KU LEUVEN

FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN

TECHNOLOGIECAMPUS GENT

Deep learning modellering voor de kwantitatieve FTIR-analyse van ternaire stabilisatorenmengsels

Laurens VAN DEN MEERSCHE

Promotor: Prof. Dr. E. Courtijn

Co-promotoren: *Ing. S. Lambert*

Ing. D. Verstraete

Ing. M. Saelens

Masterproef ingediend tot het behalen van de graad van master of Science in de industriële wetenschappen: *Chemie (Chemie)*

Academiejaar 2020-2021

Voorwoord

In deze uitdagende masterproef kon ik naast mijn passie voor chemie ook mijn grote interesse voor technologie kwijt. Ik kan dus zeggen dat deze thesis een mooi sluitstuk vormt van mijn opleiding als industrieel ingenieur chemie. Door het volgen van een multidisciplinaire richting zoals industriële wetenschappen, kon ik verschillende ingenieursaspecten toepassen en ontdekken. Dit is inzake dit eindwerk niet anders geweest.

Ik was tot voor de start van deze masterproef nog nooit in aanraking gekomen met deep learning, Python of enzympreparaten. Het schrijven van algoritmen die hierop betrekking hebben, behoorden dan ook (nog) niet toe tot mijn vaardigheden. Hoewel dit voor mij niet meteen voor de hand lag, raakte ik al snel geboeid door deze materie. Dit onderwerp combineert de wereld van de informatica met de wereld van de chemie. Deze intrigerende combinatie maakte dat het afronden van dit onderzoek nauwelijks heeft aangevoeld als een zware of lastige opdracht. Het tegendeel is namelijk waar. Het voorbije semester heeft niet alleen mijn blik verruimd, het was eveneens een leerrijke en plezierige tijd. Dit komt vooral door de fijne omgeving en professionele ondersteuning die ik vond bij Christeyns. Een eerste en vooral positieve ervaring in een chemisch bedrijf is alvast mooi meegenomen.

In de eerste plaats gaat een speciaal woord van dank uit naar mijn promotor, Prof. Dr. Eddy Courtijn. Dit voor de begeleiding, bijsturing en kritische vragen die de wetenschappelijkheid van deze thesis vergroot hebben. Daarnaast mogen Ing. Simon Lambert, Ing. Delphine Verstraete en Ing. Marjolein Saelens absoluut niet vergeten worden. Naast een bijna dagelijkse opvolging, hebben zij meegeholpen om deze thesis de vorm te geven zoals deze vandaag is. Ook Brian Demuynck hoort in dit rijtje thuis. Ik bedank hem graag voor de aangename gesprekken tussendoor en voor de suggesties omtrent mogelijke experimenten. De feedback en hulp die ik gekregen heb van al deze personen, hebben dit eindwerk naar een hoger niveau getild. Een welgemeende dank jullie wel is hier dan ook op zijn plaats.

Graag wil ik ook mijn ouders en broer bedanken voor hun onvoorwaardelijke steun in deze laatste periode van mijn studententijd. Door jullie warme omgeving heb ik dit eindwerk kunnen maken tot wat het nu is. Daarnaast is er nog een persoon die niet mag vergeten worden, namelijk mijn vriendin Emilie Van den Brande. Ik bedank haar met veel plezier om de motiverende, enthousiasmerende en hartverwarmende kracht te zijn in mijn laatste jaar als student.

Hopelijk bent u, net zoals ik, gebeten en geïntrigeerd door dit onderwerp. Ik sluit dit voorwoord graag af met u veel leesplezier toe te wensen.

Laurens Van den Meersche,

Vollezele, 11 juni 2021

Samenvatting

Wasmiddelen bevatten historisch gezien een grote hoeveelheid surfactants. Door de negatieve invloed op het milieu worden deze meer en meer vervangen door enzympreparaten. Deze zijn een ecologisch alternatief voor surfactants en bieden een economisch voordeel. Daarom is Christeyns geïnteresseerd in de kwantificatie van deze enzympreparaten. Daartoe wordt in deze thesis gebruik gemaakt van deep learning neurale netwerken. Ter analyse van de stalen wordt de snelle techniek van verzwakte totale reflectie – Fourrier transformatie infraroodspectroscopie (ATR-FTIR) gehanteerd. In de enzympreparaten worden de enzymen stabiel gehouden door stabilisatoren. De stabilisatoren waarop in dit onderzoek gefocust worden zijn sorbitol, monopropyleenglycol (MPG) en glycerol.

Deep learning werkt door middel van de opbouw van een neuraal netwerk. Door training wordt het netwerk sterker en door validatie kan het opgebouwde model beoordeeld worden. Daartoe is een grote hoeveelheid aan data nodig. Deze data wordt onderverdeeld in trainings-, test- en validatiedata waarbij de validatiedata toelaat de performantie van een model na te gaan. Dit gebeurt in deze thesis steeds via de gemiddelde absolute fout (MAE) en de absolute fout (AE) tussen de voorspelde concentratie en de werkelijke concentratie.

Om deze grote hoeveelheid aan data te verwerken, wordt in de literatuur vaak verwezen naar pre-processingsmethoden. In dit onderzoek worden principale component analyse (PCA) en standard normal variate (SNV) onderzocht. Uit de praktijk blijkt daarnaast dat bij de opname van een infraroodspectrum, een basislijnverschuiving veelvuldig voorkomt. Daarom wordt in deze thesis een algoritme geschreven dat automatisch een basislijncorrectie doorvoert. Er wordt gebruik gemaakt van een adaptive iteratively reweighted penalized least squares algoritme (AirPLS) en van een asymmetric least squares algoritme (AsLS). Daarnaast wordt ook een manuele basislijncorrectie en geen basislijncorrectie in de vergelijking mee opgenomen ter referentie.

Aangezien de verwerking van dergelijke data complex is, worden eerst binaire mengsels van stabilisatoren gemaakt en pas vervolgens ternaire mengsels. Tot slot zullen commercieel verkrijgbare enzympreparaten benaderd worden door de hoeveelheid water in de mengsels op ongeveer 50 % te brengen. Ten einde de validatie worden commercieel verkrijgbare enzympreparaten gebruikt om de modellen te beoordelen.

Uit het onderzoek volgt dat het mogelijk is om een kwantificatie van ternaire stabilisatorenmengsel uit te voeren met behulp van FTIR en deep learning. Het ’AirPLS-Geen’-model bleek hier het meest geschikt voor te zijn met een MAE-waarde van 1,77 ± 0,43. Ditzelfde model bleek ook het meest intralaboratorium-reproduceerbaar en het meest herhaalbaar te zijn. Ook werd gevonden dat een basislijncorrectie in combinatie met geen pre-processing, met name het ‘AsLS-Geen’-model, veelbelovend is in de analyse van enzympreparaten. Indien de hoeveelheid componenten verder vergroot wordt, zal de nauwkeurigheid van de inschattingen nog verder toenemen.

Trefwoorden: Deep learning, Python, ATR-FTIR, Chemometrie, Enzympreparaten

Abstract

Laundry detergents historically consisted of a large amount of surfactants. Due to the negative effects on the environment, these are systematically replaced with enzyme preparations. These preparations are an ecological and an economical high-quality alternative to surfactants. For this reason, Christeyns is interested in the quantification of these enzyme preparations. In this thesis the quantification is carried out by a deep learning neural network which processes infrared spectra. The spectra are obtained by the fast technique of attenuated total reflectance - Fourrier transformed infrared spectroscopy (ATR-FTIR). The enzyme preparations are kept stable by enzyme stabilizers. In this thesis the focus lies solely on sorbitol, glycol and propylene glycol (MPG).

Deep learning is a subdivision of artificial intelligence and works through neural networks. Python-code is used to built the models. The models are trained, tested and validated with subsets of data. The performance of each model is evaluated with the results of the validation. This is done based on the mean absolute error (MAE) and the absolute error (AE) between the estimation of the concentration and the actual concentration.

There is a necessity of large amounts of data to train the neural network. In literature many references point to the advantages of pre-processing methods. Therefor in this thesis principal components analysis (PCA) and standard normal variate (SNV) are investigated. In order to compare these methods, a reference is made to a state of no pre-processing. As the input of the models is an infrared spectrum, a baseline shift is part of the recording. There are reasons to believe that this has a negative impact on the quantification. That is why in this research, automatically correcting algorithms such as adaptive iteratively reweighted penalized least squares algorithm (AirPLS) and asymmetric least squares algorithm (AsLS) are written in Python. Also, a manual correction and no correction at all, are used as a reference to examine the performance of the algorithms.

Because of the complexity of these datasets, there is a necessity of working in phases. A first experimental phase consists of binary mixtures of stabilizers. The second phase is carried out with ternary mixtures. In a following experiment water is added to obtain a concentration of approximately 50 % in the binary and ternary mixtures of stabilizers to simulate the matrix of enzyme preparations. At last, commercially available enzyme preparations are used to validate the models.

All of these experiments showed that a quantification of ternary mixtures of enzyme stabilizers can be carried out using FTIR and deep learning. The most performant model was found to be the ‘AirPLS-Geen’-model with a MAE-score of 1,77 ± 0,43. The same conclusion was drawn based on the experiment of the intralaboratory reproducibility and the repeatability. It was proven that a manual baseline correction is less suitable to perform a quantification. In general, these experiments showed the strength of a baseline correction especially in combination with no pre-processing. The ‘AsLS-Geen’-model has a lot of potential to analyze enzyme preparations. The accuracy of the predictions will enhance further if the set of components in the training is increased.

Keywords: Deep learning, Python, ATR-FTIR, Chemometrics, Enzyme preparations

INHOUD

Symbolenlijst

Kleine letters

Symbool	Betekenis	Eenheid
b	Bias	[-]
c	Concentratie	[mol/l]
$j(\theta)$	Verliesfunctie van het perceptron	[-]
k_{cat}	Katalytische snelheidsconstante	[s^{-1}]
l	Weglengte	[cm]
m	Aantal kolommen in de elementenmatrix	[-]
$\max(x_j)$	Kolommaximum	[-]
$\min(x_j)$	Kolomminimum	[-]
n	Aantal rijen in de elementenmatrix	[-]
s_{blanco}	Standaarddeviatie van de blanco-stalen	[%]
s_j	Standaarddeviatie van de kolom	[-]
w_i	Gewicht geassocieerd met de inputwaarde	[-]
x_i	Inputwaarde	[-]
x_{ij}	Oorspronkelijke matrixelement	[-]
x'_{ij}	Gecentreerde matrixelement	[-]
x''_{ij}	Gestandaardiseerde matrixelement	[-]
x'''_{ij}	Genormaliseerde matrixelement	[-]
$\overline{x_j}$	Kolomgemiddelde	[-]
y_i	Werkelijke outputwaarde	[-]
y_{pred} / $y_{pred,i}$	Voorspelde outputwaarde	[-]

Hoofdletters

Symbool	Betekenis	Eenheid
A	Absorbantie	[-]
E	Energie van een foton	[J]
$I_{BL}(\lambda)$	Intensiteit van de basislijn in functie van de golflengte	[-]
$I_{Ruw}(\lambda)$	Intensiteit van het oorspronkelijk signaal in functie van de golflengte	[-]
$I_{SNV}(\lambda)$	Intensiteit van het signaal in een IR-spectrum in functie van de golflengte na SNV-correctie	[-]
K_M	Michaelis-Menton constante	[mol/l]
$\mathcal{L}$	Verlies-operator	[-]

LOQ	Kwantificatielimiet	[%]
N	Aantal inputwaarden	[-]
T	Transmissie	[-]
X	Data in de vorm van een elementenmatrix	[-]

