoor gekeken naar de data en de verdeling daarvan (Polyakov e.a. 2018). Er zijn veel verschillende algoritmen waaruit gekozen kan worden, afhankelijk van de taak van het systeem (Polyakov e.a. 2018).

3.3 Vereisten capteren en documenteren in ML-projecten: huidige methoden

Deze sectie tracht een overzicht te geven van actuele methoden, die in de wetenschappelijke literatuur voorgedragen worden, om vereisten te capteren en/of documenteren in ML-projecten. De methoden die worden aangehaald in het onderzoek van Ahmad, Abdelrazek e.a. 2023 worden als eerste toegelicht. Daarna worden methoden die in andere onderzoeken werden vermeld, toegelicht.

3.3.1 Methoden volgens Ahmad, Abdelrazek e.a. 2023

Het onderzoek van Ahmad, Abdelrazek e.a. 2023 betreft het *requirements engineering* proces dat doorlopen wordt om *human-centered* AI-systemen te ontwikkelen. *Human-centered* slaat op de vereisten die een gebruiker van een AI-systeem verwacht zoals eerlijkheid, vertrouwen en verklaarbaarheid (Ahmad, Abdelrazek e.a. 2023). Het is één van de weinige onderzoeken naar methodologie waarin het belang van die 'nieuwe' vereisten opgenomen wordt.

61 respondenten namen deel aan de enquête waaruit enkele opmerkelijke *business practices* naar voren komen (Ahmad, Abdelrazek e.a. 2023). De manieren waarop vereisten worden bijgehouden of neergeschreven variëren van Microsoft Office producten tot het ticketingsysteem JIRA en DOORS (een IBM product om vereisten te managen) (Ahmad, Abdelrazek e.a. 2023).

De respondenten in het onderzoek van Ahmad, Abdelrazek e.a. 2023 leerden ons eveneens dat volgende documentatie- of captatiemethoden toegepast worden op AI-projecten in de industrie:

- GORE
- Use cases
- User stories
- UML
- Informele vereisten (video, audio...)
- Practical Action Research Design Method

Drie van de 61 respondenten in dat onderzoek gaven aan dat ze vereisten niet documenteren (Ahmad, Abdelrazek e.a. 2023). De hierboven opgelijste methoden werden gebruikt als een onderzoeksbasis en zullen hieronder toegelicht worden.

GORE i* Goal-Oriented Requirements Engineering (GORE) gebruikt doelen (*goals*) om vereisten te verzamelen, modelleren, analyseren, structureren, onderhandelen en documenteren (Horkoff e.a. 2016; Van Lamsweerde 2000). Vereisten worden in deze methode dus opgesteld vanuit het perspectief de vooropgestelde doelen (*goals*) te behalen (Van Lamsweerde 2000). Intussen is GORE een verzamelnaam geworden voor verschillende raamwerken om met vereisten om te gaan (Kavakli 2002). In verschillende onderzoeken wordt GORE dan ook voorgesteld als een manier om vereisten te capteren in ML-projecten (Ahmad, Abdelrazek e.a. 2023; Ishikawa en Matsuno 2020; Khan e.a. 2022).

Het onderzoek van Barrera e.a. 2024 betreft een i* extensie van GORE, specifiek voor ML-requirements. GORE i* extensies zijn raamwerken die op een hoog niveau van abstractie opereren, en ze zijn daarom een wijdverspreide manier om gevalsspecifieke methodologieën op te stellen (Barrera e.a. 2024).

Het voorgestelde i* raamwerk houdt, aan de hand van een vragenlijst, rekening met zowel functionele als niet-functionele vereisten (Barrera e.a. 2024). De vragen uit die lijst worden gerelateerd aan vijf fases van het ontwikkelingsproces (Barrera e.a. 2024):

1.	High-level definitie van het doel
2.	Definiëren van succes aan de hand van metrieken
3.	Niet-functionele vereisten
4.	Kenmerken en beperkingen van de dataset
5.	Validatie van het model

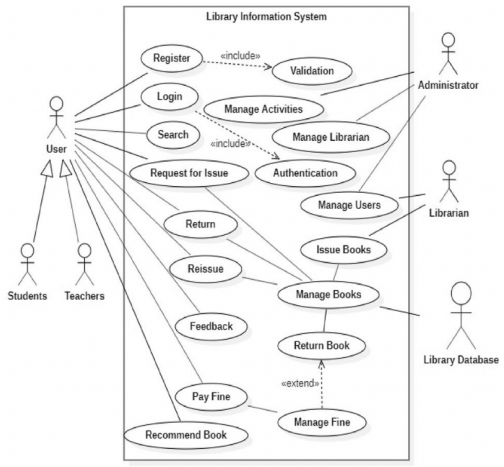
Tabel 2: (Barrera e.a. 2024)

Belangrijk om op te merken is het feit dat er in fase drie (niet-functionele vereisten) specifiek gevraagd wordt naar de nood aan verklaarbaarheid, hertraining en unbiasedness (Barrera e.a. 2024). Niet-functionele vereisten die buiten die set vallen, worden gecovered door een algemene vraag: "Is er sprake van andere kwaliteitsvereisten waaraan het ML-model moet voldoen?"(Barrera e.a. 2024). Refereer naar figuur 9 in de appendix voor de volledige set vragen uit het onderzoek van Barrera e.a. 2024.

Daarnaast bevat de voorgestelde methode een tabel en een vergelijking om het geschikte ML-algoritme te selecteren (Barrera e.a. 2024). De opgenomen algoritmes zijn de meest populaire ML-algoritmes voor classificatie, regressie en clustering (Barrera e.a. 2024). Enige uitbreiding kan dus nodig zijn indien er gebruik wordt gemaakt van een minder bekend algoritme (Barrera e.a. 2024). Volgende limitaties van de i* methode worden aangehaald: *reinforcement learning* en de aspecten van *explainable* AI worden niet ondersteund (Barrera e.a. 2024).

Use cases Een use case is volgens Cockburn 2000: ”een collectie van mogelijke sequenties aan interacties tussen het betreffende systeem en zijn gebruikers, gerelateerd aan een bepaald doel.” Kort gezegd beschrijven use cases de interactie tussen gebruiker en systeem (Dalpiaz en Sturm 2020). Use cases maken deel uit van de Unified Modeling Language

Figuur 2: Use Case Diagram Arif e.a. 2023



(Dalpiaz en Sturm 2020). Refereer naar de hoofding 'UML' voor verdere uiteenzetting van de Unified Modeling Language.

Het doel van use cases is om vereisten op te bouwen (Cockburn 2000). Volgens het onderzoek van Dalpiaz en Sturm 2020 worden use cases typisch gebruikt om functionele vereisten uit te drukken, maar bestaan er ook uitbreidingen om kwalitatieve (in dit onderzoek niet-functionele) vereisten uit te drukken. De taken die een gebruiker reeds uitvoert of wil uitvoeren met behulp van het te ontwikkelen systeem, kunnen in use cases beschreven worden (Nuseibeh en Easterbrook 2000). Een voorbeeld van een use case wordt in het onderzoek van Cockburn 2000 gegeven: een use case voor een verzekeraar kan als primaire actor de verzekerde persoon betreffen. Die verzekerde persoon heeft als doel betaald te worden voor een auto-ongeval. De use case ziet er dan als volgt uit:

1. De verzekerde persoon dient een claim in met voldoende bewijs.
2. De verzekeraar verifieert of de polis van de verzekerde geldig is.
3. De verzekeraar wijst een verzekeringsagent toe om de zaak te onderzoeken.
4. De verzekeringsagent verifieert of alle details binnen de richtlijnen van de polis vallen.
5. De verzekeraar betaalt de verzekerde persoon.

(Cockburn 2000)

Aan bovenstaande use case kunnen verschillende extensies toegevoegd worden (bijvoorbeeld: het geleverde bewijs is onvoldoende), evenals variaties (de verzekerde partij betreft een persoon, de overheid of een andere verzekeraar) (Cockburn 2000). Al die aspecten worden samengenomen in een use case diagram (Dalpiaz en Sturm 2020). Een voorbeeld van een use case diagram uit het onderzoek van Arif e.a. 2023 wordt gegeven in figuur 2.

User stories Een user story is een semi-gestructureerde manier om vereiste functionaliteit te beschrijven vanuit het perspectief van de gebruiker (Amna en Poels 2022; Dalpiaz en Sturm 2020; Raharjana e.a. 2021). Het wordt semi-gestructureerd genoemd vanwege de voorgeschreven vorm die gebruikt kan worden: ”as [WHO], I want/want to/need/can/would like [WHAT], so that [WHY].”(Raharjana e.a. 2021). WHO (wie) beschrijft de gebruiker die de vereiste wenst, WHAT is hetgene van het systeem verwacht wordt en WHY is het belang van de vereiste (Raharjana e.a. 2021). Een voordeel van user stories is het feit dat ze begrijpbaar zijn voor veel stakeholders wegens het gebruik van geschreven taal (Raharjana e.a. 2021). Daarnaast is het bewezen dat ze helpen bij het monitoren van voortgang tijdens het ontwikkelingsproces (Amna en Poels 2022). Volgens Raharjana e.a. 2021 kan juist die geschreven taal zorgen voor ambiguïteit, inconsistentie en onvolledigheid. Ook volgens Amna en Poels 2022 is ambiguïteit een probleem bij het gebruik van user stories. Het kan ervoor zorgen dat er twijfel of meerdere interpretaties van de vereisten ontstaan (Amna en Poels 2022).

Use cases vs user stories In het onderzoek van Dalpiaz en Sturm 2020 wordt de effectiviteit van use cases en user stories vergeleken aan de hand van een experiment. Daaruit werd geconcludeerd dat een beperking van use cases het volgende kan zijn: actoren worden vaak eenmalig vermeld aan het begin van de use case, daardoor kan het moeilijker zijn om wederkerende actoren of entiteiten te identificeren (Dalpiaz en Sturm 2020). In een user story wordt daarentegen de actor in elke instantie vermeld (in het ”as [WHO]” gedeelte) (Dalpiaz en Sturm 2020; Raharjana e.a. 2021). In het onderzoek van Dalpiaz en Sturm 2020 worden er geen nadelen van user stories expliciet vermeld. Volgens Raharjana e.a. 2021 en Amna en Poels 2022 zijn die er wel, zoals in de alinea hierboven vermeld.

UML UML staat voor Unified Modelling Language (Koç e.a. 2021). Dat is een verzamelnaam voor verschillende modelleringstechnieken, zoals use cases en user stories (Dalpiaz en Sturm 2020). Het laat toe een systeem op een visuele manier te definiëren en documenteren (Koç e.a. 2021). Het is de standaard geworden om vereisten te modelleren in software design (Arif e.a. 2023; Glinz 2000; Ozkaya 2019). UML biedt dus niet enkel een manier om vereisten te modelleren, maar ook een veelvuldigheid aan visuele diagrammen om systeemarchitectuur en software design te modelleren (Ozkaya 2019).

Over het algemeen wordt UML opgedeeld in twee aspecten: statische diagrammen en dynamische diagrammen, ook wel structurele- en gedragsmodellen genoemd (Arif e.a. 2023; Ozkaya 2019). Het statische of structurele diagram tracht vanzelfsprekend de systeemstructuur te modelleren aan de hand van onder andere use cases en *class*-diagrammen (Arif e.a. 2023; Ozkaya 2019). Dynamische of gedragsmodellen kunnen worden opgesteld aan de hand van, onder andere, *state machine*-diagrammen of *activity*diagrammen (Arif e.a. 2023; Ozkaya 2019).

UML heeft zijn populariteit deels te danken aan de grote hoeveelheid tools die er bestaat ter ondersteuning van de modelleringstaal (Ozkaya 2019). Het onderzoek van Glinz 2000 toont meerdere tekortkomingen van UML, waaronder het feit dat UML het gedrag van

subsystemen niet kan modelleren, alsook het feit dat het onvoldoende ondersteuning biedt om om te gaan met de interactie tussen verschillende use cases.

Video en audio Een veelvuldig gebruikte methode om vereisten te capteren is het uitvoeren van interviews (Jirotka en Luff 2006). In het onderzoek van Jirotka en Luff 2006 werden er videobeelden gemaakt van de werknemers tijdens hun dagelijks werk. Eén onderzoeker stelde ondertussen gerichte vragen om zo de vereisten nog beter te kunnen capteren (Jirotka en Luff 2006). Audio is dus een cruciaal element. Het gebruik van audiovisuele technieken tijdens het verzamelen van vereisten zorgt voor een erg gedetailleerd begrip van de situatie en een verhoogd inlevingsvermogen (Jirotka en Luff 2006).

Practical Action Research Design Method Voor de Practical Action Research Design Method werden er geen wetenschappelijke bronnen gevonden in de gebruikte databases. In het originele onderzoek van Ahmad, Abdelrazek e.a. 2023 wordt er eveneens niet uitgeweid over de inhoud van de methode.

Machine Learning Canvas Een andere mogelijkheid om vereisten expliciet te maken en te documenteren, is het Machine Learning Canvas (MLC) van Louis Dorard (Ahmad, Abdelrazek e.a. 2023; Tun e.a. 2021). Hoewel het MLC niet aangehaald werd door respondenten in het onderzoek van Ahmad, Abdelrazek e.a. 2023, bracht dat onderzoek deze methode wel naar voren als deel van de huidige literatuur. Het gebruiken van een ML Canvas zorgt ervoor dat software engineers, data scientists en ML-specialisten kunnen samenwerken tijdens het documenteren van het project en de vereisten (Ahmad, Abdelrazek e.a. 2023). Het bevat secties met betrekking tot de taak van het ML-systeem, de data die nodig is, de modellen die gebouwd zullen worden, de *value proposition* (waarde voor de eindgebruiker)... (Louis Dorard 2015).