Griekse kleine letters

Symbool	**Betekenis**	**Eenheid**
γ	Ruimtelijke deformatie (uit het vlak)	[-]
δ	Vlakke deformatie (schaarbeweging in het vlak)	[-]
ε	Molaire absorptiecoëfficiënt	[l/(cm*mol)]
η	Leersnelheid	[-]
θ	Gewicht	[-]
λ	Golflengte	[cm]
μ	Gemiddelde van de netto-intensiteiten	[-]
$\bar{\nu}$	Golfgetal	[cm^{-1}]
ν_a	Asymmetrische strekkingsvibratie	[-]
ν_s	Symmetrische strekkingsvibratie	[-]
ρ	Rocking vibratie (schommelbeweging)	[-]
σ	Standaarddeviatie van de netto-intensiteiten	[-]
τ	Twisting vibratie (draaibeweging)	[-]
ω	Wagging vibratie (kwispelbeweging)	[-]

Lijst met afkortingen

Afkorting	Betekenis	
	Engels	**Nederlands**
AE	Absolute error	Absolute fout
AI	Artificial intelligence	Artificiële intelligentie
AirPLS	Adaptive iteratively reweighted penalized least squares	***
ANNs	Artificial neural networks	Artificiële neurale netwerken
AsLS	Asymmetric least squares	***
ATR-FTIR	Attenuated total reflectance - Fourrier transformed infrared spectroscopy	Verzwakte totale reflectie – Fourrier transformatie infraroodspectroscopie
AUC	Analytical ultracentrifuge	Analytische ultracentrifuge
CNN	Convolutional neural network	Convolutioneel neuraal netwerk
DL	Deep learning	***
DoE	Design of Experiments	***
EP	Enzyme preparations	Enzympreparaat
erPLS	Extended range penalized least squares	***
GaCspline	Genetic algorithm Cubic spline baseline correction	***
HPLC	High pressure liquid chromatography	Hoge druk vloeistof-chromatografie
IAsLS	Improved asymmetric least squares	***
IL-reproduceerbaarheid	Intralaboratory reproducibility	Intralaboratorium-reproduceerbaarheid
ITC	Isothermal titration calorimetry	Isotherme titratie calorimetrie
JAsLS	Jiang’s asymmetric least squares	***
LC-MS	Liquid chromatography – mass spectrometry	Vloeistofchromatografie - massaspectrometrie
LOD	Limit of detection	Detectielimiet
LOQ	Limit of quantification	kwantificatielimiet
LSTM	Long short term memory	***

Afkorting	Betekenis	
	Engels	**Nederlands**
MAE	Mean absolute error	Gemiddelde absolute fout
ML	Machine learning	***
MPG	Propylene glycol	Monopropyleenglycol
PC	Principal component	Principale component
PCA	Principal component analysis	Principale component analyse
PCR	Principal component regression	Principale component regressie
PLS	Partial least squares	***
RMS	Root Mean Squared	Kwadratisch gemiddelde
RNN	Recurrent neural networks	Recurrent neuraal netwerk
RO	Reversed osmosis	Omgekeerde osmose
RP-HPLC-MS	Reversed phase – high pressure liquid chromatography – mass spectrometry	Omgekeerde fase – hoge druk vloeistofchromatografie – massaspectrometrie
ReLU	Rectified linear activation function	***
SDS-PAGE	Sodium dodecyl sulfate polyacrylamide gel electrophoresis	natriumdodecylsulfaat-polyacrylamide-gelelektroforese
SEC	Size-exclusion chromatography	Size-exlcusion chromatografie
SG	Savitzky-Golay	Savitzky-Golay
SGD	Stochastic gradient descent	Stochastische gradiënt daling
SNNs	Simulated neural networks	Gesimuleerde neurale netwerken
SNV	Standard normal variate	***

*Opmerking: Er staat *** wanneer er geen Nederlandse vertaling gangbaar is.*

Lijst met figuren

Hoofdstuk 2: Literatuurstudie

Hoofdstuk 3: Materialen en methode

Hoofdstuk 4: Resultaten en discussie

Lijst met tabellen

Hoofdstuk 2: Literatuurstudie

Hoofdstuk 4: Resultaten en discussie

1 INLEIDING

1.1 Christeyns

Christeyns is een familiebedrijf dat als zeepfabriek opgericht werd in 1946 door Jules Robert Christeyns. De vestiging bevindt zich nog steeds in Gent. Destijds was de professionele reinigingsindustrie, de wasserijmarkt alsook de industriële oleochemie het doelpubliek van deze zeep- en detergentenproducent. Nadat het bedrijf 43 jaar bestond, nam de familie Bostoen het management over. Tot op de dag van vandaag is Alain Bostoen de CEO van Christeyns. In 1995 is Christeyns begonnen aan een Europese expansie, wat de start betekende van een reeks buitenlandse activiteiten. Nog eens drie jaar later werd het gamma uitgebreid na een overname van Oils&Soaps Ltd. Dit bedrijf was gespecialiseerd in betonadditieven waardoor een nieuwe markt zich aanbood. De overname van JohnsonDiversey is eveneens een belangrijk moment in de geschiedenis van het bedrijf. Door deze overname verstevigde Christeyns de positie als toonaangevende Europese speler. In 2010 en 2012 werd het gamma opnieuw vergroot met de opstart van respectievelijk een voedselhygiëne afdeling en een medische afdeling. (1)

Heden ten dage bevindt de hoofdvestiging zich nog steeds in Gent en zijn er dochterbedrijven in 30 verschillende landen. Het werknemersbestand is door de vele overnamen reeds groter geworden dan 1000 werknemers. Christeyns spitst zich toe op de professionele textielverzorging, voedingshygiëne, professionele reiniging en de medische sector. De hoofdactiviteit is nog steeds de productie en optimalisatie van detergenten. Daarnaast zijn er nevenactiviteiten zoals Oscrete (bouwchemie) en Govi (proceschemie). Oscrete levert producten zoals weekmakende toevoegsels en waterdichtingsproducten om de kwaliteit van beton en dergelijke te verbeteren. De dochteronderneming Govi produceert voornamelijk dispersies en emulsies voor de textielnijverheid, voedings- en drankensector. (1)

De professionele textielverzorging is een belangrijke tak van dit bedrijf. Er wordt getracht efficiënte detergenten en desinfecterende middelen te produceren. Alsook wordt apparatuur voor industriële wasserijen aangeboden. In het bijzonder zijn innovatie en vergroening, factoren waarmee rekening wordt gehouden. Daartoe behoren bijvoorbeeld mogelijke water- en energiebesparingen bij het wassen. (1)

1.2 Doelstelling

Bij de wassing van textiel worden meer en meer vloeibare enzympreparaten gebruikt ter vervanging van de grote hoeveelheid surfactants. Deze enzympreparaten bevatten verschillende stabilisatoren waaronder glycerol, sorbitol en monopropyleenglycol (MPG). Een aanzienlijk deel van de vloeibare matrix van de enzympreparaten bestaat uit polyolen. Het doel van deze thesis is om een deep learning model op te stellen en aan de hand van het FTIR-spectrum een ternair mengsel van stabilisatoren (MPG, glycerol, sorbitol) te kwantificeren en bij uitbreiding de aanwezigheid aan te tonen van glycolen en andere stabilisatoren. Deze opdracht kadert in een confidentieel project binnen Christeyns.

Typisch bij een opname van infraroodspectrum is een basislijnverschuiving. Deze verschuivingen worden veelal manueel gecorrigeerd. Dit is niet alleen tijdsintensief maar ook moeilijk reproduceerbaar. Het is de bedoeling dat er een algoritme wordt ontwikkeld die deze basislijncorrectie zelfstandig doorvoert. Ook de problemen van intralaboratorium-reproduceerbaarheid en herhaalbaarheid inzake FTIR-opname en kwantificatie worden onderzocht. Immers wanneer een analyse op een andere dag of door een andere analist wordt uitgevoerd, worden andere resultaten verkregen. Daartoe worden verschillende soorten algoritmes opgesteld en vergeleken met elkaar opdat deze problemen worden verholpen. Er wordt dus gekeken naar welke basislijncorrecties mogelijk zijn en bijkomend of deze een gunstig effect hebben op de kwantificatie. Er zal ook worden onderzocht welke pre-processingstechnieken geschikt zijn om de kwantificatie uit te voeren met het deep learning model.

Voor de kwantificatie is het niet aangewezen om slechts met één absorptiepiek uit het infraroodspectrum te werken. Het deep learning model houdt rekening met het gehele spectrum waardoor de nauwkeurigheid vergroot. Er zal worden nagegaan of werken met hoog gecorreleerde golfgetallen een invloed heeft op de nauwkeurigheid van het model.

Door de complexiteit van het probleem, zal eerst een model opgebouwd worden voor binaire mengsels van de stabilisatoren en zal nadien de koppeling gemaakt worden ter opbouw van een deep learning model voor een ternair mengsel. Deze binaire en ternaire stabilisatorenmengsels zullen bij uitbreiding aangevuld worden met water, waarvoor eveneens een deep learning model wordt opgebouwd. Dit model wordt getoetst met commercieel verkrijgbare enzympreparaten.

2 LITERATUURSTUDIE

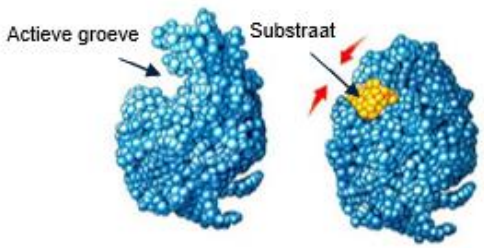
De laatste jaren is de vraag naar enzymen in detergenten toegenomen. Wanneer wasprocessen in het algemeen bekeken worden, zijn enzymen over de jaren heen een prominente rol beginnen te spelen. Deze zijn een ecologisch en economisch hoogwaardig alternatief voor de grote hoeveelheid tensio-actieve stoffen. Deze zijn door het gebruik van enzymen nu in een kleinere hoeveelheid aanwezig zijn in wasmiddelen. Door de toenemende aandacht voor het milieu, zal de belangstelling de volgende jaren nog meer stijgen. (2,3)

Hoewel er veel voordelen verbonden zijn aan het gebruik van enzymen, is vooral de kost een nadeel. Daarom is het voor Christeyns van belang om de stabiliteit van de enzymen na te gaan en op te volgen in continue systemen alsook in hun producten. Het zou dan ook interessant zijn om de matrix van de enzymenpreparaten te kunnen kwantificeren met behulp van een snelle techniek zoals FTIR. De verwerking van de daarbij horende spectrale data is echter niet eenvoudig. Dit kan wel gedaan worden met behulp van een neuraal netwerk en daaraan gekoppelde deep learning.

In deze literatuurstudie wordt eerst gekeken naar wat enzymen zijn en hoe deze geproduceerd worden. Mogelijke stabilisatoren alsook kwaliteit- en activiteitscontroles worden belicht. Daarnaast wordt de techniek van FTIR onder de loep genomen en wordt gekeken naar mogelijke manieren om ermee te kwantificeren. Daartoe zal de aandacht gevestigd worden op mogelijke basislijncorrecties. De spectrale data zal onderworpen worden aan voorbehandelingen, daarom wordt ook ingegaan op verschillende soorten van pre-processingstechnieken. Ten slotte zal er ook gekeken worden naar wat deep learning is en hoe een neuraal netwerk wordt opgebouwd.