3.3.2 Methoden uit ander onderzoek

GR4ML Het onderzoek van Nalchigar e.a. 2021 valideert een reeds opgesteld GORE raamwerk voor het modelleren van vereisten in ML-projecten: Goal-Oriented Requirements Engineering for Machine Learning (GR4ML). Het raamwerk bestaat uit drie componenten waarbij ieder component het perspectief van een stakeholder dient te vertegenwoordigen: *business*-stakeholders, data scientists en data engineers (Nalchigar e.a. 2021).

Het *business*-perspectief, ook wel de *Business View* genoemd, tracht *business goals* te vertalen in belissings- en vragendoelen (Nalchigar e.a. 2021). Het is de bedoeling te achterhalen hoe ML de vragendoelen kan beantwoorden (Nalchigar e.a. 2021). Een voorbeeld hiervan kan zijn: de *business goal* is een procesapplicatie maken, het beslissingsdoel is een beslissing omtrent een kredietaanvraag te maken en het vragendoel is: "wat zal het risicogehalte van de huidige applicant zijn?"(Nalchigar e.a. 2021). Het is dan de bedoeling dat het ML-model een antwoord geeft op dat vragendoel (Nalchigar e.a. 2021). Om die beslissings- en vragendoelen te extraheren, worden *User Stories* voorgesteld.

Het perspectief van de data scientists wordt opgenomen in de *Analytics Design View* (Nalchigar e.a. 2021). Daarin wordt een ML-oplossing gemodelleerd (Nalchigar e.a. 2021).

Zowel het algoritme als niet-functionele vereisten als KPI's worden opgenomen in de Analytics Design View (Nalchigar e.a. 2021). Dat ziet er volgens het lopende voorbeeld zo uit: de taak die uitgevoerd moet worden is de classificatie van een applicant. Die taak kan uitgevoerd worden door middel van verschillende ML-algoritmes: Naïve Bayes, Decision Trees...(Nalchigar e.a. 2021) De niet-functionele vereisten die in dit voorbeeld belangrijk zijn worden evenens gemodelleerd: robuustheid, interpreteerbaarheid en tolerantie van ontbrekende waarden (Nalchigar e.a. 2021). Tot slot worden KPI's zoals precisie en accuraatheid gemodelleerd (Nalchigar e.a. 2021). De invloed van de algoritmen op de niet-functionele vereisten wordt eveneens gemodelleerd door middel van pijlen die aanduiden of de interpreteerbaarheid zal stijgen of dalen door het gebruik van een bepaald algoritme (Nalchigar e.a. 2021).

Het derde en laatste perspectief, is dat van de data engineers (Nalchigar e.a. 2021). Het raamwerk schrijft de *Data Preparation View* voor om dat perspectief mee te nemen (Nalchigar e.a. 2021). Dat component tracht de nodige taken met betrekking tot het voorbereiden van data te modelleren (Nalchigar e.a. 2021). Concreet worden de nodige tabellen en kolommen gemodelleerd, evenals de operaties die op de ruwe data moeten gebeuren.

Na een case study kon geconcludeerd worden dat de GR4ML methode verschillende nieuwe (informatie)vereisten blootlegde en communicatie tussen de projectleden bevorderde (Nalchigar e.a. 2021). De *Business Model View* werd daarentegen als tijdsintensief ervaren, net zoals dat een gebrek aan indicatie van volledigheid werd ervaren (Nalchigar e.a. 2021). Dat gebrek aan volledigheid deed zich voor tijdens het opstellen van de use cases die de vragendoelen moeten coveren (Nalchigar e.a. 2021). Het bleek moeilijk te zijn om in te schatten of alle vragen een use case hadden gekregen (Nalchigar e.a. 2021). Er blijkt ook nood te zijn aan, onder andere, meer richtlijnen met betrekking tot het effectief modelleren, het opstellen van vragendoelen en het eventueel overslaan van een view (Nalchigar e.a. 2021).

Proof of Concept Een Proof of Concept (PoC) maken is een proces waarin data reeds verzameld wordt en de geschiktheid van het toekomstige ML-systeem al dan niet bevestigd wordt (Nakamichi e.a. 2020). De nodige technologie en user scenario's worden eveneens mee in rekening genomen om een volledig beeld te scheppen van het te ontwikkelen systeem (Nakamichi e.a. 2020). Volgens Nakamichi e.a. 2020 is het gebruikelijk om een Proof of Concept op te stellen om te anticiperen op het moeilijk te voorspellen gedrag van het ML-systeem. Het is volgens dat onderzoek moeilijk om in de beginfase van het project vereisten vast te leggen aan de hand van documentatie, dus bied een PoC een alternatief (Nakamichi e.a. 2020).

GORE-MLOps In het onderzoek van Khan e.a. 2022 wordt GORE-MLOps (volgens het onderzoek van Ishikawa en Matsuno 2020) als modelleringstechniek voor vereisten voorgesteld. GORE-MLOps is een grafische modelleertaal die technieken van *Directed Acyclic Graphs* en *User Requirements Notation* toepast om subdoelen en taken van een systeem, dat een functioneel doel tracht te behalen, te modelleren (Ishikawa en Matsuno

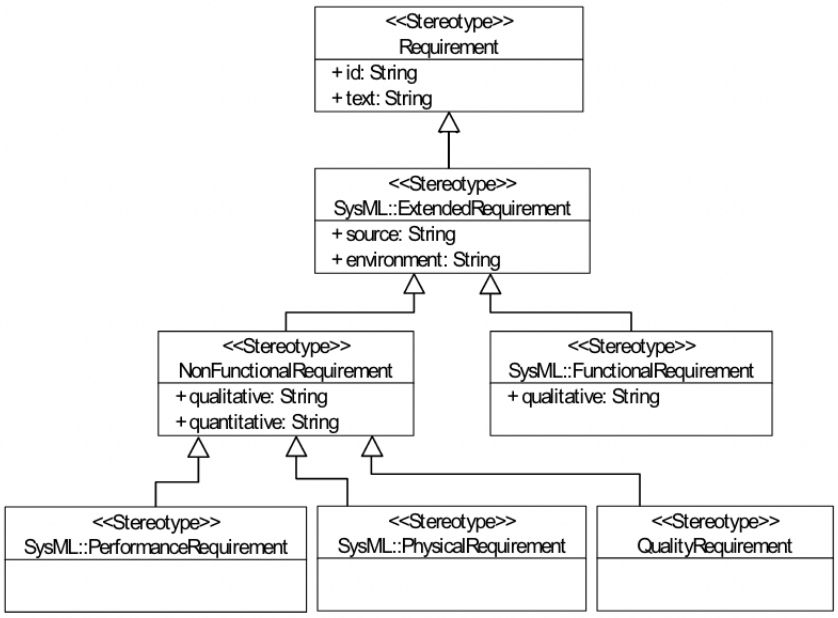
2020). Een belangrijk kenmerk van GORE-MLOps is het feit dat verschillende mogelijkheden en de impact daarvan ook gemodelleerd worden (Ishikawa en Matsuno 2020).

Daarnaast worden er elementen aan dat model toegevoegd die de haalbaarheid en/of het bewijs van het behalen van een vereiste weergeven in numerieke termen (Ishikawa en Matsuno 2020). Een voorbeeld kan zijn: het functionele doel is de classificatie van defecte producten, waaraan meerdere subdoelen gekoppeld zijn (volledig automatisch, minder menselijke interventie...) (Ishikawa en Matsuno 2020). Aan die subdoelen wordt indien nodig de haalbaarheid gekoppeld, samen met het feit of die haalbaarheid gevalideerd werd of niet (Ishikawa en Matsuno 2020). Refereer naar appendix A.3 voor een voorbeeld van het grafisch model (Goal Graph) volgens Ishikawa en Matsuno 2020.

SysML SysML is in esssentie een grafische modelleertaal die vereisten aan elkaar en aan testcases relateert (Gruber e.a. 2017). Het maakt gebruik van *requirement diagrams*, die de relatie tussen verschillende vereisten visualiseren (Gruber e.a. 2017). Ook in dat onderzoek worden vereisten onderverdeeld in de categorieën functionele en niet-functionele vereisten (Gruber e.a. 2017). Daarnaast wordt er gebruik gemaakt van verschillende andere subcategorieën zoals performantie- of kwaliteitsvereisten (Gruber e.a. 2017). In SysML kunnen vereisten afgeleid worden van een andere vereiste, kan er een uni- of bidirectionele relatie bestaan tussen twee vereisten of een *containment* relatie (Gruber e.a. 2017). Die *containment* relatie betekent dat de vereiste enkel voldaan kan zijn indien alle *contained* vereisten voldaan zijn (Gruber e.a. 2017). Volgens Khan e.a. 2022 is deze methode onvoldoende effectief om niet-functionele vereisten in ML projecten te modelleren, maar werd er reeds een toevoeging getest. Over die toevoeging werd niet gerapporteerd in het onderzoek van Khan e.a. 2022.

Refereer naar figuur 3 voor een voorbeeld van een SysML requirement diagram (Gruber e.a. 2017).

Mentaal model In het onderzoek van Aslam e.a. 2023 worden er enkele klassieke methoden voorgedragen om een *explainable*, verklaarbaar, AI-systeem te bekomen, zoals *scenario-based design, question banks* en cocreatie-technieken. Daarnaast wordt het belang van cognitieve modellen, in het onderzoek mentale modellen genoemd, onderzocht en benadrukt (Aslam e.a. 2023). Een mentaal model wordt gedefinieerd als de cognitieve structuur die het wereldbeeld van een persoon vormt, gebaseerd op ervaring, perceptie en begrip (Aslam e.a. 2023). Het evalueren en begrijpen van de mentale modellen van eindgebruikers kan ervoor zorgen dat de verklaarbaarheidsvereisten van de eindgebruiker meer accuraat gerepresenteerd worden (Aslam e.a. 2023). Er wordt getracht de factoren die het mentaal model van de gebruiker beïnvloeden, te bestuderen om betere vereisten op te stellen (Aslam e.a. 2023).



Figuur 3: SysML Gruber e.a. 2017

4 Resultaten

In deze sectie worden de resultaten van zowel de literatuurstudie als de validatieinterviews toegelicht. Ten eerste wordt er een meta-analyse uitgevoerd van de vereisten zoals ze in de literatuur voorkomen. Het is daarbij de bedoeling dat de definities verduidelijkt worden door de bestaande literatuur te aggregeren tot eenduidige definities om die daarna te valideren. De meta-analyse wordt voor zowel niet-functionele als functionele vereisten uitgevoerd. Daaropvolgend wordt het gevalideerde raamwerk voorgesteld. Dat raamwerk bevat de geaggregeerde definities die gebaseerd zijn op de academische literatuur met daaraan toegevoegd de mening van de respondenten. Tot slot wordt er een korte toelichting gegeven van de interviewresultaten met betrekking tot de vereisten en de documentatie- en captatiemethoden.

4.1 Meta-analyse van niet-functionele vereisten

Er bestaan een aantal overlappingen en discrepanties met betrekking tot de definitie van niet-functionele vereisten in de literatuur, waarbij dezelfde vereiste soms op verschillende of gelijkaardige manieren wordt gedefinieerd.

In deze sectie worden die overlappingen en discrepanties expliciet gemaakt om vervolgens keuzes te maken teneinde éénduidige definities te vormen. Die keuzes worden telkens gemotiveerd. Om de overlap te visualiseren werd er een matrix opgesteld (figuur 4) die toont welke term er in dit onderzoek gevalideerd zal worden en hoe die door andere auteurs gecategoriseerd werd. De vereisten gevolgd door een asterisk (*) worden gebruikt voor de definitie van die vereiste in dit onderzoek. De vereisten die gevolgd worden door een obelisk

(†), werden door de auteur onder dezelfde term gedefinieerd, maar werden niet opgenomen in onze definitie. De termen die op één rij staan, overlappen in definitie.

Daarnaast toont figuur 5 alle niet-functionele vereisten waarbij er geen overlapping of discrepantie aanwezig was. Indien definities van meerdere auteurs voor eenzelfde vereiste overeenkomen, staan de termen op dezelfde rij.

4.1.1 Generaliseerbaarheid vs. robuustheid

Er is een overlap tussen de term generaliseerbaarheid en robuustheid. In dit onderzoek wordt robuustheid gedefinieerd als de mogelijkheid om met ongeziene data en andere omgevingsomstandigheden om te gaan (Gjorgjevikj e.a. 2023; Heck en Schouten 2023; Hu e.a. 2020). Een belangrijke toevoeging is die van Kästner 2022: de output van het systeem wijzigt niet bij een lichte wijziging van de input.

De definitie die Heck en Schouten 2023 hanteert voor generaliseerbaarheid wordt in dit onderzoek ook onder de vereiste 'robuustheid' opgenomen. Die keuze werd gemaakt omdat de definitie die in het onderzoek van Heck en Schouten 2023 gehanteerd wordt voor generaliseerbaarheid, een subset is van de definitie die wij hanteren voor robuustheid. De definitie die Zhang e.a. 2021 hanteren, wordt in dit onderzoek niet opgenomen in de definitie van robuustheid om verwarring en erg brede definities te vermijden.

4.1.2 Traceerbare paden vs. verantwoordelijkheid

De vereiste 'verantwoordelijkheid' wordt in dit onderzoek gedefinieerd door een combinatie van de definitie van Rahimi e.a. 2019 voor traceerbare paden en de definitie van Zhang e.a. 2021 voor verantwoordelijkheid. Die keuze werd gemaakt omdat in beide definities de basis ligt bij de mogelijkheid om beslissingen van het ML-systeem te traceren.

4.1.3 Accuraatheid vs. nauwkeurigheid van het model

De definitie van accuraatheid is in dit onderzoek een combinatie van de definitie van accuraatheid door Horkoff 2019 en de definitie van 'nauwkeurigheid van het model' uit het onderzoek van Heck en Schouten 2023. Die keuze werd gemaakt omdat de nauwkeurigheid van het model accuraatheid impliceert; als het model geen correcte regels aanleert is het onwaarschijnlijk dat de output correct is ten opzichte van de werkelijkheid.

4.1.4 Interpreteerbaarheid vs verklaarbaarheid

Interpreteerbaarheid en verklaarbaarheid overlappen op verschillende aspecten van de definitie. Interpreteerbaarheid volgens Fagbola en Thakur 2019 is de mogelijkheid van mensen om, gegeven inputparameters, te kunnen voorspellen wat de output van het systeem zal zijn. Terwijl verklaarbaarheid volgens Fagbola en Thakur 2019 betrekking heeft op het kunnen begrijpen van de interne werking van het systeem. Verklaarbaarheid impliceert volgens

Fagbola en Thakur 2019; Habiba e.a. 2022; Heck en Schouten 2023; Kästner 2022; Li en Han 2023 eveneens dat er uitleg gegeven wordt aan de gebruiker over hoe het systeem tot zijn output is gekomen. Een aspect van interpreteerbaarheid volgens Habiba e.a. 2022 is het feit dat mensen de oorzaak van de output kunnen begrijpen. In dit onderzoek zal de combinatie van verklaarbaarheid volgens Fagbola en Thakur 2019; Habiba e.a. 2022; Heck en Schouten 2023; Kästner 2022; Li en Han 2023 als definitie van verklaarbaarheid gehanteerd worden. Die keuze werd gemaakt omdat beide definities in de kern de oorzaak van de beslissing willen verklaren aan de gebruiker.

Interpreteerbaarheid zal in dit onderzoek gedefinieerd worden als de definitie van interpreteerbaarheid volgens Gjorgjevikj e.a. 2023; Habiba e.a. 2022; Kästner 2022. Het zwaartepunt ligt daarbij bij het begrijpen van de werking van het model. Het onderzoek van Fagbola en Thakur 2019 haalt daarbij aan dat een gebruiker gegeven enkele inputs de output moet kunnen voorspellen om interpreteerbaar te zijn. Het is belangrijk op te merken dat dat aspect niet opgenomen wordt in de definitie van interpreteerbaarheid in dit onderzoek.

4.1.5 Verklaarbaarheid vs transparantie

De definitie die Horkoff 2019 en Zhang e.a. 2021 aan transparantie geven, komt overeen met de definitie van verklaarbaarheid die in dit onderzoek gehanteerd zal worden. Daarom zullen die definities niet opgenomen worden in de definitie van transparantie, maar wel in die van interpreteerbaarheid.

4.1.6 Transparantie

Voor de definitie van transparantie werd er in dit onderzoek voor gekozen het merendeel van de definities in de literatuur niet op te nemen in de eigen definitie. Het enige aspect dat wordt opgenomen in de eigen definitie is dat van Fagbola en Thakur 2019 en Gjorgjevikj e.a. 2023. Volgens beide auteurs kan transparantie op zowel het volledige model, de componenten en/of het trainingsalgoritme toegepast worden. Dat aspect wordt wel meegenomen in de definitie die dit onderzoek zal hanteren.

De andere definities zoals gegeven door Fagbola en Thakur 2019; Horkoff 2019; Zhang e.a. 2021, worden reeds opgenomen in de definitie van verklaarbaarheid. In dit onderzoek zal transparantie gedefinieerd worden als de mogelijkheid om de werking van een ML-systeem te kunnen 'zien', onafhankelijk van de verklaarbaarheid of interpreteerbaarheid ervan.

4.1.7 Menselijke autonomie

In het onderzoek van Heck en Schouten 2023 werd menselijke autonomie als vereiste opgenomen. In dit onderzoek wordt er echter voor gekozen die vereiste te laten vallen. Die keuze werd gemaakt omdat de verantwoording in dat onderzoek geen vereiste van een ML-project (volgens onze definitie) vormt.

4.1.8 Veiligheid

De definitie die Gjorgjevikj e.a. 2023 voorschrijft lijkt eerder op een definitie van zekerheid. Wegens de ambiguïteit van die definitie, zal het niet opgenomen worden in dit onderzoek. Er werd besloten veiligheid te definiëren volgens het onderzoek van Fagbola en Thakur 2019; Kuwajima en Ishikawa 2019 en Zhang e.a. 2021.

4.1.9 Eerlijkheid vs. zonder vooroordelen

De definities van eerlijkheid en *unbiasedness* overlappen deels. Eerlijkheid wordt door Fagbola en Thakur 2019 als synoniem voor *unbiasedness* gezien. Daarnaast wordt eerlijkheid gedefinieerd als de afwezigheid van vooroordeel en favoritisme ten opzichte van individuen of groepen (Gjorgjevikj e.a. 2023). *Unbiasedness* wordt eveneens gezien als de afwezigheid van discriminatie op sociaal of economisch gebied (Heck en Schouten 2023; Zhang e.a. 2021). Alle bovenstaande definities zullen in dit onderzoek onder het overkoepelend begrip 'zonder vooroordelen' opgenomen worden.

4.1.10 Zonder vooroordelen vs. accuraatheid

Ook overlapt 'zonder vooroordelen' volgens de definitie van Laato e.a. 2022 deels met de definitie van accuraatheid van Horkoff 2019, waarin gezegd wordt dat de output van het systeem correct is ten opzichte van de werkelijkheid. In de definitie van *biasedness* volgens Laato e.a. 2022 gaat het over het afwijken van de output van het normale. Die definities zullen niet opgenomen worden onder het begrip 'zonder vooroordelen' (*unbiasedness*) in dit onderzoek, omdat correctheid ten opzichte van de werkelijkheid geen objectiviteit impliceert.

4.1.11 Betrouwbaarheid vs. robuustheid, veiligheid, transparantie, interpreteerbaarheid

Tot slot is de definitie van betrouwbaarheid een samenkomst van verschillende andere niet-functionele vereisten. Volgens Ronanki 2023 overlapt het met het volgen van de regelgeving en ethiek. Volgens Zhang e.a. 2021 vereist betrouwbaarheid onder andere robuustheid, veiligheid, transparantie en interpreteerbaarheid. Daarnaast impliceert betrouwbaarheid volgens Ahmad, Bano e.a. 2021 verklaarbaarheid. In dit onderzoek zal betrouwbaarheid gedefinieerd worden aan de hand van de definitie van Ronanki 2023 en Zhang e.a. 2021. De definitie van Ahmad 2021 zal niet onder de noemer betrouwbaarheid gecategoriseerd worden in dit onderzoek vanwege het feit dat enkel verklaarbaarheid niet noodzakelijk leidt tot een betrouwbaar systeem.

	Horkoff 2019	Fagbola & Thakur2019	Habiba e.a 2022	Gjorgjevikj e.a. 2019	Heck en Schouten 2023	Zhang e.a. 2021	T. Li & Han 2023	Rahimi e.a. 2019	Ranawana e.a. 2021	Hu e.a. 2020	Ahmad 2021	Ronanki 2023	Laato e.a. 2022	Kästner 2022
Robuustheid				Robuustheid*	Robuustheid*					Robuustheid*				
					Generaliseerbaarheid*									
						Robuustheid†								Robuustheid*
Verantwoordelijkheid						Verantwoordelijkheid*		Traceerbare paden*						
Accurraatheid	Accurraatheid*				Nauwkeurigheid model*									
Verklaarbaarheid	Transparantie*	Verklaarbaarheid*	Verklaarbaarheid*	Interpreteerbaarheid†	Verklaarbaarheid*	Transparantie*	Verklaarbaarheid*							Verklaarbaarheid*
		Transparantie†												
			Interpreteerbaarheid*											
Transparantie	Transparantie†	Verklaarbaarheid†	Verklaarbaarheid*	Verklaarbaarheid†	Verklaarbaarheid†	Transparantie†	Verklaarbaarheid†							
		Transparantie†		Transparantie†										
Interpreteerbaarheid		Interpreteerbaarheid*		Interpreteerbaarheid*										Interpreteerbaarheid*
			Interpreteerbaarheid*											
		Interpreteerbaarheid†												
Unbiasedness		Eerlijkheid*		Eerlijkheid*	Unbiasedness*	Unbiasedness*								
	Accurraatheid†												Unbiasedness†	
Betrouwbaarheid						Betrouwbaarheid*						Betrouwbaarheid*		
											Verklaarbaarheid†			

Figuur 4: Matrix overlappende niet-functionele

	Horkoff 2019	Faghois & Thakur 2019	Habibie a. 2022	Gjorgevikj a. 2019	Heck en Schouten 2022	Zhang a. 2021	Wan a. 2020	Rahimi a. 2019	Iannone a. a. 2021	Hu a. a. 2020	Ahmad 2021	Ronanki 2023	Lastdrag a. 2022	Dey 2022	Heyn a. a. 2021	Kinkér en Melis 2016	Vogelsang en Borg 2019	Thinyane a. a. 2020	Kuwajima a. a. 2019	Kästner 2022
Evaluatie						Evaluatie*	Evaluatie*													
Datavereisten						Datavereisten*					Datavereisten*			Datavereisten*	Datavereisten*	Datavereisten*				
Data versioning													Data versioning*							
Datatransparantie						Datatransparantie*														
Dataconsistentie										Dataconsistentie*							Dataconsistentie*			
Correctheid van data																	Correctheid van data*			
Data labeling							Data labeling*										Data labeling*			
Exploratieve data-analyse																	Exploratieve data-analyse*			
Beschermde attributen																	Beschermde attributen*			
Afstemmen van het model							Afstemmen van het model*													
Ethiek		Ethiek*									Ethiek*							Ethiek*		
Privacy	Privacy*			Privacy*	Privacy*														Privacy*	
Veiligheid		Veiligheid*		Veiligheid*		Veiligheid*													Veiligheid*	
Generaliseerbaarheid						Generaliseerbaarheid*														
Controleerbaarheid						Controleerbaarheid*														
Cloud vendor													Cloud vendor*							
Effectiviteit in het samenwerken						Effectiviteit in het samenwerken*														
Menselijke autonomie						Menselijke autonomie*														
Inference latency																			Inference latency*	
Inference throughput																			Inference throughput*	
Training latency																			Training latency*	
Grootte van het model																			Grootte van het model*	
Deterministische voorspellingen																			Deterministische voorspellingen*	
Schaalbaarheid																			Schaalbaarheid*	
Kost																			Kost*	
Naleven regelgeving													Naleven regelgeving*							

Figuur 5: Matrix overige niet-functionele vereisten

4.2 Meta-analyse van functionele vereisten

De definities van functionele vereisten overlappen niet met elkaar. Daarom zal dit onderzoek de definities valideren die in sectie *3.2.2 Functionele vereisten* gegeven worden. Een figuur werd opgesteld om een overzicht te verschaffen van de functionele vereisten zoals ze in de literatuur voorkomen (figuur 6).

Term	Auteur(s)
Documenteren assumpties	Gjorgjevikj e.a. 2019
Documenteren afhankelijkheden	Gjorgjevikj e.a. 2019
Systeemarchitectuur bepalen	Ranawana e.a. 2021
Verrijken van data	Almajalid e.a. 2018, Vogelsang en Borg 2019
Frequentie hertraining	Vogelsang en Borg 2019
Algoritme bepalen	Polyakov e.a. 2018

Figuur 6: Matrix functionele vereisten

4.3 Raamwerk

Onderstaande sectie geeft een overzicht van de definities die in dit onderzoek voorgesteld worden als relevant bij het uitvoeren van machine learning projecten. De definities worden gerangschikt op basis van in- en output. De inputvereisten zijn de vereisten die te maken hebben met de systeemconfiguratie en data. De outputvereisten zijn vereisten waaraan de output van het model moet voldoen. Deze definities kwamen tot stand door de meta-analyse van de literatuur, alsook de resultaten van de validatie-interviews. Het getal naast de vereiste geeft het aantal citaties van de onderzoeken die die vereisten definiëren weer. Dat zorgt voor een indicatie van adoptie van de vereiste. Refereer naar grafiek 8 voor een visualisatie van het aantal citaties per vereiste.

4.3.1 Niet-functionele vereisten: inputvereisten

Transparantie, transparency (15) In dit onderzoek zal transparantie gedefinieerd worden als de mogelijkheid om de werking van een ML-systeem te kunnen 'zien', onafhankelijk van de verklaarbaarheid of interpreteerbaarheid ervan. Transparantie kan toegepast worden op drie niveau's, volgens het onderzoek van Fagbola en Thakur 2019 en Gjorgjevikj e.a. 2023: het volledige model, de componenten en/of het trainingsalgoritme. Uit interviews blijkt dat transparantie weinig waarde toevoegt voor de gebruiker zolang er geen interpreteerbaarheid of verklaarbaarheid aan wordt gekoppeld, wegens het technische karakter van transparantie.

Verantwoordelijkheid, accountability (7) Verantwoordelijkheid van een ML-systeem wordt gedefinieerd als volgt: alle componenten en acties van een ML-systeem zijn traceerbaar en verantwoordelijk (Zhang e.a. 2021). Bijvoorbeeld: een zelfrijdende auto moet een log bijhouden van alle acties uitgevoerd door zowel de bestuurder als het autonome systeem (Zhang e.a. 2021). Het belang van zo'n log werd bevestigd in de interviews. Daarnaast is het opvolgen van de overeenstemming tussen de vooropgestelde richtlijnen met betrekking tot design en programmeren, mogelijk (Rahimi e.a. 2019).

Veiligheid, safety (52) Een veilig ML-systeem zorgt ervoor dat het gebruik door mensen veilig is (Fagbola en Thakur 2019). Daarnaast impliceert het ook een robuust systeem dat bestendig is tegen aanvallen (Kuwajima en Ishikawa 2019). Het is ook belangrijk dat het systeem accuraat, betrouwbaar en reproduceerbaar is volgens het onderzoek van Kuwajima en Ishikawa 2019. Veiligheid kan in het gedrang worden gebracht door externe aanvallen zoals *data poisoning* (Zhang e.a. 2021).