2.1 Enzymen

Enzymen zijn proteïnen die zich in een quaternaire opvouwingsstructuur bevinden. Hierdoor zijn deze in staat een reactie te katalyseren. Deze biokatalyse is zowel stereo- als regioselectief. Dit komt door de functionele groepen van de aminozuren binnenin de actieve groeve van het enzym. De reactie wordt versneld door een verlaging van de activeringsenergie van de reactie. Theoretisch gezien kan de werking en selectiviteit van het enzym verklaard worden door het zogenoemde induced-fit model of door het lock-and-key model. Op Figuur 2.1 is een enzym te zien waarop de actieve groeve is aangeduid. Dit is de plaats waar het enzym de katalytische werking uitoefent. (4)



Figuur 2.1: Een enzym in quaternaire toestand met aanduiding van de actieve groeve (5)

Ook geeft Figuur 2.1 de quaternaire opvouwing weer. Bij denaturatie (door temperatuur, agitatie, pH-veranderingen, …) treedt een degradatie op van deze quaternaire structuur en verliest het enzym zijn werking. Een ver doorgedreven procescontrole is dus noodzakelijk opdat het enzym de reactie kan blijven katalyseren. (4)

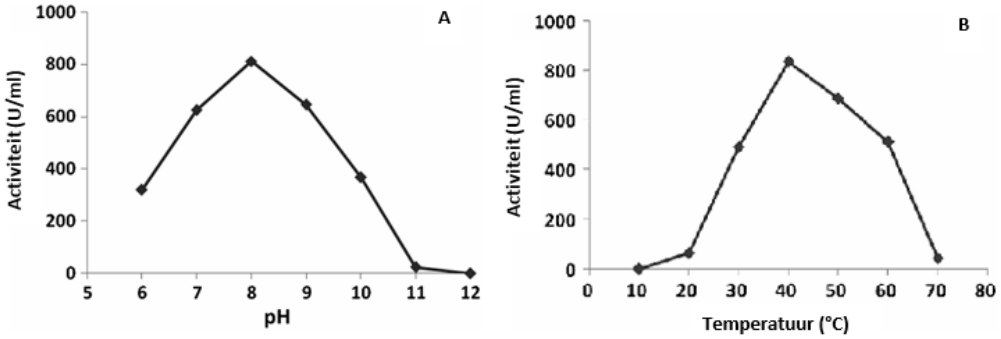
2.1.1 Enzymen in het wasproces

Enzymen die interessant zijn in wasprocessen zijn onder andere lipasen, proteasen, amylasen, cellulasen en mannanasen. Lipasen katalyseren de vetafbraak (bijvoorbeeld olievlekken) terwijl proteasen zullen instaan voor de afbraak van proteïnen (bijvoorbeeld bloedvlekken). Tabel 2.1 geeft een overzicht van deze enzymen en de functie die deze hebben in wasprocessen. (6)

Tabel 2.1: Overzicht van enzymen die gebruikt worden in wasprocessen (6–8)

Enzym	Functie	Toepassing
Lipase	Afbraak van vetten	Olievlekken verwijderen
Protease	Afbraak van proteïnen	Bloedvlekken verwijderen
Amylase	Afbraak van amylose (niet-vertakt zetmeel)	Vlekken van barbecuesaus verwijderen
Cellulase	Afbraak cellulose; verzachten en beschermen van kledingvezels	Grasvlekken verwijderen
Mannanase	Afbraak van mannose; gommen	Vast vuil verwijderen

Proteasen hebben een pH-optimum gelegen tussen 6 en 12 en een werkgebied qua temperatuur tussen 30 °C en 70 °C afhankelijk van de producerende species. Proteasen zijn kwetsbare enzymen in de zin dat ze gemakkelijk aangetast worden door een pH- en temperatuursverschuiving. In wasmiddelen is het aanwezige bleekmiddel een agressief bestanddeel voor proteasen. De enzymen zullen in grote mate dienen gestabiliseerd te worden. Figuur 2.2.a geeft een pH-optimum weer voor een protease en het b-gedeelte van Figuur 2.2, stelt een temperatuursoptimum voor. (6–8)



Figuur 2.2: Optimale omstandigheden voor proteasen (*Bacillus aquimaris*): a) pH-optimum: 8 en b) Temperatuursoptimum: 40 °C (9)

Amylasen (in het bijzonder α-amylase) hebben een vergelijkbaar pH-optimum zoals de proteasen en kunnen temperaturen verdragen tot 100 °C. Deze enzymen hebben overigens een synergetisch effect met de optische witmakers die in wasmiddelen aanwezig zijn. Om de stabiliteit van amylasen te vergroten, kan calcium (in zoutvorm) toegevoegd worden. Ook deze enzymen zijn vatbaar voor aantasting door bleekmiddel. (6–8)

Een ander belangrijk type enzym dat zich in wasmiddelen bevindt, zijn de lipasen. Dit enzym is historisch gezien moeilijker te produceren dan andere enzymen. Door *protein engineering* is de opbrengst sterk verhoogd doorheen de jaren. Ook hier is het werkgebied van de pH vergelijkbaar met dat van de amylasen en de proteasen. Het temperatuursbereik kan tot onder 20 °C gaan. Ook hier kan calcium toegevoegd worden om de stabiliteit en activiteit te verhogen. (6–8)

Er zijn ook nog andere enzymen aanwezig in wasmiddelen, waaronder cellulasen en mannanasen. Cellulase wordt vooral toegevoegd omdat het de vuilverwijdering vergemakkelijkt en de kledingsvezels beschermt. Dit enzym nestelt zich tussenin de textielvezels en geeft een tijdelijke sterkte aan de kledij, wat gunstig is voor de verwijdering van het vuil en het tegengaan van beschadigingen. Verven van organische oorsprong kunnen door dit enzym afgebroken worden. Mannanasen verwijderen bijvoorbeeld vlekken van barbecuesaus, ketchup of chocolade. Dit enzym breekt mannose en gommen af. Gommen zijn polysachariden waarbij mannose een samenstellend suiker is, vaak in combinatie met glucose. De gom werkt als een soort verdikker en kan ervoor zorgen dat er vlekken op kledij worden gevormd door vast vuil. (6–8)

Het gebruik van enzymen is een biologisch verantwoorde manier om textiel te wassen. Zo zal er bijvoorbeeld minder bleekmiddel nodig zijn wanneer enzymen worden gebruikt. Dit zorgt voor een lagere organische belasting van het afvalwater. Vereisten waaraan de gebruikte enzymen moeten voldoen, zijn onder andere: onschadelijkheid voor de vezels, doeltreffendheid en het verbeteren van de kleur en witheid van het textiel. Alsook de bescherming van de gebruiker van het textiel is belangrijk, bijvoorbeeld naar allergieën toe. (10,11)

2.1.2 Productie van enzymen

Om enzymen op industriële schaal te produceren, wordt veelal gebruik gemaakt van een geroerde tankreactor. Het roeren is noodzakelijk met als voornaamste reden dat de nutriënten verdeeld worden over de volledige inhoud van de tank. Aerobe celculturen worden het vaakst gebruikt en zijn volledig ondergedompeld in het medium dat zich in de reactor bevindt.

Het gemakkelijkste proces vindt plaats wanneer het enzym extracellulair wordt geproduceerd. Dit is ook de meest gangbare methode in de industriële productie. Op die manier kan een relatief eenvoudige afscheiding van micro-organisme en enzym plaatsvinden. In 2014 zou er op dergelijke wijze naar schatting 500.000 kg aan proteasen geproduceerd zijn. Tabel 2.2 geeft een overzicht van welke micro-organismen verantwoordelijk zijn voor de productie van de belangrijkste enzymen die gebruikt worden in het wasproces. Zo zal het thermostabiele α-amylase geproduceerd worden door *Bacillus licheniformis*. (6,11)

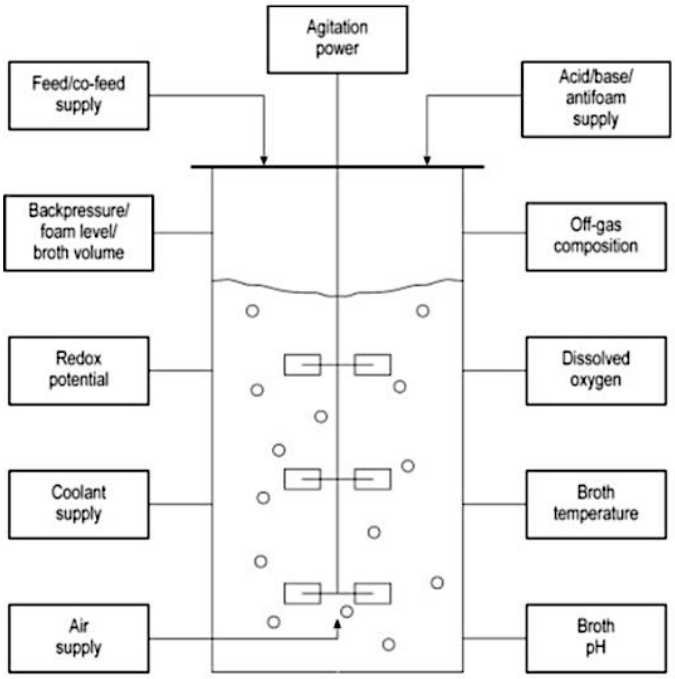
Tabel 2.2: Overzicht van enzym-producerende micro-organismen (11)

Enzym	Micro-organisme
Protease	*Aspergillus oryzae, Bacillus aquimaris*
Lipase	*Aspergillus oryzae, A. flavus*
Amylase	*Aspergillus* sp., *Bacillus licheniformis*
Cellulase	*Aspergillus niger, Bacillus sp.*
Mannanase	*Bacillus sp.*

In het productieproces zijn er twee aparte voorbereidingsstappen. Een eerste stap bestaat erin de celcultuur klaar te maken voor productie, een andere stap gaat over het optimaliseren van het medium en steriliseren van de apparatuur. Eens deze stappen voltrokken zijn, wordt de celcultuur in de reactor toegevoegd en kan de fermentatie beginnen. Nadien worden enzym, micro-organisme en medium van elkaar gescheiden. Er wordt gepoogd om zo veel mogelijk recyclage van het medium te waarborgen alsook van de micro-organismen. (6,11)

Belangrijk bij de productie van enzymen is dat er rekening gehouden wordt met de specifieke condities die de micro-organismen vereisen. De metabolismen zijn verschillend per celcultuur, wat ervoor zorgt dat de productie strikte temperatuurs- en pH-grenzen vereist. Ook tijd is een belangrijke factor naar conversie toe. Opnieuw is dit iets wat afhangt van het gebruikte micro-organisme en zijn er dus andere individuele optimale omstandigheden. Problemen met opgeloste zuurstof en het ontstaan van dode zones, zijn limitaties voor de grootte van de fermentor. Afhankelijk van het micro-organisme kan een fermentor 20 m^3 tot 200 m^3 groot zijn. (6,12)

Zoals reeds aangehaald, moet het proces gecontroleerd verlopen. Zo zullen onder andere de pH, zuurstofgehalte, temperatuur, potentiaal, agitatie en voedingsconcentratie gemeten worden over de reactor heen. Figuur 2.3 geeft dergelijke controlepunten weer. (6)

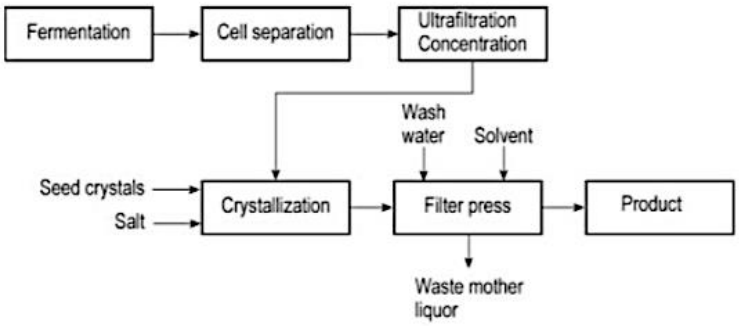


Figuur 2.3: Een fermentor met aanduiding van de controleparameters (6)

Na de productie wordt er aan isolatie van de geproduceerde enzymen gedaan. De isolatie start meestal met een filtratiestap om de grovere, vaste stoffen te scheiden van de vloeistoffase. Vacuümfiltratie en het gebruik van een filterpers, behoren tot de mogelijkheden om de filtratie te voltrekken. Ook centrifuge is mogelijk om de scheiding te verwezenlijken. (6)