Robuustheid, robustness (30) In dit onderzoek zal robuustheid gedefinieerd worden als de mogelijkheid om met ongeziene data en andere omgevingsomstandigheden om te gaan (Gjorgjevikj e.a. 2023; Heck en Schouten 2023; Hu e.a. 2020). Ook de definitie volgens Kästner 2022 wordt hierin opgenomen: "robuustheid geeft aan in welke mate de voorspellingen van een model stabiel blijven wanneer de input licht gewijzigd wordt". De regels die het ML-model leert uit de trainingsdata, zullen ook gelden voor nieuwe data (Heck en Schouten 2023). Volgens een respondent is robuustheid belangrijk om een goede

accuraatheid te behalen. Het voorbeeld dat die respondent aanhaalde is het volgende: wanneer je een foto van een madeliefje overdag of 's nachts met een flash uploadt in een tool, moet die tool in beide gevallen het madeliefje kunnen herkennen.

Deterministische voorspellingen, deterministic predictions (0) Wanneer een model deterministische voorspellingen toepast, wil dat zeggen dat het altijd dezelfde voorspelling zal doen wanneer dezelfde input gegeven wordt (Kästner 2022). Volgens twee respondenten kan het verkrijgen van exact dezelfde output na het ingeven van exact dezelfde input in sommige gevallen belangrijk zijn. Een voorbeeld dat werd gegeven door een respondent is het labelen van foto's in een *computer vision*-model: wanneer je tweemaal dezelfde foto geeft aan het model, wil je de tweede keer dezelfde labels zien als de eerste keer. In andere gevallen, zoals bij ChatGPT haalde diezelfde respondent aan, wil je dat de output licht wijzigt en de voorspellingen dus niet deterministisch zijn.

Datavereisten, data requirements (63) Wegens de grote hoeveelheid aan kwalitatieve data die nodig is om een werkend ML-systeem te bouwen, zijn volgende datavereisten belangrijk: beschikbaarheid, kwaliteit, selectie en het testen van data (Ahmad, Bano e.a. 2021; Zhang e.a. 2021). Het is eveneens belangrijk om te bepalen welke data opgeslagen moet worden (Kinkiri en Melis 2016).

Data labeling Er wordt een label toegekend dat voor elk dataobject aanduidt wat het is (Wan e.a. 2020). Wanneer data gelabeld wordt door mensen, kan je er bijna zeker van zijn dat de data bevooroordeeld is (Vogelsang en Borg 2019). In een domein zoals gezondheidszorg, vergt het labelen van data een expert (Pareek e.a. 2022). Let wel: een respondent gaf aan dat die menselijke invloed op het labelen van data zelden voor een *bias* probleem zorgt.

Afstemmen van het model, model tuning (192) De parameters die het model gebruikt, worden aangepast en eventuele problemen in het model worden geïdentificeerd (Wan e.a. 2020).

Data versioning (20) Wanneer er meerdere modellen getraind worden in een ML-project kunnen er problemen optreden die moeilijk traceerbaar zijn (Laato e.a. 2022). Het is daarom belangrijk dat er bijgehouden wordt welk model met welke data getraind werd (Laato e.a. 2022). Ook in de praktijk blijkt dit een belangrijke vereiste te zijn, volgens een respondent is het bijhouden van configuraties cruciaal wanneer er meerdere modellen worden getest.

Cloud vendor (20) Een manier om data op te slaan is om dat in een cloud te doen, de rol van een cloud vendor is daarom belangrijk (Laato e.a. 2022). Volgens twee respondenten kan het zeker belangrijk zijn om de cloud vendor op voorhand vast te leggen, maar is dat niet altijd zo.

Naleven van de regelgeving, regulatory compliance (20) Wetgeving zoals GDPR en ander beleid, is een belangrijke vereiste waarmee rekening gehouden moet worden voor en door veel AI-systemen (Laato e.a. 2022). Het is daarom belangrijk om de regelgeving in de toepasselijke projectcontext goed te begrijpen (Laato e.a. 2022). Uit de interviews blijkt een bewustzijn van de huidige en evoluerende regelgeving. Dat heeft zich volgens een respondent

reeds geuit in de nood aan additionele documentatie tijdens het ontwikkelingsproces van het systeem. Een andere respondent benadrukt het naleven van de GDPR-wetgeving.

Dataconsistentie, data consistency (216) Een consistente instroom van kwalitatieve data is een vereiste voor een goed-werkend ML-systeem (Ranawana en Karunananda 2021). Wanneer een systeem grote hoeveelheden manueel gelabelde data vereist, is dat moeilijk te verzekeren en begint men met een hoeveelheid waarmee het ML-systeem kan convergeren (Ranawana en Karunananda 2021). Volgens Vogelsang en Borg 2019 is dataconsistentie gerelateerd aan het format en de representatie van de data die hetzelfde moet zijn voor de gehele dataset. Uit de interviews blijkt dat wanneer er onvoldoende gelabelde data aanwezig is, de kost en de duurtijd van een ML-project onredelijk kan worden.

Correctheid van de data, data correctness (201) Het is belangrijk dat de gebruikte data correct is, en dat kan gerelateerd zijn aan de manier waarop de data verzameld werd (Vogelsang en Borg 2019). Uit de interviews blijkt dat het van groot belang is dat de praktische omgeving (de werkelijkheid) weergegeven wordt in de trainingset. Wanneer de praktische omgeving verandert, moet het systeem opnieuw getraind worden.

Correctheid van de data kan een probleem vormen bij het gebruik van publieke datasets. Die zijn volgens Vogelsang en Borg 2019 niet altijd betrouwbaar, omdat er niemand betaald wordt om zo'n grote hoeveelheid aan data te onderhouden.

Beschermde attributen, protected attributes (201) Wanneer het belangrijk is om in het project *bias* te elimineren, moet er nagedacht worden over de attributen die beschermd zullen worden (Vogelsang en Borg 2019). Beschermde attributen zijn attributen in de dataset die niet gebruikt zullen worden tijdens de classificatie die het ML-systeem uitvoert (Vogelsang en Borg 2019).

Training latency (0) *Training latency* meet hoe lang het duurt om een model te (her)trainen (Kästner 2022). De mening van de respondenten over deze vereiste is verdeeld, maar wanneer er een mens nodig is om de hertraining uit te voeren, is het zeker een relevante vereiste.

Schaalbaarheid, scalability (0) Schaalbaarheid meet hoe de verwerkingscapaciteit van het systeem opgeschaald kan worden (Kästner 2022). Dat wordt vaak gedaan door middel van een cloud infrastructuur (Kästner 2022).

Deze vereiste kan gerelateerd worden aan de outputvereisten 'middelen' en 'tijdsduur' die toegelicht worden in sectie 4.3.2. Onder schaalbaarheid kan men namelijk verstaan hoeveel additionele middelen er nodig zijn om een volgend niveau van een bepaalde richttijd te behalen. Een respondent gaf aan dat het in de industrie belangrijk is om een bepaald volume aan data binnen een bepaalde tijd te verwerken, en het systeem dus schaalbaar moet zijn.

Effectief samenwerken, collaboration effectiveness (0) Deze vereiste betekent dat het systeem en de gebruiker kunnen samenwerken door het feit dat het systeem de correcties van de gebruiker direct meeneemt in de hertraining (Heck en Schouten 2023).

Privacy (123) Het gebruik van persoonlijke data tijdens het trainen van een ML-model kan ervoor zorgen dat die gevoelige informatie gestolen wordt (Gjorgjevikj e.a. 2023; Horkoff 2019; Kästner 2022). In het onderzoek van Heck en Schouten 2023 werd er daarom gekozen om privacygevoelige data niet te gebruiken of verzamelen in het ML-systeem.

Deze vereiste werd bevestigd via interviews, met als nuance dat er dan gekozen kan worden voor een on-device ML-systeem, waarbij er geen data verzonden wordt naar een datacenter van een derde partij. Dat zou de privacy van een ML-systeem kunnen verhogen.

Ethiek, ethics (67) De vele mogelijkheden die een ML-systeem met zich meebrengt, kunnen een positieve maar ook een negatieve invloed hebben op de samenleving (Thinyane en Goldkind 2020). Wanneer men rekening houdt met ethiek tijdens het ontwikkelen van een ML-systeem, wordt er bijgedragen aan het globale welzijn van de mensheid (Thinyane en Goldkind 2020). In Ahmad, Bano e.a. 2021 worden de ethische richtlijnen van de Europese Commissie aangehaald. Daarin wordt ethiek gekoppeld aan respect voor de mensheid, het voorkomen van schade, eerlijkheid en verklaarbaarheid (Ahmad 2021). Rekening houden met deze vereiste doet men best vanaf het begin van het project, de trainingdata en validatiedata kan namelijk reeds onethische output veroorzaken (Fagbola en Thakur 2019). In dit onderzoek wordt deze definitie niet beperkt tot de richtlijnen van de Europese Commissie. Uit de interviews blijkt dat ethiek zeker relevant is en dat het evenzeer belangrijk is dat ethiek gereguleerd wordt wegens de ambiguïteit ervan. Een andere respondent vermeldde dat ze tijdens het ontwikkelen nagaan of de data ethisch is.

4.3.2 Niet-functionele vereisten: outputvereisten

Accuraatheid, accuracy (202) Accuraatheid meet de correctheid van de output van het ML-systeem ten opzichte van de werkelijkheid (Horkoff 2022). Maatstaven die voor accuraatheid gebruikt kunnen worden zijn bijvoorbeeld *precision* en *recall*, maar ook *lift* (Horkoff 2022; Vogelsang en Borg 2019). Het impliceert dus dat het ML-systeem correcte regels uit de trainingsdata heeft geleerd, wat in het onderzoek van Heck en Schouten 2023 gedefinieerd wordt als nauwkeurigheid. In dit onderzoek vallen beide definities onder de noemer *accuraatheid*. Een voorbeeld hiervan is dat het systeem iets moet kunnen herkennen (bijvoorbeeld een bloemsoort in een weide), met een minimale zekerheid van 75 procent (Heck en Schouten 2023). Elke respondent vermeldde het belang van *performance*maatstaven zoals accuraatheid tijdens het uitvoeren van ML-projecten.

Verklaarbaarheid, explainability (24) Verklaarbaarheid vraagt dat er uitleg gegeven wordt aan de gebruiker over hoe het systeem tot een beslissing is gekomen, waardoor het vertrouwen van de gebruiker stijgt (Fagbola en Thakur 2019; Habiba e.a. 2022; Heck en Schouten 2023; Kästner 2022; Li en Han 2023). Het ondersteunt de interpreteerbaarheid van het systeem (Fagbola en Thakur 2019). De nadruk ligt in dit onderzoek op de vraag: waarom maakte het systeem de beslissing? Een respondent benadrukte dat het belangrijk is dat de definitie van verklaarbaarheid tussen klant en ontwikkelaar afgestemd wordt, want iets dat verklaarbaar is voor een ontwikkelaar is niet per definitie verklaarbaar voor de gebruiker.

Interpreteerbaarheid, interpretability (3) Wegens het gebrek aan enigheid over de definitie van interpreteerbaarheid, zal het in dit onderzoek verklaard worden als volgt: interpreteerbaarheid betekent dat een aspect, of het gehele ML-model, begrijpbaar is voor een mens (Gjorgjevikj e.a. 2023). Welk aspect dat is wordt in dit onderzoek niet vastgelegd. De toevoeging van Kästner 2022 is hierbij belangrijk: het begrijpen van de interne werking om *debugging* en auditdoeleinden te faciliteren. Een respondent gaf aan dat interpreteerbaarheid, volgens deze definitie, vooral belangrijk is voor het vertrouwen van de gebruiker in de oplossing. Interpreteerbaarheid werd ook in de interviews vaak aangehaald als synoniem voor verklaarbaarheid.

Zonder vooroordelen, unbiasedness (22) Zonder vooroordelen staat voor gelijkheid en onpartijdigheid van het ML-systeem (Fagbola en Thakur 2019). Het is de afwezigheid van vooroordeel of favoritisme ten opzichte van individuen of groepen, gebaseerd op bepaalde inherente of verkregen karakteristieken, naar de definitie van eerlijkheid volgens Gjorgjevikj e.a. 2023. Een ML-systeem is *biased* wanneer de output oneerlijk of discriminerend is tegenover iemand op sociaal of economisch gebied (Heck en Schouten 2023; Zhang e.a. 2021).

Betrouwbaarheid, trustworthiness (8) Betrouwbaarheid wordt in dit onderzoek gedefinieerd als het gebruik en de implementatie van een ML-systeem dat de opgelegde regulering volgt en rekening houdt met ethische principes, naar de definitie van Ronanki 2023. Het impliceert dat de output van het systeem vertrouwd kan worden door de gebruiker (Zhang e.a. 2021). Het vereist, volgens Zhang e.a. 2021 menselijke interventie, robuustheid, veiligheid, data governance en privacy, transparantie en interpreteerbaarheid. Uit interviews blijkt dat betrouwbaarheid vaak teruggeleid wordt naar verklaarbaarheid. Een vraag die vaak voorkomt in de industrie is: wanneer vertrouwt de organisatie de voorspelling? Met als antwoord: wanneer het duidelijk is waarom het systeem een bepaalde voorspelling maakt.

Evaluatie, assessment (199) Een ML-systeem kan zichzelf testen en evalueren op *bias*, of onderworpen worden aan onafhankelijke, externe evaluatie (Zhang e.a. 2021). De haalbaarheid en het bestaan van zelf-evaluatie door het ML-systeem werd betwijfeld in de interviews. Daarnaast kan een ML-systeem geëvalueerd worden op basis van gedefinieerde evaluatieparameters met behulp van testdata (Wan e.a. 2020). Volgens een interviewrespondent is evaluatie cruciaal om aan de klant te kunnen tonen of het product waarde zal leveren of niet.

Middelen, resources (0) Er zijn verschillende middelen die een ML-model kan verbruiken. Denk bijvoorbeeld aan de grootte van het model. De grootte van het model wordt vaak gemeten aan de hand van de bestandsgrootte ervan (Kästner 2022). Het is belangrijk om op te merken dat de grootte van het model ook bepaalt hoeveel gegevens verzonden moeten worden bij een update van het model, maar het bepaalt ook de opslag die het model vereist (Kästner 2022). Tijdens de interviews is de grootte van het model vooral belangrijk gebleken wanneer het model op eigen infrastructuur moet runnen (bijvoorbeeld een laptop). In andere gevallen blijkt de grootte van het model geen limiterende factor.

Daarnaast kan er ook rekening gehouden worden met het energieverbruik van het model. In sommige scenario's is het belangrijk om het energieverbruik per predictie zo laag mogelijk te houden (Kästner 2022). Denk bijvoorbeeld aan smartphones, die afhankelijk zijn van de beperkte batterijcapaciteit (Kästner 2022). Daarenboven kost het trainen van ML-systemen veel energie en stoot het eveneens CO2 uit door de hoeveelheid rekenkracht die daarvoor nodig is (Kästner 2022). Het trainen van diepe neurale netwerken kan dus een enorme hoeveelheid aan CO2 uitstoten (Kästner 2022). Die uitstoot blijkt volgens drie respondenten belang te winnen, hoewel het momenteel moeilijk meetbaar is in de praktijk. Eén respondent haalde aan dat de locatie en het tijdstip van het hertrainen een impact kan hebben op het energieverbruik (bijvoorbeeld 's nachts het model trainen met overgebleven groene energie).

Tot slot brengt een ML-model ook een kost met zich mee: de gebruikte hardware, de grootte van het model, het gebruik van GPU's en hoge latency- en throughputvereisten (Kästner 2022). Het zijn allemaal factoren die de kostprijs van het ML-systeem beïnvloeden (Kästner 2022). Die wordt vaak uitgedrukt in kost per voorspelling (Kästner 2022). De kost van een cloudvendor werd in de interviews aangehaald als een relevante kost.

Tijdsduur (0) Er zijn verschillende duurtijden waarmee rekening gehouden kan worden tijdens het ontwikkelen van een ML-systeem. *Inference latency* meet de duurtijd van het maken van één predictie (Kästner 2022). Stel dat je een ML-systeem bouwt dat als taak heeft om frauduleuze transacties van creditcards te detecteren, dan is het belangrijk dat de latency onder enkele seconden blijft (Kästner 2022).

Daarnaast is er *inference throughput*, een maatstaf die aanduidt hoeveel predicties er binnen een bepaalde tijd gemaakt kunnen worden (Kästner 2022). In het voorbeeld van de fraudedetectie op creditcardtransacties is throughput erg belangrijk omdat er vaak grote volumes van transacties zijn (Kästner 2022).

Zowel *inference latency* als *inference throughput* werden tijdens de interviews bestempeld als belangrijke vereisten.

4.3.3 Functionele vereisten: inputvereisten

Verrijken van data, data augmentation (315) Veel ML-algoritmes vereisen een grote hoeveelheid aan data om optimaal te werken (Almajalid e.a. 2018). Vaak is de hoeveelheid beschikbare data echter beperkt, waardoor ervoor gekozen wordt om de dataset te verrijken (Almajalid e.a. 2018). Dat wil zeggen dat de hoeveelheid dataobjecten vergroot wordt (Almajalid e.a. 2018). Een respondent in het onderzoek van Vogelsang en Borg 2019 verrijkte een dataset door data van andere bronnen toe te voegen. Een respondent van onze validatie-interviews past dataverrijking toe in *computer vision* projecten door beeldmateriaal te roteren of zelf data te genereren. Merk op: het toevoegen van de kolommen (de dimensionaliteit vergroten) is ook een manier om data te verrijken.

Documentatievereisten, documenting (3) In Gjorgjevikj e.a. 2023 worden enkele vereisten voorgesteld die te maken hebben met het documentatieproces. Ten eerste zouden gemaakte assumpties gedocumenteerd moeten worden (Gjorgjevikj e.a. 2023). Assumpties

worden gemaakt wanneer bepaalde aspecten van het probleem dat moet worden opgelost, of de operationele data, niet waarneembaar zijn (Gjorgjevikj e.a. 2023). Dan worden er assumpties gemaakt om een gepast ML-algoritme te kiezen (Gjorgjevikj e.a. 2023). Een voorbeeld van een assumptie kan zijn: de assumptie dat de statistische eigenschappen over de gehele dataset gelijk zijn (Gjorgjevikj e.a. 2023). Andere assumpties kunnen te maken hebben met de trainingdata, de *loss*-functie, het optimalisatiealgoritme... Het identificeren en documenteren van deze assumpties zorgt ervoor dat ze niet over het hoofd gezien worden tijdens het ontwikkelingsproces en dat de effecten ervan gemonitord kunnen worden (Gjorgjevikj e.a. 2023).

Naast het documenteren van assumpties, is het eveneens belangrijk om afhankelijkheden te documenteren (Gjorgjevikj e.a. 2023). Afhankelijkheden zijn volgens Gjorgjevikj e.a. 2023 externe factoren of componenten waarvan het systeem afhankelijk is, maar waarvan het beheer buiten de controle van het systeem ligt. ML-systemen zijn onder andere afhankelijk van de data die ze gebruiken en van de veelvuldig gebruikte software *libraries* om het model te bouwen (Gjorgjevikj e.a. 2023). Een respondent gaf de afhankelijkheid van datakwaliteit expliciet aan tijdens het interview.

Systeemarchitectuur bepalen, specification of system architecture (15) In het onderzoek van Ranawana en Karunananda 2021 wordt het belang van technische documentatie tijdens het ontwikkelen van een ML-systeem benadrukt. Ten eerste is het belangrijk om te weten in welke systeemarchitectuur het ML-systeem of component zal werken (Ranawana en Karunananda 2021).

Ten tweede worden tijdens het ontwikkelen van de softwaremodules de user interface, het ML-framework en de nodige modules gedefinieerd en geïmplementeerd (Ranawana en Karunananda 2021).

Tot slot worden ook de inputs van het ML-systeem ontworpen (Ranawana en Karunananda 2021).

Tijdens de interviews werd het belang van de integratie van het ML-component in een ander systeem benadrukt. Bij zo'n projecten is het belangrijk na te gaan of de gewenste performantie eveneens behaald wordt na integratie. Denk bijvoorbeeld aan de integratie van een ML-component aan een productielijn. In dat geval kan de controleerbaarheid van de output ook belangrijk zijn. Dat is de mogelijkheid van de gebruiker om de beslissing van het systeem te overschrijven (Heck en Schouten 2023).

Een andere respondent gaf aan dat de systeemarchitectuur alsook de inputs onderworpen zijn aan een iteratief ontwikkelingsproces.

Frequentie van hertrainen, retraining frequency (201) Het onderhouden van een ML-systeem vereist hertraining (Vogelsang en Borg 2019). Hertraining wordt uitgevoerd zodat het ML-systeem zich kan aanpassen aan recente data (Vogelsang en Borg 2019; Wan e.a. 2020). Het is daarom volgens Vogelsang en Borg 2019 belangrijk dat er wordt bepaald wanneer het systeem hertraind zal worden, evenals hoe dat zal gebeuren. Volgens twee

respondenten is het vastleggen van de hertraining erg belangrijk, omdat daar een menselijke factor aan te pas komt. Vaak wordt die hertrainingscyclus contractueel vastgelegd.

Algoritme bepalen, determine the algorithm (201) Er moet een algoritme gekozen worden dat gebruikt zal worden in het ML-systeem (Polyakov e.a. 2018). Vaak wordt er daarvoor gekeken naar de data en de verdeling daarvan (Polyakov e.a. 2018). Er zijn veel verschillende algoritmen waaruit gekozen kan worden, afhankelijk van de taak van het systeem (Polyakov e.a. 2018).

Tabel 3: Het in dit onderzoek voorgestelde raamwerk van gevalideerde vereisten

Niet-functionele vereisten	
1	Accuraatheid
2	Afstemmen van het model
3	Beschermde attributen
4	Betrouwbaarheid
5	Cloud vendor
6	Datacorrectheid
7	Dataconsistentie
8	Data labeling
9	Data versioning
10	Datavereisten
11	Deterministische voorspellingen
12	Evaluatie
13	Ethiek
14	Effectief samenwerken
15	Interpreteerbaarheid
16	Middelen
17	Naleven van de regelgeving
18	Privacy
19	Robuustheid
20	Schaalbaarheid
21	Tijdsduur
22	Transparantie
23	Training latency
24	Veiligheid
25	Verklaarbaarheid
26	Verantwoordelijkheid
27	Zonder vooroordelen
Functionele vereisten	
28	Algoritme bepalen
29	Data verrijken
30	Documentatie
31	Frequentie van hertrainen
32	Systeemarchitectuur bepalen

4.4 Interviewresultaten

Bovenstaande definities bevatten reeds de informatie die gewonnen werd uit de interviews. In deze sectie worden die resultaten expliciet benoemd. Voor de motivatie achter de keuzes die gemaakt werden met betrekking tot het raamwerk, wordt er verwezen naar de discussie.

4.4.1 Interviewresultaten: raamwerk

Het grootste deel van de vereisten kon gevalideerd worden. Het kwam voor dat de respondent een andere term gebruikte om eenzelfde vereiste aan te duiden ('wordt het systeem gedeployed op eigen infrastructuur?' werd onder de vereiste 'systeemarchitectuur bepalen' geplaatst). Enkele vereisten werden meerdere keren gevalideerd omdat het bij toeval gebeurde dat een respondent een in een ander interview gevalideerde vereiste zelf aanhaalde. Refereer naar figuur 7 voor een visualisatie van het aantal validaties per vereiste.

Verschillende respondenten legden de nadruk op het energieverbruik en de kost van een ML-systeem. Er werd dan ook besloten om de vereisten 'energieverbruik' en 'kost' op te nemen onder de vereiste 'middelen', omdat ze beide als een verbruiksmiddel van ML-systemen gezien worden. Ook het naleven van de regelgeving werd meermaals spontaan aangehaald door respondenten, meer specifiek de GDPR en de AI-act van de Europese Unie.

Doch zijn er drie vereisten die niet gevalideerd konden worden door de respondenten: datatransparantie, feature engineering en controleerbaarheid. Datatransparantie kon niet gevalideerd worden omdat de respondent aanhaalde dat dat in de praktijk niet gebruikelijk is. Feature engineering werd door één respondent niet als vereiste gezien, maar als middel om het programmeren eventueel makkelijker te maken. Door een andere respondent die met *computer vision* werkt, werd feature engineering niet gebruikt maar kon die respondent de waarde ervan wel inzien. Controleerbaarheid werd dan weer gezien als niet relevant voor ML, omdat de definitie niet impliceert dat het model ervan bijleert. Door de validatie van de vereiste 'systeemarchitectuur bepalen' werd het duidelijk dat controleerbaarheid wel van belang zou kunnen zijn indien de systeemarchitectuur dat vereist. Daarom werd de vereiste 'controleerbaarheid' onder de vereiste 'systeemarchitectuur bepalen' opgenomen.

Tot slot werd er besloten de vereisten 'inference latency' en 'inference throughput' onder één vereiste te categoriseren, namelijk 'tijdsduur'. De vereisten bleken na de interviews nauw verwant aan elkaar gezien ze beide relateren aan de tijdsduur van één of meerdere predicties.

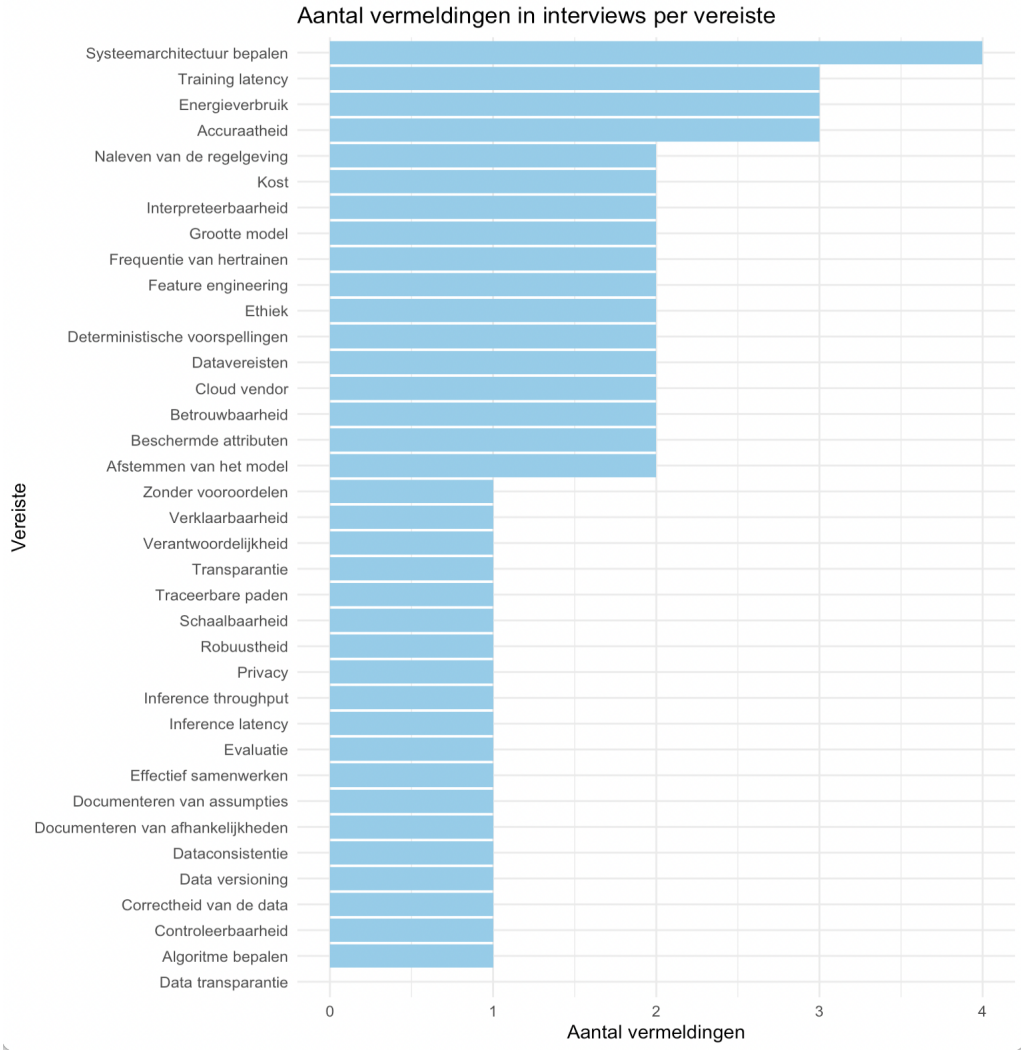
4.4.2 Interviewresultaten: methoden

Tijdens de interviews werd er ook gevraagd naar de captatie- en documentatiemethoden die al dan niet gebruikt worden door de respondenten. In deze sectie zullen de resultaten kort toegelicht worden.