Na de filtratie of centrifuge gebeurt een aanrijking van het product door verwijdering van het resterende medium. Thermische opconcentratie wordt het meeste gebruikt. De temperatuur moet goed in de gaten gehouden worden daar er geen denaturatie van de enzymen mag optreden. Eventueel kan de thermische methode vervangen worden door een precipitatie of membraanfiltratie indien het enzym te temperatuurgevoelig is voor een thermische opconcentratie. (6)

Dit geconcentreerde product bevat nog relatief veel onzuiverheden waaronder hardnekkige sporen van het medium. Dit kan naargelang de toepassing onderworpen worden aan een zuiveringsstap. Kristallisatie gevolgd door een filterpers met waswater wordt het meest ingezet om de zuivering te voltrekken. Figuur 2.4 geeft een processchema van dergelijke kristallisatie weer. (6)



Figuur 2.4: Processchema van een industriële kristallisatie ter zuivering van enzymen (6)

Figuur 2.4 toont dat na ultrafiltratie, het product overgaat naar een kristallisator. Hier worden kiemen (seed crystals genoemd op Figuur 2.4) toegevoegd alsook zout (bijvoorbeeld $CaSO_4$). Na de kristallisatie gaat het product met de kristallen naar een filterpers waar het kristalmengsel gewassen wordt met water, eventueel aangevuld met een solvent. Op die manier worden enerzijds zouten verwijderd met het water en anderzijds worden ongewenste organische vervuilingen met het solvent meegevoerd. De vloeistof die overblijft, kan eventueel gerecycleerd worden om opnieuw dienst te doen als wasvloeistof. Uiteindelijk wordt het product verkregen en opgeslagen. (6)

Hedendaagse inzichten vanuit de *protein engineering* en genetische modificatie, geven een optimistisch toekomstperspectief voor de industriële productie van enzymen. Genetische modificatie kan er bijvoorbeeld voor zorgen dat de celculturen meer en sneller enzymen kunnen produceren. Deze enzymen kunnen bijvoorbeeld zo gemodificeerd worden dat ze in een andere temperatuurs- en pH-bereik werkzaam zijn. (6,12)

Zoals aangegeven worden enzymen vaak extracellulair geproduceerd. Ook is het mogelijk om intracellulaire enzymen te produceren. In dit geval zal er een bijkomende stap zijn waarbij de micro-organismen worden gedisrupteerd. Dit kan bijvoorbeeld op fysische wijze door een drukverhoging of door ultrasoon geluid. Ook mechanisch disrupteren is mogelijk door bijvoorbeeld een kogelmolen. Als scheidingsstap voor het vast-vloeistof mengsel, kan hier aan extractie worden gedaan. (6)

2.1.3 Stabiliseren van enzymen en de rol van glycolen

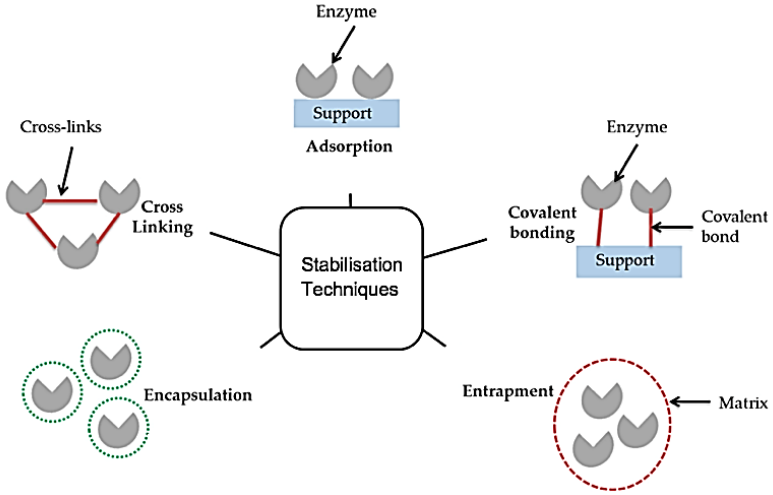
Het is van belang dat enzymen stabiel gehouden worden zodat ze de beoogde reacties kunnen katalyseren. Daarvoor worden stabilisatoren gebruikt. Deze zijn nodig om de enzymen te beschermen tegen denaturatie als gevolg van de inwerking van zuren, basen, zouten en agitatie. Omtrent de theoretische achtergrond van een verhoogde stabilisatie, zouden twee verschillende verklaringen zijn. (13,14)

Een eerste manier van benaderen kijkt naar de deactivatie van de proteïnen. De deactivatie door denaturatie wordt sterk benadeeld door extra hinder die de beschermende (viskeuze) laag met zich meebrengt. De ontvouwing van proteïnen en bij uitbreiding van het enzym, worden in energie verhoogd. Door deze energieverhoging zal de ontvouwing met meer moeite

plaatsvinden. Zo wordt het enzym beschermd tegen oxidaties, deamidaties, inwerking van zouten en agitatie. (13,14)

Een andere mogelijke verklaring zou kunnen zijn dat er een afstotende en aantrekkende kracht heerst tussen enerzijds het enzym en anderzijds de stabiliserende laag. Hierdoor zou een minimum in potentiële energie voorkomen waardoor het complex enzym-stabilisator gestabiliseerd wordt.

Figuur 2.5 geeft een overzicht van verschillende stabilisatiemethoden. (13,14)



Figuur 2.5: Schematische weergave van de stabilisatiemethoden voor enzymen (15)

2.1.3.1 Inkapseling

Een eerste stabilisatiemethode die belicht wordt, is de inkapseling (encapsulation op Figuur 2.5). Hierbij wordt er een film gevormd rondom het enzym. Het ontvouwen wordt verhinderd door het omsluiten met een laag polyethyleenglycolen (PEG), wat zorgt voor stabilisatie. Ook synthetische polymeren zoals polyvinylpyrrolidon hebben hetzelfde effect. De binding die aangegaan wordt met het enzym, wordt geclassificeerd als niet-covalent en kan een waterstofbrugbinding, een elektrostatische interactie of een hydrofobe interactie omvatten. De stabiliserende laag rondom het enzym zou daarenboven zogenoemde onproductieve bindingen voorkomen waardoor de enzymactiviteit vergroot wordt. (11,16,17)

2.1.3.2 Omgevingscondities

Ook is het mogelijk om de omgevingscondities te sturen waardoor de enzymen worden gestabiliseerd. Een mogelijk doel van deze sturing is om ongewenste proteolyse tegen te gaan. Dit kan bijvoorbeeld gesimuleerd worden door een pH-gebied te kiezen waarbij ongewenste proteolyse tegengegaan wordt maar waarbij de werking van de andere enzymen ongestoord blijft. Dit dient experimenteel vastgesteld te worden. Een andere oplossing om ongewenste proteolyse tegen te gaan, is modificatie van de andere enzymen in het mengsel. Zo kan een cofactor en/of ligand worden toegevoegd om ongewenste proteolyse tegen te gaan. In de volgende paragraaf wordt hier dieper op in gegaan. (18)

2.1.3.3 Ongewenste proteolyse

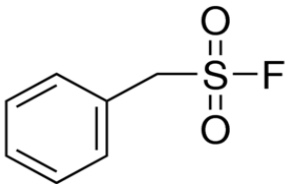
Een enzym is zelf ook een proteïne. Er dient dus extra aandacht gevestigd te worden op ongewenste proteolyse bij mengsels van enzymen. Proteolyse is de reactie die plaatsvindt wanneer een protease reageert en een proteïne afbreekt. In wasmiddelen worden vaak verscheidene enzymen toegevoegd zodat het wasmiddel verschillende vuilafzettingen

simultaan kan verwijderen. Bijvoorbeeld proteasen en lipasen worden telkenmale in een wasmiddel aangetroffen. Het is dan belangrijk dat een lipase bijvoorbeeld niet wordt afgebroken door een protease. Dit is de ongewenste proteolyse. (18,21)

Door sturen van de omgevingscondities of door een gepaste keuze van protease-inhibitoren, wordt de ongewenste proteolyse tegengegaan. Een inhibitor is een molecule die eveneens bindt aan het enzym en die ervoor zorgt dat de reactie niet kan doorgaan, bijvoorbeeld door een conformationele verandering van de actieve groeve. Deze verandering zorgt ervoor dat chemische groepen afgeschermd worden waardoor het enzym niet meer kan binden met het substraat. Dit proces is reversibel; door meer substraat toe te voegen, wordt het proces terug ongedaan gemaakt. Figuur 2.6 geeft visueel weer hoe dergelijke conformationele verandering eruitziet. (18,21)

Figuur 2.6: Conformationele verandering van de actieve groeve door inhibitie (22)

Figuur 2.6 toont dat na het binden van een inhibitor op de allosterische bindingsplaats, er een conformationele verandering wordt doorgevoerd waardoor het substraat niet meer kan binden op de actieve groeve. Het benzylsulfonyl fluoride is een voorbeeld van een protease inhibitor dat covalent bindt aan het enzym. Dit zal geen conformationele verandering teweegbrengen maar zal ervoor zorgen dat de actieve groeve niet meer bereikbaar is. Figuur 2.7 geeft de moleculaire structuur weer van deze inhibitor. (18)



Figuur 2.7: Moleculaire structuur van de inhibitor benzylsulfonyl fluoride (23)

2.1.3.4 Immobilisatie

Figuur 2.5 toont ook dat het mogelijk is om via immobilisatie van het enzym een stabilisatie te verkrijgen. Immobilisatie op een oppervlak zorgt ervoor dat een enzym gehinderd wordt in beweging. De chemische en fysische factoren van het oppervlak alsook de grootte van de proteïnen, spelen een rol bij de immobilisatie. De immobilisatie kan chemische of fysisch van aard zijn, met respectievelijk een covalente binding of elektrostatische interactie. Dit wordt eveneens weergegeven op Figuur 2.5. Daarnaast kan door chemische modificatie en cross-linking van de enzymen onderling, de stabiliteit eveneens verhoogd worden. (19)

2.1.3.5 Entrapment

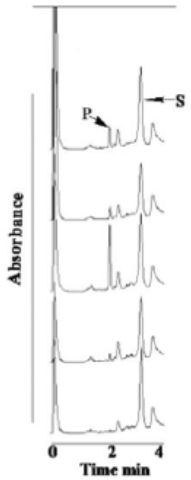
Daarnaast toont Figuur 2.5 nog een andere stabilisatiemogelijkheid, namelijk ‘entrapment’. Hierbij wordt een wijziging van de matrix doorgevoerd waardoor het enzym gehinderd wordt inzake denaturatie. Dit is een techniek die veelvuldig wordt toegepast in de wasmiddelenindustrie. Hiervoor komen glycerol, sorbitol en monopropyleenglycol (MPG) in aanmerking. Deze manier van stabiliseren wordt ook wel medium-engineering genoemd. Dit wordt uitgebreider verklaard voor glycerol. (19,20)

Een mogelijk verklaring zou kunnen steunen op het feit dat glycerol zorgt voor een ordening van de watermoleculen die in de matrix zitten en daarbij aanleiding geeft tot een preferentiële hydratatie van bepaalde aminozuren ten opzichte van andere. Dit wordt veroorzaakt door elektrostatische interacties die ervoor zorgen dat de glycerolmoleculen zich in een bepaalde manier gaan oriënteren rondom het enzym en zo aansturen op een meer compacte formatie van het enzym zelf. Door die compactere vorm zou er een stabilisatie optreden. Glycerol zou vooral interacties aangaan met de actieve groeve van het enzym en als een soort amfifiele (tweeslachtige) film interageren. De hoeveelheid glycerol die toegevoegd dient te worden hangt af van de aard van het enzym (hydrofiliciteit). Andere stabilisatoren met hetzelfde effect zoals glycerol, zijn onder andere mannitol, xylitol en adonitol. (19,20)

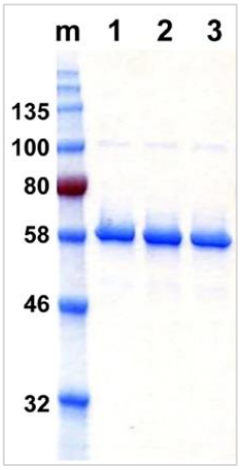
2.1.4 Kwaliteits- en activiteitscontrole

De productie van enzymen wordt steeds gevolgd door een kwaliteitscontrole. Deze kan gebeuren op verschillende manieren. Het gebruik van een referentiestaal is vaak de start van een kwaliteitscontrole. Dit is een referentie-enzym waarvan er zekerheid is naar effectiviteit toe. Ten eerste dient deze referentie zelf onderworpen te worden aan een aantal testen. Dit kan een size-exclusion chromatografie (SEC) en een analytische ultracentrifuge (AUC) omvatten. (24)

Een belangrijk aspect van de kwaliteitscontrole is de identificatie van het geproduceerde enzym. Dit dient immers het beoogde product te zijn opdat deze de juiste reacties kan katalyseren. Voor de identificatie van het enzym zijn allerhande technieken geschikt, onder andere vloeistofchromatografie in combinatie met massaspectrometrie (LC-MS) en omgekeerde fase – hoge druk vloeistofchromatografie in combinatie met massaspectrometrie (RP-HPLC-MS). De LC of HPLC dient om het te analyseren monster te scheiden in de samenstellende bestanddelen. De MS doet dienst als detector waarbij het monster uiteenvalt in karakteristieke molecuulfragmenten. (6,24,25)



Figuur 2.8: Chromatogram van een HPLC-kwaliteitscontrole (25)



Figuur 2.9: SDS-PAGE van proteïnen; m-baan is een referentie en baan 1 tot 3 staan voor verschillende tijdstippen tussen 1 uur en 4 uur (26)

Naast de identiteit is ook de zuiverheid van belang. Dit kan ook getest worden met een HPLC-instrument. Een voorbeeld van een kwaliteitstest met HPLC wordt gegeven in Figuur 2.8. Er worden hierbij vijf stalen vergeleken waarbij het bovenste staal het referentiestaal is. (6,24,25)

Daarnaast zijn er nog andere technieken de zuiverheid analyseren. Daartoe behoort de natriumdodecylsulfaat-polyacrylamide-gelelektroforese (SDS-PAGE). Deze techniek is overigens geschikt voor lage resoluties en lage concentraties. De onderste limiet ligt ongeveer op 50 μg/ml. Een belangrijk nadeel van SDS-PAGE is de benodigde tijd om de techniek uit te voeren. Dergelijke analysetijden kunnen oplopen tot enkele uren. Een voorbeeld van een SDS-PAGE wordt gegeven in Figuur 2.9. Hierbij wordt gekeken of een proteïne nog steeds dezelfde activiteit vertoont in een tijdspanne van 1 uur tot 4 uur na toevoeging aan een vaccin. (6,24,26)

Nog een andere parameter die getest wordt, is de enzymactiviteit. Deze worden onderzocht via activiteitsmetingen die bijvoorbeeld gedaan worden met calorimetrische methoden. De activiteit is immers een maat voor de werking van het enzym. Een veel gebruikte calorimetrische techniek, is isotherme titratie calorimetrie (ITC). Daarbij wordt de vrijgekomen of geabsorbeerde warmte gemeten in een adiabate cel. Deze warmte is gecorreleerd met het breken of vormen van bindingen in het enzym. De techniek berust op de vergelijking van de te bestuderen reactie ten opzichte van een referentiecel. De hoeveelheid warmte wordt geregistreerd en verwerkt waaruit de Michaelis-Menton constante (K_M) en de katalytische constante (k_{cat}) volgen. ITC is een robuuste techniek en kan zowel endotherme als exotherme reacties verwerken. Ook is het mogelijk om activiteitsmetingen te doen door middel van conductometrie. Het is namelijk vaak zo dat de ladingen van de reagerende moleculen veranderen, waardoor het ionisch karakter verandert. (6,24,27,28)

Daarnaast kan ook UV-spectrofotometrie gebruikt worden om activiteitsmetingen te doen. Wanneer bijvoorbeeld de activiteit van het katalase gemeten wordt, kan aan UV-spectrofotometrie gedaan worden. Dit enzym zet waterstofperoxide om naar water en zuurstof. Dit is met andere woorden een reductie. Aan dit mengsel wordt een verbinding toegevoegd die geoxideerd kan worden. Deze verbinding dient te absorberen bij een bepaalde golflengte zonder dat er storingen zijn van andere componenten. Door een monitoring van de verandering van absorbantie bij die specifieke golflengte, kan de activiteit afgeleid worden. (29)

2.1.5 Voordelen van het gebruik van enzymen

Enzymen worden in de natuur aangetroffen en zijn biologisch van aard. In de wasmiddelenindustrie worden enzymen al decennialang ingezet. Na intensief onderzoek verkleinde de hoeveelheid benodigd enzym stelselmatig. De enzymen omvatten nu minder dan 1 % van het totale wasmiddel. Er dient daarenboven steeds minder enzym gebruikt te worden voor dezelfde reiniging te voltrekken. Dit is een positief punt voor het milieu, namelijk minder organische lozingen in het afvalwater. (2,3)

Daarnaast leidt het gebruik van enzymen er ook toe dat er minder energie verbruikt moet worden. Deze opereren namelijk bij een lagere temperatuur en daarenboven wordt de wastijd ingekort. Daardoor zou er ongeveer vier keer minder energie verbruikt worden tijdens het wasproces. Ook is er ongeveer vier keer minder water nodig om te wassen door het gebruik van enzymen. (2,3)

Het detergent zelf heeft door de jaren heen een milder alkalisch karakter gekregen, wat opnieuw bij de lozing van het afvalwater een pluspunt is. Een ander voordeel is de specificiteit van de reacties. Hierdoor worden de vezels van de kledij weerhouden van ongewenste reacties. (2,3)

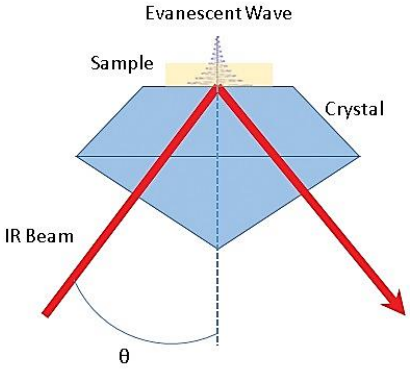
2.2 ATR-FTIR: werking en kwantificatie

2.2.1 Werkings- en meetprincipe

Infraroodspectroscopie wordt veelal gebruikt onder de vorm van ATR-FTIR. ATR staat voor attenuated total reflectance ofwel letterlijk vertaald: verzwakte totale reflectie. De verzwakking vindt zijn oorsprong in het feit dat de laserstraal meermaals in het monster indringt. Bij elke indringing wordt een deel van de energie geabsorbeerd. Daardoor is de laserstraal in de eindsituatie verzwakt tegenover de beginsituatie. FTIR is de afkorting van Fourrier transformed infrared spectroscopy ofwel Fourrier transformatie infraroodspectroscopie. Deze techniek laat toe om op een eenvoudige manier, zonder veel staalvoorbereiding, een analyse te voltrekken en resultaten te verkrijgen onder de vorm van een infraroodspectrum. In veel gevallen kan er op een directe wijze staal aangebracht worden als vloeistof of in vaste vorm. (30–32)

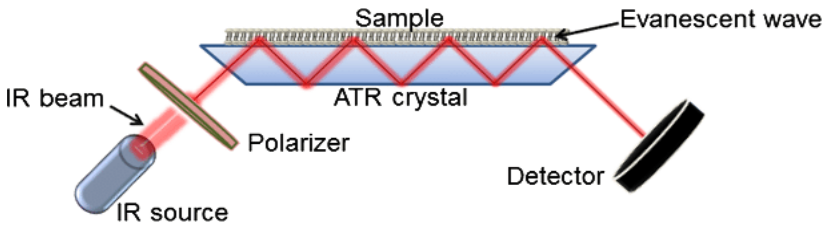
Het werkingsprincipe wordt voorgesteld in Figuur 2.10. Er wordt gemeten door een laserstraal (golflengte in het infrarode gebied) te richten op een kristal onder een hoek die voorafgaand ingesteld wordt. Vaak wordt voor een hoek van 45 ° gekozen (θ op Figuur 2.10). De laserstraal wordt zodanig gericht dat deze meermaals breekt in het kristal. Er wordt gekozen voor een kristal met een hogere brekingsindex dan het te onderzoeken staal. Op die manier wordt totale interne breking verkregen. De laserstraal gaat van een dicht medium (kristal) over naar een licht medium (staal) waardoor er een afbuiging plaatsvindt en bijgevolg interne breking. Dit is naast de juiste hoek instellen een tweede vereiste om totale interne breking te verkrijgen. Afhankelijk van de toepassing wordt gekozen voor een ander kristal. Typische kristallen die hiervoor in aanmerking komen, zijn onder andere germanium, silicium en zink-silicium kristallen. Ook diamant is hiervoor geschikt en wordt veelvuldig gebruikt. (30,31,33)

De gebroken straal zorgt voor reflecties en deze passeren het monster verscheidene keren. De straal dringt ongeveer 0,5 µm tot 2 µm door in het aangebrachte staal. De laserstraal die doordringt in het monster, wordt aangeduid met de benaming van evanescent wave op Figuur 2.10 (in het Nederlands wordt de term verdwijnende golf gehanteerd). (30,31,33)



Figuur 2.10: Werkingsprincipe van ATR (33)

Door het direct contact tussen het monster en het kristal worden de gereflecteerde stralen maximaal geabsorbeerd. Dit verklaart het belang van de direct applicatie. De herhaaldelijke passage zorgt voor een grote resolutie van het opgenomen spectrum. De stralen die niet geabsorbeerd worden door het staal, worden teruggekaatst naar een detector. Deze zorgen vervolgens voor de vorming van het infraroodspectrum met behulp van Fourrier-transformaties. Dit is te zien op Figuur 2.11. (30,31)



Figuur 2.11: Vereenvoudigd werkingsprincipe van een ATR-FTIR-toestel (31)

Figuur 2.11 toont dat de infrarode straal wordt gebroken op en in het kristal. Het kristal zorgt enerzijds voor interne reflecties en anderzijds voor een herhaaldelijke indringing van de infrarode straal in het monster. Hierdoor wordt een resulterende straal verkregen die karakteristiek is voor het staal en aanleiding geeft tot een duidelijk spectrum. De gebundelde straal verlaat het kristal in de richting van de detector. (30,34)

De krachtige techniek van de Fourrier-transformaties kent vele voordelen waaronder het feit dat er bespaard wordt op solventen en monsters (minder aanbrengen op het kristal). ATR-FTIR is overigens minder tijdrovend dan bijvoorbeeld HPLC, is niet-destructief en het spectrum geeft daarenboven veel informatie over de functionele groepen. (30,34)

Tabel 2.3 geeft een overzicht van de verschillende regio's van het infrarode gebied. FTIR-metingen worden uitgevoerd in het midden-infraroodgebied. Vooral de golfgetallen zijn belangrijk bij het opnemen van een infraroodspectrum daar deze veelal in de x-as staan van dergelijk IR-spectrum. Het golfgetal is niets anders dan het omgekeerde van de golflengte en kan berekend worden via Vergelijking 2.1. (34,35)

$$\bar{\nu} = \frac{1}{\lambda} \qquad \textbf{(2.1)}$$

Met $\bar{\nu}$ het golfgetal in cm^{-1} en λ de golflengte in cm

Tabel 2.3: Overzicht van de regio's van het infraroodspectrum (34,35)