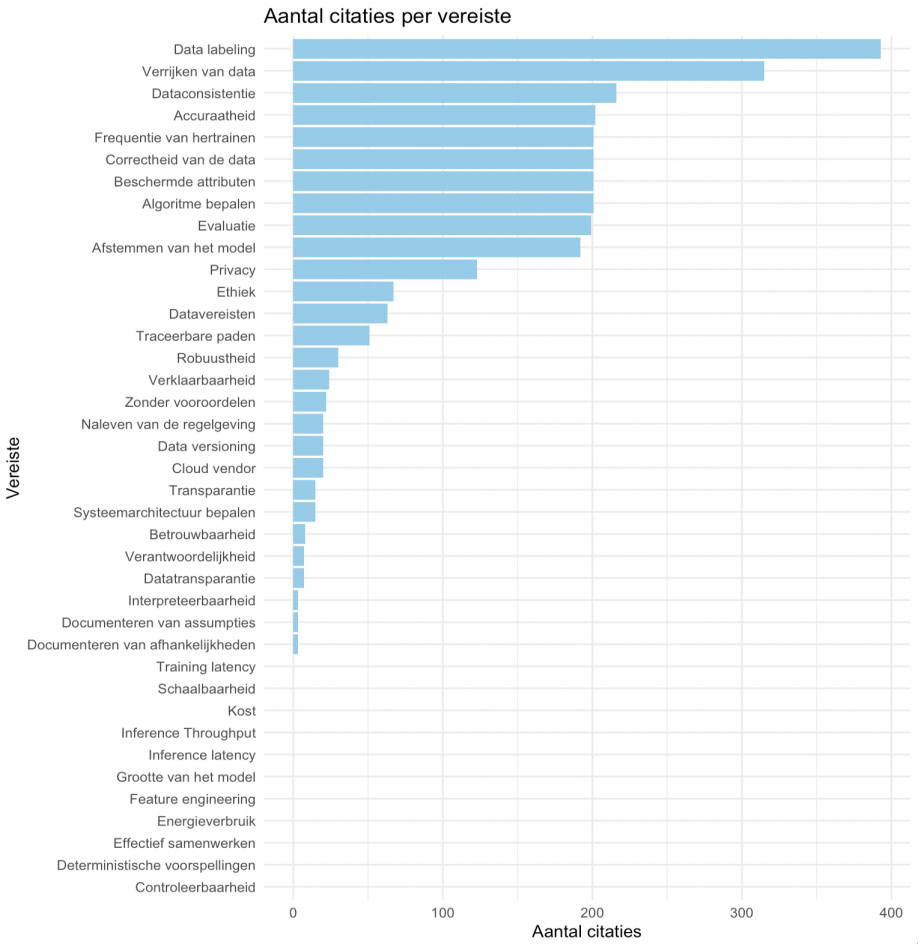
Het capteren en documenteren van vereisten gebeurt in het geval van respondent 3 niet. De andere respondenten doen het capteren verbaal, aan de hand van klantencontact dat vaak plaatsvindt in het presales proces. Respondent 2 gaf aan dat er soms workshops georganiseerd

worden bij de klant om de vereisten te capteren. Het documenteren van die vereisten gebeurt contractueel of niet. Respondenten 1 en 2 leggen bepaalde *performance*maatstaven vast in het contract of in de offerte. Respondent 4 gaf aan use cases te gebruiken, die daarna in PowerPoint slides gegoten worden om te communiceren met de klant.

Drie van de vier respondenten (1, 2 en 4) gaven aan dat ze criteria voor succes (en dus doelen) opstellen waarnaar tijdens de ontwikkeling gestreeft wordt. De metrieken die de respondenten gebruiken gaan van accuraatheid of nauwkeurigheid tot een false-positive rate. Opvallend is het feit dat diezelfde drie respondenten het belang van een iteratief ontwikkelingsproces benadrukken. Twee respondenten gaven eveneens aan met een Proof of Concept te werken. Dat komt volgens elk van hen door de onvoorspelbaarheid van een ML-systeem. Vereisten zijn daarbij van belang om de verwachtingen van de klant te managen maar ook om de noden te vervullen.



Figuur 7: Aantal vermeldingen per vereiste



Figuur 8: Aantal citaties per vereiste

5 Discussie

Door middel van dit onderzoek werd er getracht een raamwerk naar voren te brengen waarnaar gerefereerd kan worden bij het uitvoeren van een ML-project. Volgende onderzoeksvragen zullen in deze sectie beantwoord worden:

- OV 1: Welke vereisten zijn relevant in ML-projecten?
- OV 2: Welke methoden om die vereisten in kaart te brengen, worden reeds toegepast?

5.1 OV 1: Welke vereisten zijn relevant in ML-projecten?

Er blijken een heel aantal nieuwe relevante vereisten te zijn wanneer het om ML-projecten gaat, vergeleken met traditionele softwareontwikkeling. Refereer naar sectie 4.3 voor de volledige oplijsting van de vereisten. Opvallend is dat een aantal van die vereisten (interpreteerbaarheid, betrouwbaarheid, eerlijkheid...) te maken hebben met menselijke interpretatie. Die vereisten blijken moeilijk te definiëren: definities van eenzelfde vereiste, geschreven door

verschillende auteurs verschillen van elkaar, denk bijvoorbeeld aan de definitie van veiligheid volgens Gjorgjevikj e.a. 2023 en die van Fagbola en Thakur 2019. Ook de benaming van de vereisten kon verschillen over de onderzoeken heen: de definitie van generaliseerbaarheid volgens Heck en Schouten 2023 komt overeen met de definitie van robuustheid volgens Gjorgjevikj e.a. 2023. De bevindingen uit de literatuurstudie komen dus overeen met de notie van meerdere auteurs: dat er discussie bestaat omtrent het definiëren van vereisten zoals veiligheid maar ook interpreteerbaarheid en verklaarbaarheid.

Toch blijken de nieuwe vereisten relevant te zijn volgens de respondenten van de interviews. Meerdere van hen gaven aan rekening te houden met de interpreteerbaarheid van een ML-model om het vertrouwen van de gebruiker te verhogen. Andere nieuwe vereisten vloeien voort uit het technische aspect van ML, zoals het kiezen van een geschikt algoritme, maar ook hoe vaak het model hertraind zal worden. Ook die vereisten bleken relevant te zijn.

Naast de relevante vereisten, bleken enkele vereisten niet gevalideerd door de interviews. Datatransparantie werd buiten beschouwing gelaten door het feit dat een respondent vermeldde dat de data die gebruikt wordt, van de klant is. Er is volgens die respondent dus weinig tot geen sprake van het gebruik van publieke datasets in hun projecten. Vaak is enkel data van de klant relevant in een project.

Ook was het de bedoeling om feature engineering als vereiste toe te voegen aan de hand van *grey literature*, maar het belang daarvan werd ontkracht in de interviews. Feature engineering wordt gezien als een middel om een doel te bereiken (de performantie van het systeem te verhogen, de rekentijd te verlagen...), maar niet als vereiste.

Daarnaast werd de vereiste 'controleerbaarheid' opgenomen onder de vereiste 'systeemarchitectuur bepalen' door een validatieinterview. In de definitie van controleerbaarheid werd louter het aanpassen van output vermeld (Heck en Schouten 2023). Een respondent gaf aan dat wanneer het algoritme niet van de outputwijziging leert, de vereiste geen betrekking heeft op ML. Gezien dat het na integratie met bijvoorbeeld een productielijn wel belangrijk kan zijn om de output te controleren of overschrijven, ook al leert het systeem er niet van bij, werd de vereiste niet volledig geëlimineerd.

De vereisten 'energieverbruik' en 'kost' werden ook onder een overkoepelende vereiste opgenomen, namelijk 'middelen'. Beide vereisten worden als relevante verbruiksmiddelen van een ML-systeem gezien. Om die reden werden ze dus onder één vereiste gecategoriseerd.

Een laatste samenvoeging is die van 'inference latency' en 'inference throughput' onder de term 'tijdsduur'. Beide vereisten konden gevalideerd worden maar werden uit efficiëntieoverwegingen onder dezelfde vereiste gecategoriseerd gezien ze beide relateren aan de tijdsduur van één of meerdere predicties.

Er werden uiteindelijk 27 niet-functionele vereisten en vijf functionele vereisten voorgesteld in het raamwerk dat dit onderzoek naar voren brengt. Al die vereisten werden gevalideerd door domeinexperten. Het betreft een zo volledig mogelijk raamwerk met vereisten die relevant (kunnen) zijn tijdens ML-projecten. Een rode draad om als ontwikkelaar

aan vast te houden. Naar het beste van ons weten werd dit nog niet bijgedragen in de actuele literatuur.

5.2 OV 2: Welke methoden om die vereisten in kaart te brengen, worden reeds toegepast?

Hoewel formele documentatie- en captatiemethoden vaak gebruikt worden tijdens het ontwikkelen van traditionele software, blijkt dat niet het geval te zijn in ML-projecten. Literatuur leert ons dat er een tekort is aan ML-specifieke methoden, meer bepaald methoden die rekening houden met de nieuwe, ML-specifieke, vereisten (Ahmad, Abdelrazek e.a. 2023).

In de praktijk is de adoptie van de kleine hoeveelheid specifieke methoden laag, bleek uit de interviews en de literatuur. Men kan zelfs spreken van een afwezigheid van enige formele methodologie bij de respondenten die deelnamen aan dit onderzoek. Wijdverspreide methoden zoals user stories komen zelden voor in de ML-specifieke literatuur en werden door slechts één respondent vermeld. Dit onderzoek bevestigt de hypothese van Yoshioka e.a. 2021: er is een gebrek aan formele kennis omtrent het in kaart brengen van de specifieke vereisten die AI-projecten met zich mee brengen.

Opvallend is het feit dat het capteren van vereisten zowel in de theoretische methoden als die van de respondenten, aan de basis verbale communicatie gebruiken. GORE i*, use cases, user stories... ze vertrekken allemaal van concepten geëxtraheerd van de gebruiker. Ook in de interviews werd het belang van veelvuldige iteratieve communicatie met de klant meermaals benadrukt.

De domeinexperten leerden ons dat een *Proof of Concept* (PoC) een relevante tool is. Dat komt door het feit dat het moeilijk te voorspellen is hoe, en of, het ML-component zal werken en of het de beoogde performantie zal behalen. Na zo'n PoC wordt er dan overgegaan tot eventuele aanscherping, annulatie of verderzetting van het project. Die redenering komt overeen met de voorgestelde theorie van Nakamichi e.a. 2020, waarin het belang van een PoC om precies dezelfde redenen wordt benadrukt.

Een link met de literatuur over de methodologie kan eveneens gelegd worden met GORE, in die methodologie wordt voorgeschreven dat vereisten in de vorm van doelen worden opgesteld. Drie van de vier respondenten gaven aan doelen op te stellen samen met de klant, in de vorm van performancemaatstaven. Die doelen worden vervolgens bij twee van de respondenten contractueel vastgelegd.

Ondanks het gebrek aan een formele methodologie blijkt het belangrijk te zijn rekening te houden met de verschillende vereisten die dit onderzoek definieert.

Door de toelichting van de stand van zaken omtrent huidige methoden, wordt er getracht een basis te verschaffen aan ontwikkelaars die aan de slag willen met het raamwerk.

5.3 Validiteitsrisico

Dit onderzoek werd uitgevoerd met zorg voor de validiteit en repliceerbaarheid ervan. Doch moet erkend worden dat er enige validiteitsrisico's blijven bestaan.

Een eerste risico is de selectie van geraadpleegde databases. Het is mogelijk dat er in andere databases aanvullende literatuur aanwezig is die buiten de literatuur van dit onderzoek valt.

Daarnaast brengen interviews ook validiteitsrisico's met zich mee. Enerzijds is er selection bias: de respondenten werden geselecteerd aan de hand van het beschikbare netwerk van de onderzoeker. De hoeveelheid respondenten werd ook beperkt in dit onderzoek, waardoor additionele inzichten mogelijk zijn bij bevraging van een grotere populatie. Het is dan ook niet de bedoeling de mening van deze vier respondenten te extrapoleren naar de volledige industrie, maar louter het opgestelde raamwerk aan een eerste validatie te onderwerpen.

De profielen van de respondenten zijn redelijk verschillend. Twee respondenten zijn actief in de klantgerichte industrie, één respondent is een onderzoeker en de laatste respondent is co-founder van een startup. Het is belangrijk daarbij te erkennen dat de werkwijze van een startup fundamenteel anders is dan die van een ouder bedrijf. Het is ook mogelijk dat door het interviewen van een team lead de werkelijke praktijken van de uitvoerende data-analisten minder accuraat worden weergegeven.

Tot slot kan de vraagstelling een impact hebben op de gegeven antwoorden.

5.4 Aanbevelingen voor toekomstig onderzoek

Er zijn verschillende manieren om verder te bouwen op dit onderzoek. Ten eerste kan men het voorgestelde raamwerk verder valideren aan de hand van case-studies. In dit onderzoek werd de validatie beperkt tot vier respondenten. Een bevraging bij een groter publiek met meer diverse profielen kan van belang zijn alvorens het raamwerk te implementeren. Denk bijvoorbeeld aan projecten in sectoren waarin veiligheid een cruciale rol speelt.

Daarnaast kan dit raamwerk dienen als onderzoeksbasis voor nieuwe ontwikkelingsmethoden voor ML-projecten. Het overzicht van de relevante vereisten kan een leidraad zijn in het ontwikkelen van een gestructureerde methodologie of het uitbreiden van een bestaande methodologie. Er is momenteel een gebrek aan specifieke captatiemethoden om nieuwe vereisten zoals betrouwbaarheid en verklaarbaarheid effectief te kunnen staven bij stakeholders. Verdere uitbreiding is mogelijk door te onderzoeken hoe de relevante vereisten geïmplementeerd en vervolgens geëvalueerd kunnen worden in werkelijke ML-systemen.

6 Conclusie

Dit onderzoek tracht door middel van een kritische literatuurstudie en validatieinterviews een antwoord te bieden op volgende onderzoeksvragen:

- Welke vereisten zijn relevant in ML-projecten?
- Welke methoden om die vereisten in kaart te brengen, worden reeds toegepast?

Uit de kritische literatuurstudie bleek dat er een groot aantal nieuwe vereisten komen kijken bij het ontwikkelen van ML-systemen. 44 vereisten werden geïdentificeerd uit de academische literatuur. Daaropvolgend werd er eveneens aan de hand van een literatuurstudie antwoord geboden op de tweede onderzoeksvraag. Uit die studie werden er zes ML-specifieke methoden geïdentificeerd (GORE I*, GR4ML, MLC, GORE MLOps, SysML, Mental Models) maar bleek er ook sprake te zijn van andere traditionele methoden om vereisten te capteren en documenteren, zoals user stories, use cases en proof of concept.

De resultaten van de literatuurstudie werden verwerkt door middel van een meta-analyse en validatieinterviews. In de meta-analyse werd getracht ambiguïteit van de definities van vereisten te elimineren. Dat resulteerde in een set vereisten die gevalideerd werd door middel van semi-gestructureerde interviews met vier domeinexperten. Uit die interviews bleek dat de overgrote meerderheid van vereisten relevant is voor de industrie. Er werd dan ook een finaal raamwerk aan vereisten opgesteld dat 27 niet-functionele vereisten en 5 functionele vereisten telt. Refereer naar tabel 3 voor het ontwikkelde raamwerk.

Ook werd er tijdens de validatieinterviews geïnformeerd naar de captatie- en documentatiemethoden die worden toegepast in de industrie. Daaruit bleek dat er geen formele methodologie wordt toegepast in de praktijk en er vooral op *performance*maatstaven wordt afgegaan. Use cases en proof of concept werden wel vermeld door enkele respondenten. Een implicitie link met de Goal-Oriented Requirements Engineering-methodologie kan eveneens worden gelegd.

In dit onderzoek kon er een lijst met gevalideerde en relevante vereisten voorgelegd worden. Daarnaast worden formele methoden om die vereisten te capteren en documenteren, niet (bewust) toegepast in de industrie. Dit onderzoek heeft als doel een rode draad te zijn voor ontwikkelaars en stakeholders van ML-projecten. Toekomstig onderzoek kan het voorgestelde raamwerk valideren bij een grotere steekproef alsook trachten een methodologie op te stellen die steunt op de gevalideerde set vereisten.

Referenties

Abran, A. & Moore, J. W. (Red.). (2001). *Guide to the software engineering body of knowledge: trial version [version 0.95]; SWEBOK*; a project of the Software Engineering Coordinating Commettee*. IEEE Computer Society.

Ahmad, K. (2021). Human-centric Requirements Engineering for Artificial Intelligence Software Systems. *2021 IEEE 29th International Requirements Engineering Conference (RE)*, 468–473. https://doi.org/10.1109/RE51729.2021.00070

Ahmad, K., Abdelrazek, M., Arora, C., Bano, M. & Grundy, J. (2023). Requirements practices and gaps when engineering human-centered Artificial Intelligence systems. *Applied Soft Computing*, *143*, 110421. https://doi.org/10.1016/j.asoc.2023.110421

Ahmad, K., Bano, M., Abdelrazek, M., Arora, C. & Grundy, J. (2021). What's up with Requirements Engineering for Artificial Intelligence Systems? *2021 IEEE 29th International Requirements Engineering Conference (RE)*, 1–12. https://doi.org/10.1109/RE51729.2021.00008

Al Alamin, M. A. & Uddin, G. (2021). Quality Assurance Challenges For Machine Learning Software Applications During Software Development Life Cycle Phases. *2021 IEEE International Conference on Autonomous Systems (ICAS)*, 1–5. https://doi.org/10.1109/ICAS49788.2021.9551151

Almajalid, R., Shan, J., Du, Y. & Zhang, M. (2018). Development of a Deep-Learning-Based Method for Breast Ultrasound Image Segmentation. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1103–1108. https://doi.org/10.1109/ICMLA.2018.00179

Amna, A. R. & Poels, G. (2022). Ambiguity in user stories: A systematic literature review. *Information and Software Technology*, *145*, 106824. https://doi.org/10.1016/j.infsof.2022.106824

Arif, M., Mohammad, C. W. & Sadiq, M. (2023). UML and NFR-framework based method for the analysis of the requirements of an information system. *International Journal of Information Technology*, *15*(1), 411–422. https://doi.org/10.1007/s41870-022-01112-7

Aslam, M., Segura-Velandia, D. & Goh, Y. M. (2023). A Conceptual Model Framework for XAI Requirement Elicitation of Application Domain System. *IEEE Access*, *11*, 108080–108091. https://doi.org/10.1109/ACCESS.2023.3315605

Barrera, J. M., Reina-Reina, A., Lavalle, A., Maté, A. & Trujillo, J. (2024). An extension of iStar for Machine Learning requirements by following the PRISE methodology. *Computer Standards & Interfaces*, *88*, 103806. https://doi.org/10.1016/j.csi.2023.103806

Cockburn, A. (2000). Structuring Use cases with goals.

Dalpiaz, F. & Sturm, A. (2020). Conceptualizing Requirements Using User Stories and Use Cases: A Controlled Experiment [Series Title: Lecture Notes in Computer

Science]. In N. Madhavji, L. Pasquale, A. Ferrari & S. Gnesi (Red.), *Requirements Engineering: Foundation for Software Quality* (pp. 221–238). Springer International Publishing. https://doi.org/10.1007/978-3-030-44429-7_16

Dey, S. (2022). Evidence-driven Data Requirements Engineering and Data Uncertainty Assessment of Machine Learning-based Safety-critical Systems. *2022 IEEE 30th International Requirements Engineering Conference (RE)*, 219–224. https://doi.org/10.1109/RE54965.2022.00027

Fagbola, T. M. & Thakur, S. C. (2019). Towards the Development of Artificial Intelligence-based Systems: Human-Centered Functional Requirements and Open Problems. *2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 200–204. https://doi.org/10.1109/ICIIBMS46890.2019.8991505

Gjorgjevikj, A., Mishev, K., Antovski, L. & Trajanov, D. (2023). Requirements Engineering in Machine Learning Projects. *IEEE Access*, *11*, 72186–72208. https://doi.org/10.1109/ACCESS.2023.3294840

Glinz, M. (2000). Problems and deficiencies of UML as a requirements specification language. *Tenth International Workshop on Software Specification and Design. IWSSD-10 2000*, 11–22. https://doi.org/10.1109/IWSSD.2000.891122

Gruber, K., Huemer, J., Zimmermann, A. & Maschotta, R. (2017). Integrated description of functional and non-functional requirements for automotive systems design using SysML. *2017 7th IEEE International Conference on System Engineering and Technology (ICSET)*, 27–31. https://doi.org/10.1109/ICSEngT.2017.8123415

Habiba, U.-E., Bogner, J. & Wagner, S. (2022). Can Requirements Engineering Support Explainable Artificial Intelligence? Towards a User-Centric Approach for Explainability Requirements. *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, 162–165. https://doi.org/10.1109/REW56159.2022.00038

Heck, P. & Schouten, G. (2023). Defining Quality Requirements for a Trustworthy AI Wildflower Monitoring Platform. *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, 119–126. https://doi.org/10.1109/CAIN58948.2023.00029

Heyn, H.-M., Knauss, E., Muhammad, A. P., Eriksson, O., Linder, J., Subbiah, P., Pradhan, S. K. & Tungal, S. (2021). Requirement Engineering Challenges for AI-intense Systems Development. *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, 89–96. https://doi.org/10.1109/WAIN52551.2021.00020

Horkoff, J. (2019). Non-Functional Requirements for Machine Learning: Challenges and New Directions. *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 386–391. https://doi.org/10.1109/RE.2019.00050

Horkoff, J. (2022). Keynote - Requirements Engineering for Machine Learning: Non-functional Requirements as Core Functions. *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, 141–141. https://doi.org/10.1109/REW56159.2022.00034

Horkoff, J., Aydemir, F. B., Cardoso, E., Li, T., Mate, A., Paja, E., Salnitri, M., Mylopoulos, J. & Giorgini, P. (2016). Goal-Oriented Requirements Engineering: A Systematic Literature Map. *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 106–115. https://doi.org/10.1109/RE.2016.41

Hu, B. C., Salay, R., Czarnecki, K., Rahimi, M., Selim, G. & Chechik, M. (2020). Towards Requirements Specification for Machine-learned Perception Based on Human Performance. *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, 48–51. https://doi.org/10.1109/AIRE51212.2020.00014

IBM. (g.d.). *Get it right the first time: writing better requirements*. https://www.ibm.com/docs/en/SSYQBZ_9.6.1/com.ibm.doors.requirements.doc/topics/get_it_right_the_first_time.pdf

Ishikawa, F. & Matsuno, Y. (2020). Evidence-driven Requirements Engineering for Uncertainty of Machine Learning-based Systems. *2020 IEEE 28th International Requirements Engineering Conference (RE)*, 346–351. https://doi.org/10.1109/RE48521.2020.00046

Jirotka, M. & Luff, P. (2006). Supporting requirements with video-based analysis. *IEEE Software*, *23*(3), 42–44. https://doi.org/10.1109/MS.2006.84

Kallio, H., Pietilä, A.-M., Johnson, M. & Kangasniemi, M. (2016). Systematic methodological review: developing a framework for a qualitative semi [U+2010] structured interview guide. *Journal of Advanced Nursing*, *72*(12), 2954–2965. https://doi.org/10.1111/jan.13031

Kästner, C. (2022). Quality Attributes of ML Components. https://ckaestne.medium.com/quality-drivers-in-architectures-for-ml-enabled-systems-836f21c44334

Kavakli, E. (2002). Goal-Oriented Requirements Engineering: A Unifying Framework. *Requirements Engineering*, *6*(4), 237–251. https://doi.org/10.1007/PL00010362

Khan, A., Siddiqui, I. F., Shaikh, M., Anwar, S. & Shaikh, M. (2022). Handling Non-Fuctional Requirements in IoT-based Machine Learning Systems. *2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 477–479. https://doi.org/10.1109/ECTIDAMTNCON53731.2022.9720403

Kinkiri, S. & Melis, W. J. C. (2016). Reducing data storage requirements for machine learning algorithms using principle component analysis. *2016 International Conference on Applied System Innovation (ICASI)*, 1–4. https://doi.org/10.1109/ICASI.2016.7539804

Koç, H., Erdoğan, A. M., Barjakly, Y. & Peker, S. (2021). UML Diagrams in Software Engineering Research: A Systematic Literature Review. *The 7th International Management Information Systems Conference*, 13. https://doi.org/10.3390/proceedings2021074013

Kuwajima, H. & Ishikawa, F. (2019). Adapting SQuaRE for Quality Assessment of Artificial Intelligence Systems. *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 13–18. https://doi.org/10.1109/ISSREW.2019.00035

Laato, S., Birkstedt, T., Mäantymäki, M., Minkkinen, M. & Mikkonen, T. (2022). AI governance in the system development life cycle: insights on responsible machine learning engineering. *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, 113–123. https://doi.org/10.1145/3522664.3528598

Li, T. & Han, L. (2023). Dealing with Explainability Requirements for Machine Learning Systems. *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 1203–1208. https://doi.org/10.1109/COMPSAC57700.2023.00182

Louis Dorard. (2015). Machine Learning Canvas. https://www.machinelearningcanvas.com/

Louise Barriball, K. & While, A. (1994). Collecting data using a semi [U+2010] structured interview: a discussion paper. *Journal of Advanced Nursing*, *19*(2), 328–335. https://doi.org/10.1111/j.1365-2648.1994.tb01088.x

Nakamichi, K., Ohashi, K., Namba, I., Yamamoto, R., Aoyama, M., Joeckel, L., Siebert, J. & Heidrich, J. (2020). Requirements-Driven Method to Determine Quality Characteristics and Measurements for Machine Learning Software and Its Evaluation. *2020 IEEE 28th International Requirements Engineering Conference (RE)*, 260–270. https://doi.org/10.1109/RE48521.2020.00036

Nalchigar, S., Yu, E. & Keshavjee, K. (2021). Modeling machine learning requirements from three perspectives: a case report from the healthcare domain. *Requirements Engineering*, *26*(2), 237–254. https://doi.org/10.1007/s00766-020-00343-z

Nuseibeh, B. & Easterbrook, S. (2000). Requirements engineering: a roadmap. *Proceedings of the Conference on The Future of Software Engineering*, 35–46. https://doi.org/10.1145/336512.336523

Ozkaya, M. (2019). Are the UML modelling tools powerful enough for practitioners? A literature review. *IET Software*, *13*(5), 338–354. https://doi.org/10.1049/iet-sen.2018.5409

Page, M., McKenzie, J., Bossuyt, P., Boutron, I., Hoffmann, T. & Mulrow, C. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews.