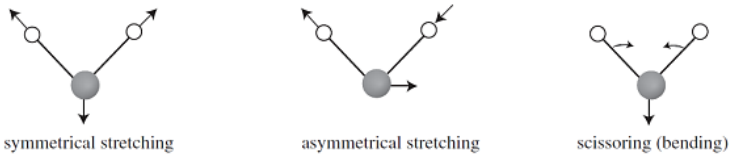
Regio	Golflengte λ (µm)	Golfgetal $\bar{\nu}$ (cm^{-1})	Analysesoort
Nabij	0,78 – 2,50	12800 - 4000	Kwantitatief en kwalitatief
Midden	2,50 - 50	4000 - 200	Kwantitatief en kwalitatief
Ver	50 - 100	200 -10	Kwalitatief

Tabel 2.3 toont aan dat louter het nabije en midden infrarode gebied voor kwantitatieve doeleinden geschikt zijn. Aangezien FTIR vooral in het midden IR-gebied gebruikt wordt, kan deze dus ook dienen ter kwantificatie. Dit heeft te maken met de energie van de infrarode stralen in dat gebied. Immers geldt dat hoe kleiner de golflengte is, des te hoger de energie van de golf zal zijn. (35–37)

Belangrijk om aan te geven, is dat niet elk energieniveau toegestaan is bij een energie-overgang in een molecule. Slechts enkele discrete waarden van de energie zijn toegelaten. De banen in een molecule hebben elk een bepaald energieniveau waardoor de uitwisseling van energie bij absorptie en relaxatie gekwantificeerd zal zijn. (35,38)

De excitatiepatronen zijn gecorreleerd met deze van een (niet-)harmonische oscillator. Een oscillator kan gezien worden als een gewicht dat aan een veer verbonden is. Toegepast op een molecule stellen de twee atomen de gewichten voor en fungeert de binding tussen de twee atomen als veer. Er zijn verschillende soorten vibraties mogelijk waaronder de

strekkingsvibratie (symmetrisch of asymmetrisch), schaarvibraties, kwispelvibraties en twistvibraties. De strekkingsvibratie en schaarbeweging worden weergegeven in Figuur 2.12. (35,38)



Figuur 2.12: Weergave van de strekkingsvibraties en de schaarvibratie (39)

De strekkingsvibraties van een functionele groep zijn kenmerkend en leiden zo tot een gebied waarbinnen (bijna) alle strekkingsvibraties voorkomen. Deze regio strekt zich uit van ongeveer 4000 cm^{-1} tot 1500 cm^{-1}. De regio die daarop volgt van 1500 cm^{-1} tot 400 cm^{-1}, wordt het fingerprintgebied genoemd. Dit gebied geeft minder aanleiding tot identificatie van functionele groepen maar geeft bijvoorbeeld informatie over primaire of secundaire positie van de hydroxylgroep. De regio heeft dus een sterke waarde met betrekking tot de identiteit van de molecule, vandaar de benaming fingerprintregio. (35)

Figuur 2.13 geeft een overzicht van de verschillende functionele groepen en de daarbij horende karakteristieke plaatsen van de absorptiepieken. Ook de eigenschappen van de pieken zijn terug te vinden in Figuur 2.13. (35)

IR Absorptions of Common Functional Groups

Functional Group	*Absorption Location (cm^{-1})*	*Absorption Intensity*
Alkane (C–H)	2,850–2,975	Medium to strong
Alcohol (O–H)	3,400–3,700	Strong, broad
Alkene (C=C)	1,640–1,680	Weak to medium
(C=C–H)	3,020–3,100	Medium
Alkyne		
(C≡C)	2,100–2,250	Medium
(C≡C–H)	3,300	Strong
Nitrile (C≡N)	2,200–2,250	Medium
Aromatics	1,650–2,000	Weak
Amines (N–H)	3,300–3,350	Medium
Carbonyls (C=O)		Strong
Aldehyde (CHO)	1,720–1,740	
Ketone (RCOR)	1,715	
Ester (RCOOR)	1,735–1,750	
Acid (RCOOH)	1,700–1,725	

Figuur 2.13: Overzicht van de verschillende functionele groepen en de plaats van de desbetreffende pieken in het IR-spectrum (40)

Door absorptie van de infrarode straal wordt de molecule in geëxciteerde toestand gebracht. Deze toestand is onstabiel en wordt bereikt door interne resonanties en een verandering van dipoolmoment van de moleculen. Door relaxatie wordt een deel van de energie, of alle energie afgestaan onder de vorm van straling. Zodoende zal de molecule terugkeren naar een stabiele grondtoestand. Het is deze straling die de detector bereikt op Figuur 2.11. (34,41)

Typisch gaan sterk geconjugeerde systemen en functionele groepen binnen de organische chemie absorberen binnen dit gebied. Hieronder vallen onder andere: carbonzuren, esters, alkenen, alkynen, aldehyden en ketonen. Veelal zullen anorganische componenten niet absorberen in het infrarode gebied. Moleculen zoals water en koolstofdioxide zullen wel absorptiepieken vertonen. Dit komt omdat er binnen deze moleculen wel energie-overgangen

bestaan na bestraling met infrarood licht. Een molecule zoals stikstofgas, zal deze verandering niet voltrekken en er zal dus bijgevolg ook geen absorptie zijn. (34,41)

2.2.2 Kwantificeren met ATR-FTIR

De voordelen van de ATR-FTIR-techniek hebben voornamelijk betrekking op het kwalitatief evalueren van een bepaald staal. Echter zou het gunstig zijn om ATR-FTIR ook te kunnen gebruiken om kwantitatieve analyses te doen. Het feit dat er op een snelle en niet-destructieve manier een staal kan worden geanalyseerd, biedt een groot voordeel ten opzichte van de meer conventionele technieken om te kwantificeren zoals HPLC. (34,42)

In een eerste benadering kan een kwantificatie starten door het opstellen van een ijklijn. Dit gebeurt door gebruik te maken van standaarden met gekende concentraties. Dit is een methode die gangbaar is in het UV-VIS gebied. Bijvoorbeeld in een situatie waarin een onbekend ethanol-mengsel dient gekwantificeerd te worden, zullen ethanol-standaarden worden gebruikt ter constructie van de ijklijn. Vervolgens wordt gekeken naar de piek in het IR-spectrum dat geassocieerd is met de hydroxylgroep. Deze wordt gebruikt ter kwantificatie. De wet van Lambert-Beer wordt gebruikt om een verband te leggen tussen de concentratie en de absorbantie. Deze wordt wet gegeven in Vergelijking 2.2. (34,42)

$$A = \varepsilon * l * c \qquad \textbf{(2.2)}$$

Waarbij A staat voor de absorbantie (-),

ε is de molaire absorptiecoëfficiënt (l/(cm*mol)),

l staat voor de weglengte afgelegd door het licht (cm) en

c is de concentratie (mol/l)

De lengte l is een bekende en ook de molaire absorptiecoëfficiënt ε kan opgezocht of experimenteel bepaald worden. Zodoende blijven enkel de absorbantie A over die wordt opgemeten alsook de onbekende concentratie c. Er kan dus eenduidig een verband tussen deze grootheden worden opgesteld. Een onbekend ethanol-staal kan op deze manier geanalyseerd worden. Inzake de kwantificatie met FTIR gaat dit echter niet zo eenvoudig. (34,42,43)

Voor FTIR-metingen zijn er restricties gebonden aan het werken met een ijklijn. Er wordt slechts gekeken naar één bepaalde piek om deze te correleren met een concentratie. Het is echter beter om te werken met een serie van pieken waarmee de concentratie gecorreleerd kan worden. Daarnaast is de analyse naar beneden begrensd afhankelijk van de detectielimiet (LOD) en kwantificatielimiet (LOQ). Daarenboven veronderstelt de van wet Lambert-Beer een lineair verband tussen concentratie en absorbantie, wat niet altijd het geval is voor kleine en grote concentraties. Ook gaan storingen, die groter worden naarmate de concentratie kleiner wordt, de meting drastisch beïnvloeden. Verschuivingen van de pieken kunnen eveneens zorgen voor een vertekend beeld. De afwijkingen die met deze manier van werken gepaard gaan, hebben een belangrijk effect op de analyse. Indien er daarenboven (te snel) twee fasen zouden ontstaan in het aangebrachte monster, zal de analyse incorrecte resultaten opleveren. Dit aangezien de laserstraal slechts enkele micrometers in het monster doordringt waardoor een groot deel van de informatie verloren gaat. (34,42,44)

Er kan een standaard root mean squared (RMS) methode gehanteerd worden om te berekenen wat de fout is op de meetresultaten. Ook is het mogelijk om gebruik te maken van

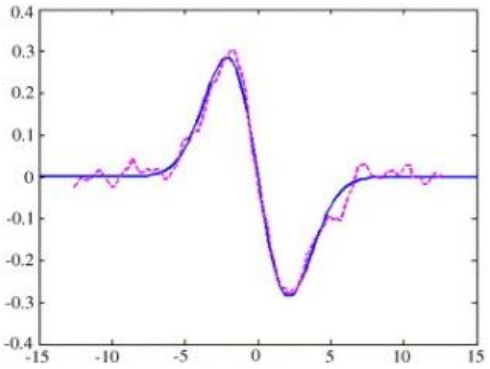
principale component analyse (PCA) en partial least squares (PLS). Dit maakt dat er met behulp van software berekend wordt welke zones van het spectrum of series van pieken relevant zijn voor kwantificatie. Op de techniek van PLS en PCA wordt in paragraaf 2.3.3.2 van deze thesis verder ingegaan. Door PCA of PLS toe te passen, wordt duidelijk welke gebieden van het spectrum kunnen gebruikt worden als referentie om te kwantificeren en welke niet. (34,42,45,46)

De PLS-methode lijkt alvast een goede basis om te gebruiken in kwantitatief onderzoek met een FTIR-toestel. PLS maakt de kwantificatie met FTIR krachtiger. De concentraties moeten echter hoog genoeg zijn opdat de absorptiepieken op hun beurt hoog genoeg zouden zijn. Alleen dan kan de kwantificatie nauwkeurig gebeuren. Wat precies hoog genoeg is, hangt af van staal tot staal. Om te kijken of FTIR en HPLC vergelijkbare resultaten geven qua kwantificatie, wordt gebruik gemaakt van het onderzoek *Quantification of brain lipids by FTIR spectroscopy and partial least squares regression*. In dit onderzoek werden lipiden gekwantificeerd via FTIR-metingen. Figuur 2.14 geeft een overzicht van de resultaten van het onderzoek. De laatste kolom geeft de resultaten weer gebruik makend van HPLC. (47)

Lipids	Gray matter	
	PLS	HPLC
CH	20.7 ± 1.7	19.6
PE	30.2 ± 3.1	30.7
PC	14.5 ± 2.1	25.1
GC	12.5 ± 1.1	7.2
SM	5.4 ± 1.2	3.2
PS	–	7.2
SUL	–	0.7
Sum	83.3 ± 7.1	93.7

Figuur 2.14: Weergave van het verschil in kwantitatief resultaat tussen ATR-FTIR in combinatie met PLS en HPLC (47)

Op Figuur 2.14 is te zien dat er een vrij grote onzekerheid is op het gesommeerde resultaat (een standaarddeviatie van 7,1). Daarbij valt op dat sommige lipiden accurater kunnen worden bepaald en grotere overeenkomsten vertonen met HPLC dan andere. Het zal dus afhangen van staal tot staal en van verbinding tot verbinding, met welke zekerheid de concentratie kan worden bepaald. Verder bleek ook dat de grootte van de molaire absorptiecoëfficiënt hieraan gecorreleerd is. Hoe groter de waarde ervan is, des te nauwkeuriger de bepaling van de concentratie wordt. (47)



Figuur 2.15: De transformatie van een signaal met ruis (rood) naar een glad signaal (blauw) door de SG-filter (48)

Nadelen van de ATR-FTIR-techniek zijn onder andere ruis en een basislijnverschuiving. Een bandfilter zou de oplossing kunnen zijn waarvan de Savitzky-Golay-filter (SG-filter) een voorbeeld is. Dergelijke filter slaagt erin om achtergrondsignalen en ruis van het signaal te

verwijderen. Een Savitzky-Golay-filter is een bandfilter die het signaal gladder maakt. De wiskunde erachter is gebaseerd op gedifferentieerde polynomen. Aangezien de wiskunde achter deze techniek al snel ingewikkeld wordt, zal hier niet verder op ingegaan worden. Deze filter kan onder andere door de software van Matlab worden gesimuleerd. Figuur 2.15 geeft visueel weer wat er gebeurt bij de transformatie van het oorspronkelijke signaal (rood) naar het getransformeerde signaal (blauw). (48–50)

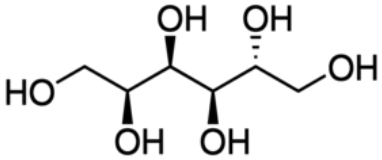
Uit al de voorgenoemde onderzoeken blijkt dat ATR-FTIR wel degelijk kan dienen ter kwantificatie. Wanneer gewerkt wordt met de PCA- en PLS-methode, wordt de kwantificatie nauwkeuriger. Filters kunnen er daarenboven voor zorgen dat ruis en basislijnverschuivingen verminderen.

2.2.3 Interpretatie van de structuren en de spectra van sorbitol, glycerol en monopropyleenglycol

Bij de kwantificatie met ATR-FTIR zijn de infraroodspectra van wezenlijk belang. Het is dus interessant om te kijken hoe de spectra van de stabilisatoren sorbitol, glycerol en monopropyleenglycol (MPG) eruitzien. Er wordt vooral getracht een globale interpretatie te geven aan de spectra. De beschrijving van de pieken in het fingerprintgebied wordt in deze paragraaf minder uitvoerig gedaan. De bedoeling is om een overeenkomst aan te tonen tussen enerzijds de moleculaire structuur en anderzijds het IR-spectrum.

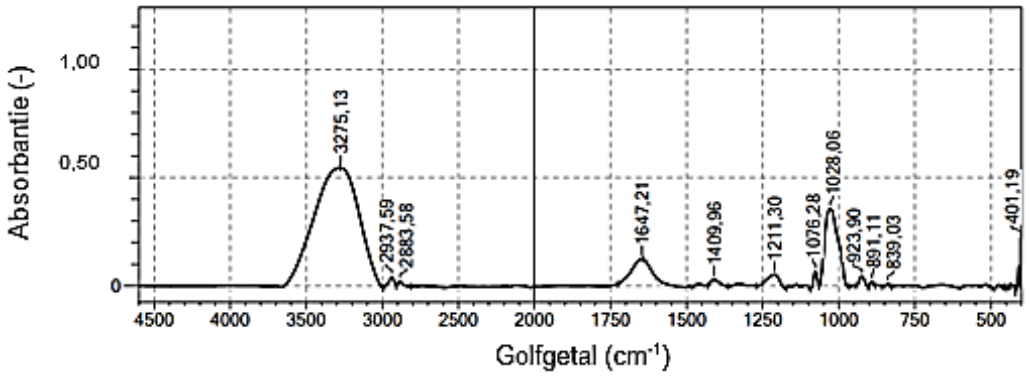
2.2.3.1 Sorbitol

Sorbitol is een suikeralcohol waarvan de moleculaire structuur gegeven wordt in Figuur 2.16.



Figuur 2.16: Moleculaire structuur van sorbitol (51)

Figuur 2.16 geeft weer dat sorbitol zes hydroxylgroepen bevat, waarvan er twee primair en vier secundair zijn. Primaire groepen zijn gebonden op een koolstofatoom dat op zich gebonden is op een ander koolstofatoom. Secundair gebonden groepen daarentegen zijn gebonden op een koolstofatoom dat zelf gebonden is op twee andere koolstofatomen. De moleculaire structuur van sorbitol toont dat er zes koolstofatomen aanwezig zijn. Het infraroodspectrum van sorbitol wordt gegeven in Figuur 2.17. (38)



Figuur 2.17: Infraroodspectrum van sorbitol

Een eerste belangrijke piek is deze van de hydroxylgroep die zich situeert rond 3275 cm^{-1}. Deze piek is vrij breed doch intens wat typisch is voor een hydroxylgroep. Wat opvalt is dat deze piek veel prominenter aanwezig is dan de alkaanpieken rond 2937 cm^{-1} en 2883 cm^{-1}. Het alkaangedeelte heeft relatief gezien dus minder aandeel in de structuur. Deze pieken wijzen op het voorkomen van ofwel CH_3-groepen en/of CH_2-groepen. Daarenboven is er geen piek aanwezig is rond 725 cm^{-1}, wat wijst op het feit dat er minder dan vier CH_2-groepen in de structuur zitten. De aanwezigheid van de piek rond 1076 cm^{-1} wijst op het voorkomen van een secundair alcohol, zoals de piek rond 1028 cm^{-1} duidt op de aanwezigheid van primaire hydroxylgroepen. Al deze bevindingen worden samengevat in Tabel 2.4 en zijn in overeenstemming te brengen met de moleculaire structuur die getoond wordt in Figuur 2.16.

Tabel 2.4: Beknopte analyse van het IR-spectrum van sorbitol

Vibratie	Golfgetal (cm^{-1})	Vibratie	Golfgetal (cm^{-1})
ν_{OH}	3275,13	ν_{C-O} (sec)	1076,28
ν_{a,CH_2}	2937,59	ν_{C-O} (prim)	1028,06
ν_{s,CH_2}	2883,58		

Ter vervollediging wordt Tabel 2.5 gegeven die theoretisch weergeeft waarvoor de pieken van de -CH_2OH-binding in de fingerprintregio en de regio daarboven staan. Ook Tabel 2.6 heeft diezelfde insteek voor de -CHOH-binding. Zo verklaart Tabel 2.5 bijvoorbeeld waarvoor de piek rond 900 cm^{-1} staat, namelijk een rock-vibratie van de CH_2-groep. (38)

Tabel 2.5: Overzicht van de pieken geassocieerd met de -CH_2OH-binding (38)

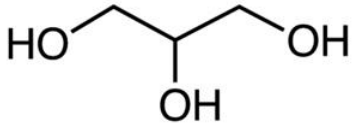
Vibratie	Golfgetal (cm^{-1})	Vibratie	Golfgetal (cm^{-1})	Vibratie	Golfgetal (cm^{-1})
ν_{OH}	3350 ± 70	δ_{OH}	1400 ± 40	ρ_{s,CH_2}	895 ± 65
ν_{a,CH_2}	2945 ± 45	ω_{CH_2}	1335 ± 55	γ_{OH}	635 ± 35
ν_{s,CH_2}	2885 ± 45	τ_{CH_2}	1240 ± 60	δ_{C-O}	465 ± 55
δ_{CH_2}	1445 ± 35	ν_{C-O}	1045 ± 45		

Tabel 2.6: Overzicht van de pieken geassocieerd met de -CHOH-binding (38)

Vibratie	Golfgetal (cm^{-1})	Vibratie	Golfgetal (cm^{-1})	Vibratie	Golfgetal (cm^{-1})
ν_{OH}	3370 ± 30	ω_{CH}	1365 ± 35	γ_{OH}	630 ± 30
ν_{CH}	2940 ± 40	δ_{CH}	1320 ± 30	δ_{C-O}	470 ± 30
δ_{OH}	1400 ± 30	ν_{C-O}	1110 ± 30	γ_{C-O}	360 ± 30

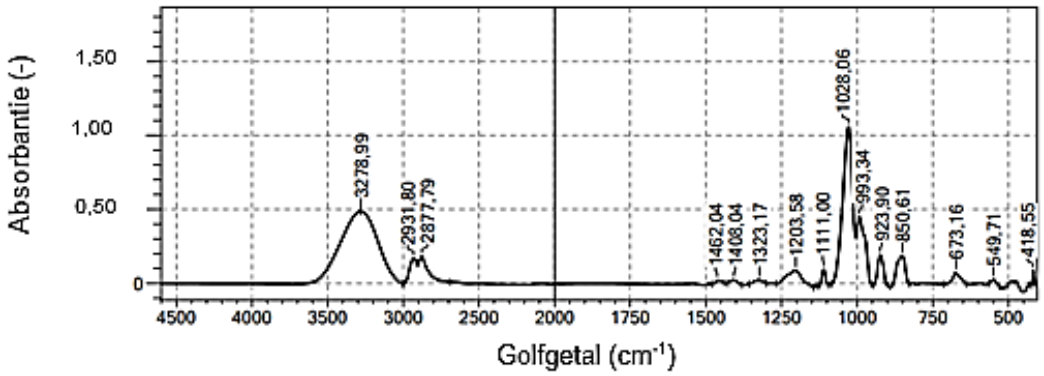
2.2.3.2 Glycerol

De moleculaire structuur van glycerol bevat dezelfde grote groepen zoals sorbitol, toch zijn er enkele belangrijke verschillen. De structuur van glycerol kan worden teruggevonden in Figuur 2.18.



Figuur 2.18: Moleculaire structuur van glycerol (52)

Figuur 2.18 geeft weer dat er in de moleculaire structuur van glycerol drie hydroxylgroepen aanwezig zijn. Daarenboven zijn er nu twee primair en slechts één secundair gebonden. Figuur 2.18 toont eveneens dat er drie koolstofatomen zijn waarvan er twee voorkomen als CH_2 en één voorkomt onder de vorm van CH. Figuur 2.19 visualiseert het IR-spectrum van glycerol.



Figuur 2.19: Infraroodspectrum van glycerol

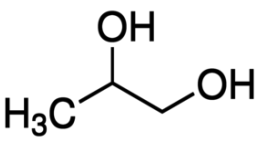
In Figuur 2.19 zijn verschillende pieken aangeduid die van belang zijn bij de interpretatie van het spectrum. De piek die correspondeert met de hydroxylgroepen, werd gevonden nabij 3278 cm^{-1}. De pieken tussen 2877 cm^{-1} en 2931 cm^{-1} zijn afkomstig van de strekkingsvibraties van de CH-groep. Verder valt op dat er pieken zijn die uitstrekken rond 1462 cm^{-1} en 1408 cm^{-1}. Deze zijn geassocieerd met de buigvibratie van de CH_2- en OH-binding. Verder is de piek die afkomstig is van het primaire alcohol (1028 cm^{-1}) duidelijk aanwezig. De piek corresponderend met de secundaire hydroxylgroep (1111 cm^{-1}) is kleiner dan die van de primaire OH-groep. Opnieuw is er een afwezigheid van de piek rond 725 cm^{-1} waaruit blijkt dat het aantal CH_2-groepen kleiner is dan vier. Ook hier wordt een correlatie gevonden tussen de moleculaire structuur en het infraroodspectrum. De bevindingen worden samengevat in Tabel 2.7. Voor de betekenis van de andere pieken wordt verwezen naar Tabel 2.5 en Tabel 2.6. (38,53)

Tabel 2.7: Beknopte analyse van het IR-spectrum van glycerol

Vibratie	Golfgetal (cm^{-1})	Vibratie	Golfgetal (cm^{-1})
$\boldsymbol{\nu_{OH}}$	3278,99	δ_{OH}	1408,04
$\boldsymbol{\nu_{a,CH_2}}$	2931,80	$\boldsymbol{\nu_{C-O}}$ (sec)	1111,00
$\boldsymbol{\nu_{s,CH_2}}$	2877,79	$\boldsymbol{\nu_{C-O}}$ (prim)	1028,06
$\boldsymbol{\delta_{CH_2}}$	1462,04		

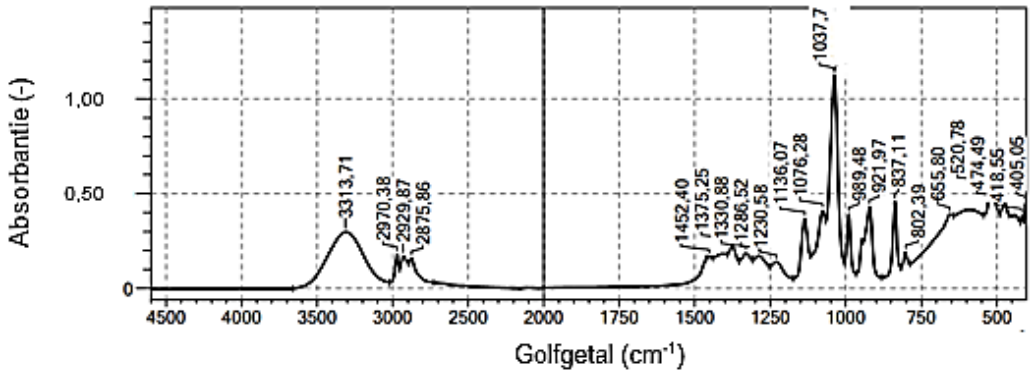
2.2.3.3 Monopropyleenglycol

Net zoals sorbitol en glycerol, wordt monopropyleenglycol (MPG) gekenmerkt door het voorkomen van verschillende hydroxylgroepen. De moleculaire structuur van MPG wordt gegeven in Figuur 2.20.