Pareek, A., Lungren, M. P. & Halabi, S. S. (2022). The requirements for performing artificial-intelligence-related research and model development. *Pediatric Radiology*, *52*(11), 2094–2100. https://doi.org/10.1007/s00247-022-05483-8

Pearse, N. (2019). An Illustration of Deductive Analysis in Qualitative Research. https://doi.org/10.34190/RM.19.006

Polyakov, E. V., Mazhanov, M. S., Rolich, A. Y., Voskov, L. S., Kachalova, M. V. & Polyakov, S. V. (2018). Investigation and development of the intelligent voice assistant for the Internet of Things using machine learning. *2018 Moscow Workshop on Electronic*

and Networking Technologies (MWENT), 1–5. https://doi.org/10.1109/MWENT.2018.8337236

Raharjana, I. K., Siahaan, D. & Fatichah, C. (2021). User Stories and Natural Language Processing: A Systematic Literature Review. *IEEE Access*, *9*, 53811–53826. https://doi.org/10.1109/ACCESS.2021.3070606

Rahimi, M., Guo, J. L., Kokaly, S. & Chechik, M. (2019). Toward Requirements Specification for Machine-Learned Components. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 241–244. https://doi.org/10.1109/REW.2019.00049

Ranawana, R. & Karunananda, A. S. (2021). An Agile Software Development Life Cycle Model for Machine Learning Application Development. *2021 5th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*, 1–6. https://doi.org/10.1109/SLAAI-ICAI54477.2021.9664736

Ronanki, K. (2023). Towards an AI-centric Requirements Engineering Framework for Trustworthy AI. *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 278–280. https://doi.org/10.1109/ICSE-Companion58688.2023.00075

Shao, W. & Wang, X. (2022). Modeling Data Requirements for Machine Learning Systems. *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*, 97–100. https://doi.org/10.1109/ICSESS54813.2022.9930317

Thinyane, M. & Goldkind, L. (2020). A Multi-Aspectual Requirements Analysis for Artificial Intelligence for Well-being. *2020 IEEE First International Workshop on Requirements Engineering for Well-Being, Aging, and Health (REWBAH)*, 11–18. https://doi.org/10.1109/REWBAH51211.2020.00008

Tun, H. T., Husen, J. H., Yoshioka, N., Washizaki, H. & Fukazawa, Y. (2021). Goal-Centralized Metamodel Based Requirements Integration for Machine Learning Systems. *2021 28th Asia-Pacific Software Engineering Conference Workshops (APSEC Workshops)*, 13–16. https://doi.org/10.1109/APSECW53869.2021.00013

Van Lamsweerde, A. (2000). Goal-oriented requirements engineering: a guided tour. *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, 249–262. https://doi.org/10.1109/ISRE.2001.948567

Vogelsang, A. & Borg, M. (2019). Requirements Engineering for Machine Learning: Perspectives from Data Scientists. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 245–251. https://doi.org/10.1109/REW.2019.00050

Wan, Z., Xia, X., Lo, D. & Murphy, G. C. (2020). How does Machine Learning Change Software Development Practices? *IEEE Transactions on Software Engineering*, 1–1. https://doi.org/10.1109/TSE.2019.2937083

Westfall, L. (2005). Software requirements engineering: what, why, who, when, and how.

Yoshioka, N., Husen, J. H., Tun, H. T., Chen, Z., Washizaki, H. & Fukazawa, Y. (2021). Landscape of Requirements Engineering for Machine Learning-based AI Systems.

2021 28th Asia-Pacific Software Engineering Conference Workshops (APSEC Workshops), 5–8. https://doi.org/10.1109/APSECW53869.2021.00011

Zhang, T., Qin, Y. & Li, Q. (2021). Trusted Artificial Intelligence: Technique Requirements and Best Practices. *2021 International Conference on Cyberworlds (CW)*, 303–306. https://doi.org/10.1109/CW52790.2021.00058

A Appendices

A.1 Interviewdocumentatie

A.1.1 Interviewvragen

De interviews werden uitgevoerd volgens een semi-gestructureerde interviewmethode. Dat wil zeggen dat er naast de vooropgestelde vragen, ruimte is om meer informatie of verduidelijking te vragen (Louise Barriball en While 1994). Onderstaand worden de interviewvragen weergegeven die tijdens elk interview gesteld werden:

1. Beschrijf uw functie en relatie met machine learning projecten.
2. Omschrijf kort het ontwikkelingsproces dat u doorloopt tijdens het uitvoeren van een ML-project.
    (a) Is het gebaseerd op een theoretisch raamwerk?
3. Wat is uw definitie van 'vereisten' in ML-projecten?
    (a) Op welke manier specificeert u wat de software wel en niet mag of moet doen?
4. Wat is, volgens u, het belang van vereisten in ML-projecten?
    (a) Indien niet belangrijk: hoe houdt u rekening met de wensen van de klant en/of eindgebruiker tijdens het project?
5. In welke categorieën splitst u vereisten op, indien van toepassing?
6. Wat zijn volgens u typische vereisten die voorkomen tijdens een ML-project? Geef een voorbeeld indien mogelijk.
    (a) Hoe capteert u die vereisten?
    (b) Hoe documenteert u die vereisten?
    (c) Zijn er vereisten die volgens u pas laattijdig ontdekt worden?

A.1.2 Codeboek

In dit codeboek wordt er gewerkt volgens deductieve codering. Aan de hand van de voorafgaande literatuurstudie werden er codes opgesteld. Na afloop van de interviews werd de informatie ingedeeld volgens die codes en in sommige gevallen werden er codes toegevoegd.

O	Ontwikkelingsmethoden
C	Captatiemethoden
D	Documentatiemethoden
B	Belang van vereisten
V	Vereiste

Tabel 4: Legende codeboek

Onderstaand wordt de lijst met gebruikte codes weergegeven (tabel 5).

Codeboek			
O1	Ontwikkelingsmethode	V16	Datatransparantie
C1	Proof of Concept	V17	Betrouwbaarheid
C2	Workshops	V18	Transparantie
C3 & D1	GORE	V19	Systeemarchitectuur bepalen
D2	UML	V20	Data versioning
D3	Geen documentatie	V21	Cloud vendor
D4	Informele documentatie	V22	Naleven van de regelgeving
B1	Moeilijk gedrag te voorspellen	V23	Zonder vooroordelen
B2	Noden van de klant vervullen	V24	Verklaarbaarheid
B3	Vertrouwen van de klant formaliseren	V25	Robuustheid
V1	Controleerbaarheid	V26	Traceerbare paden
V2	Effectief samenwerken	V27	Datavereisten
V3	Feature engineering	V28	Ethiek
V4	Inference latency	V29	Privacy
V5	Inference throughput	V30	Afstemmen van het model
V6	Training latency	V31	Evaluatie
V7	Grootte model	V32	Correctheid van de data
V8	Energieverbruik	V33	Beschermde attributen
V9	Deterministische voorspellingen	V34	Frequentie van hertrainen
V10	Schaalbaarheid	V35	Algoritme bepalen
V11	Kost	V36	Accuraatheid
V12	Interpreteerbaarheid	V37	Dataconsistentie
V13	Documenteren van assumpties	V38	Verrijken van data
V14	Documenteren van afhankelijkheden	V39	Data labeling
V15	Verantwoordelijkheid		

Tabel 5: Codeboek gebruikt voor interviews

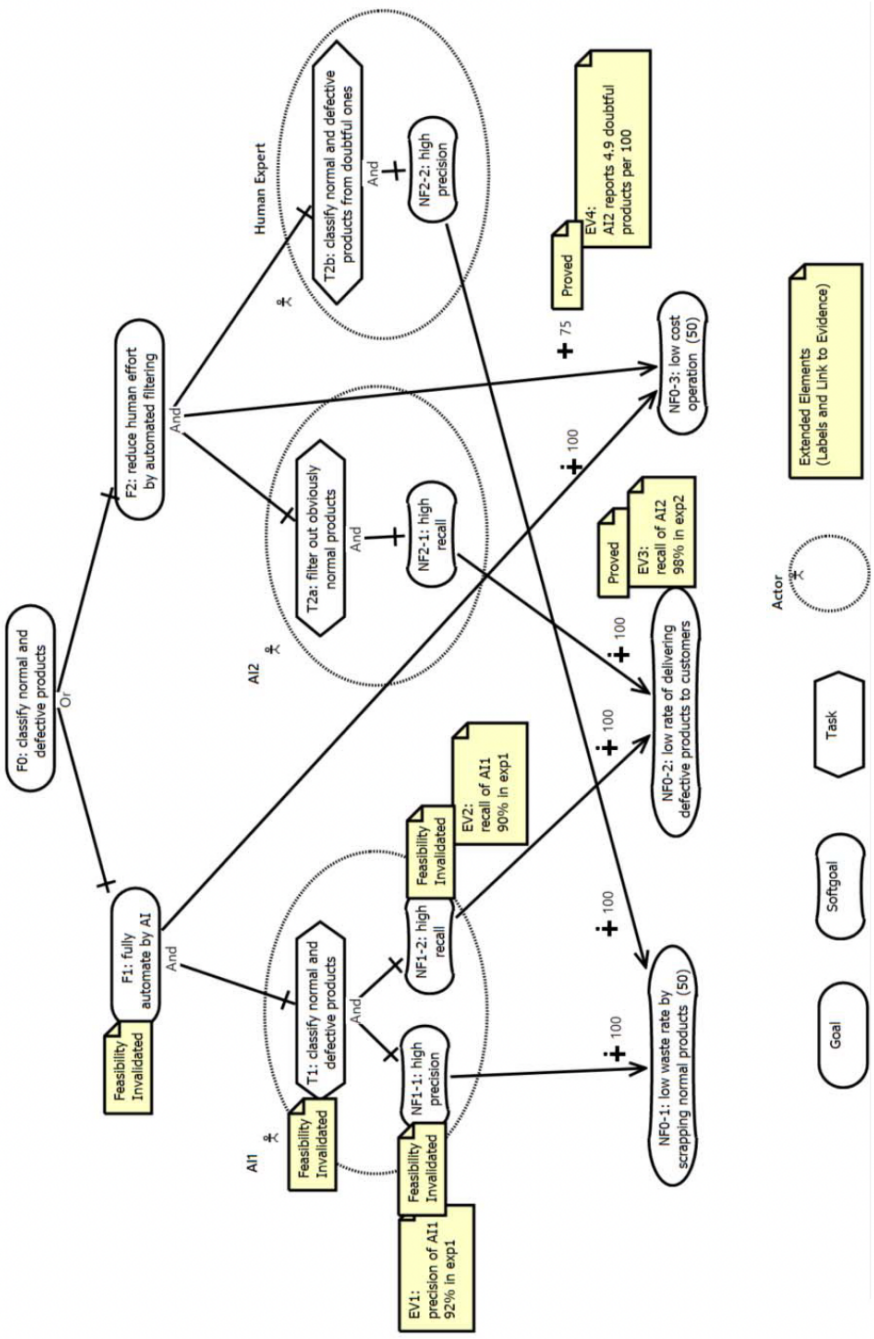
A.2 Vragen uit de GORE I* methode volgens Barrera e.a. 2024

Figuur 9: GORE I* (Barrera e.a. 2024)

Table 3
Questionnaire tool for ML experts to aid in ML requirements capture.

Question	Metamodel Class/es affected
Q1. From a qualitative point of view, which objectives do I have? Is a regression/clustering/classification model required?	*Goal* *MLGoal(ClassificationGoal, RegressionGoal, ClusteringGoal)*
Q2. Which metric is the proper one to measure the planned goal? Which value of that metric would consider the project as success?	*Indicators* (*ClassificationIndicator*, *RegressionIndicator*, *ClusteringIndicator*)
Q3. Is the explainability of the model or the effect of every dimension on each prediction mandatory?	*MLQuality* (*Explainability/Transparency*)
Q4. Can the behaviour of data be changed? New instances can be added, or it is static?	*MLQuality* (*ConceptDriftAdaptation*)
Q5. Is data biased? Is it necessary any additional preprocessing step with the aim of providing equity in data?	*MLQuality* (*Equity*)
Q6. Is there any other specific ML quality aspect that the ML model should have?	*MLQuality*
Q7. Should the system provide a prediction within a time limit?	*Dataset*
Q8. Which data should be considered for the model? Must an agnostic approach be followed to deal with the problem, or is domain knowledge available?	*Dataset* *Source*
Q9. Which granularity of data is suitable for achieving the goal? Is the required granularity available?	*Dataset* (*Parameters temporalResolution*) *and refreshPeriod*)
Q10. Is external validation required, or is it enough with internal validation?	*Source* *Dataset*

A.3 GORE-MLOps Goal Graph volgens Ishikawa en Matsuno 2020



Figuur 10: GORE-MLOps Goal Graph (Ishikawa en Matsuno 2020)