Figuur 2.20: Moleculaire structuur van monopropyleenglycol (54)

Op Figuur 2.20 is te zien dat er zowel een primaire als een secundaire alcoholgroep aanwezig is. Er is in deze structuur een CH-, een CH_2- en een CH_3-groep aanwezig. Het IR-spectrum wordt gegeven in Figuur 2.21.



Figuur 2.21: Infraroodspectrum van monopropyleenglycol

Allereerst toont Figuur 2.21 de prominent aanwezige hydroxylpiek rond 3313 cm^{-1}. Ook hier is er sprake van een brede, intense piek. Er valt wel op dat deze piek niet groter is dan de piek van het alkaandeel gevonden bij 2900 cm^{-1}. Het alkaangedeelte en hydroxyldeel van de molecule worden dus ongeveer gelijk vertegenwoordigd. De piek rond 2970 cm^{-1} is afkomstig van de CH_3-groep. Nog valt op dat er geen piek aanwezig is rond 725 cm^{-1}. Er zijn dus ook hier minder dan vier CH_2-groepen aanwezig in de structuur van de molecule. De aanwezigheid van de piek rond 1136 cm^{-1} wijst op het voorkomen van een secundair alcohol. De piek rond 1037 cm^{-1} wijst dan weer op de aanwezigheid van een primaire hydroxylgroep. De serie pieken rond 650 cm^{-1} zijn te wijten aan de ruimtelijke buigvibraties van de OH-groepen. Deze bevindingen worden samengevat in Tabel 2.8. Andere pieken die zichtbaar zijn en afkomstig zijn van de CH_2OH-groep en van de CHOH-groep zijn terug te vinden respectievelijk Tabel 2.5 en in Tabel 2.6. (38)

Tabel 2.8: Beknopte analyse van het IR-spectrum van monopropyleenglycol

Vibratie	Golfgetal (cm^{-1})	Vibratie	Golfgetal (cm^{-1})
$\boldsymbol{\nu_{OH}}$	3313,71	$\boldsymbol{\nu_{C-O}}$ (sec)	1136,07
$\boldsymbol{\nu_{a,CH_3}}$	2970,38	$\boldsymbol{\nu_{C-O}}$ (prim)	1037,70
$\boldsymbol{\nu_{a,CH_2}}$	2929,87	$\boldsymbol{\gamma_{OH}}$	655,80
$\boldsymbol{\nu_{s,CH_2}}$	2875,86		

2.3 Basislijncorrecties en pre-processingsmethoden

Om aan kwantificatie te kunnen doen met deep learning, is er eerst een voorbehandeling nodig van de data. Daartoe kunnen verschillende aspecten behoren waaronder een basislijncorrectie, data-scaling en pre-processingsmethoden. Deze aspecten worden in wat volgt besproken. Deze stappen dienen vooraf te gaan aan de opbouw van een neuraal netwerk. Figuur 2.22 geeft een overzicht van het proces in een flow-diagram.

Figuur 2.22: Voorbereiding van data ter opbouw van een neuraal netwerk

Het is belangrijk om aan te geven dat in het kader van deze thesis, indien er een basislijncorrectie wordt uitgevoerd, deze voor de pre-processingsstap en/of data-scaling zal worden voltrokken. Immers wordt eerst een 'nieuw' spectrum geconstrueerd alvorens een datareductie of dataselectie te doen. Op die manier zal het model robuuster zijn aangezien er geen verlies is aan informatie door een voorafgaande pre-processing.

2.3.1 Probleem van basislijnverschuiving en reproduceerbaarheid

Een basislijn is van groot belang bij de kwantificatie aangezien er steeds gerefereerd wordt naar een piekhoogte (en dus de absorbantie). Om deze te kunnen karakteriseren, is er dus een verwijzing nodig naar de basislijn. De piekhoogte is de hoogte van de top van een piek tot de basislijn. Bij een kwantificatie is het dus van belang dat de basislijn correct ingeschat wordt. Daarenboven moet de inschatting steeds op dezelfde manier gebeuren zodanig dat de kwantificering eenduidig is voor elk staal. (55–57)

Ook de reproduceerbaarheid en de herhaalbaarheid moeten onder controle worden gehouden. Immers mag de tijdsdimensie of de persoon die de analyse uitvoert, geen invloed hebben op het eindresultaat. Er zijn verschillende methoden hiervoor aangewezen zoals polynoom-fitting en interpolatie-fitting. Echter zijn deze methoden semi-manueel en dus tijdsintensief. Ook zijn deze methoden subjectief en daardoor meestal niet-reproduceerbaar. (55–57)

Ontwikkelde computeralgoritmen zijn objectief, snel en geven aanleiding tot reproduceerbare resultaten. Daarom wordt de aandacht gevestigd op deze methoden. Een kort overzicht van de methoden die zullen worden besproken, wordt weergegeven in Tabel 2.9. Ook wordt aangegeven wat de voor- en nadelen zijn van deze basislijncorrecties. (55–57)

Tabel 2.9: Overzicht van de basislijncorrecties en van de belangrijkste voor- en nadelen (55–60)

Basislijncorrectie	Voordelen	Nadelen
Manuele correctie	Inzicht in het spectrum	Trage correctie, ruis blijft aanwezig
AsLS (Asymmetric least squares)	Snelle techniek, eenvoudige Python-code	Kans op overfitting, inschatten van parameters
IAsLS (Improved Asymmetric least squares)	Lage RMSE-waarde, rekening met eerste afgeleide	Moeilijke Python-implementatie, inschatten van parameters

Tabel 2.9: Overzicht van de basislijncorrecties en van de belangrijkste voor- en nadelen (vervolg) (55–60)

Basislijncorrectie	Voordelen	Nadelen
GaCspline (Genetic algorithm cubic spline)	Detectie en verwijdering van storingen, zelfstandige inschatting van de basislijn	Ingewikkelde Python-code door genetisch karakter
ErPLS (Extended range penalized least squares)	Automatische inschatting, kleine basislijnverschuivingen	Python-implementatie vereist uitbreiding van het spectrum
AirPLS (Adaptive iteratively reweighted penalized least squares)	Eenvoudige Python-implementatie	Inschatting van coëfficiënten

2.3.1.1 Manuele basislijncorrectie

Na het opnemen van een FTIR-spectrum kan een manuele basislijncorrectie worden doorgevoerd. Het menselijk oog is goed in het herkennen van patronen en inschatten van punten ter correctie van de spectra. Ook geeft een manuele correctie een goed inzicht in het spectrum. De meest prominente pieken worden het snelste opgemerkt en worden het nauwkeurigst gecorrigeerd. Naast het feit dat de reproduceerbaarheid van deze techniek laag is, immers schat een ander persoon de basislijn anders in, zijn er nog andere beperkingen verbonden aan deze techniek. (58)

Ten eerste is de snelheid waarmee de basislijn gecorrigeerd wordt laag. De inschatting duurt beduidend langer dan wanneer een algoritme de basislijn inschat. Ook wordt de ruis met een automatische basislijncorrectie in de meeste gevallen geëlimineerd, wat niet het geval is bij een manuele correctie. Daarnaast is het de bedoeling om een kwantificering uit te voeren met behulp van het IR-spectrum en uit onderzoek is gebleken dat een automatische basislijncorrectie aanleiding geeft tot een betere inschatting van de concentraties. Dit is wel onder de voorwaarde dat de luchtvochtigheid een constante factor is gedurende de opname. In het geval van een verschil in luchtvochtigheid tussenin verschillende opnamen, kan het zijn dat een manuele correctie betere inschattingen levert van de concentratie. Daarom zal in het praktische gedeelte van deze thesis, de manuele basislijncorrectie meegenomen worden in de vergelijking met automatische basislijncorrecties. (58)

2.3.1.2 Improved asymmetric least squares en asymmetric least squares

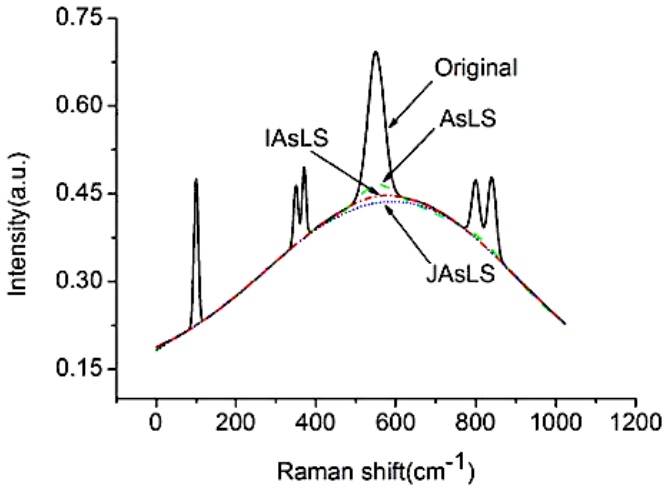
Twee methoden die geschikt zijn om aan basislijncorrectie te doen en overigens sterk verwant zijn met elkaar, zijn IAsLS en AsLS. Deze termen staan respectievelijk voor improved asymmetric least squares en asymmetric least squares. AsLS is gebaseerd op de Whittaker smoother en een differentiële wiskundige techniek. Deze smoother is een manier om het signaal gladder te maken zonder verlies aan relatieve gewichten. De differentiële techniek neemt de tweede afgeleide van de curve om de trends waar te nemen. (55,56)

In de praktijk wordt echter gezien dat louter werken met de tweede afgeleide verschillende nadelen heeft. Een nadeel is bijvoorbeeld dat het juiste verloop in de basislijn zit, maar dat deze niet volledig aansluit. Dit is te zien op Figuur 2.23 waarbij duidelijk wordt dat de AsLS wel degelijk de trend volgt maar niet de juiste hoogte inschat van de basislijn. Indien de eerste

afgeleide eveneens in acht wordt genomen, zal de basislijncorrectie beter aansluiten aan de werkelijke basislijn. Deze verbetering wordt afgekort als JAsLS, waarbij de ‘J’ verwijst naar Jiang. Dit is de naam van de onderzoeker die deze verbetering heeft ontdekt. (55,56)

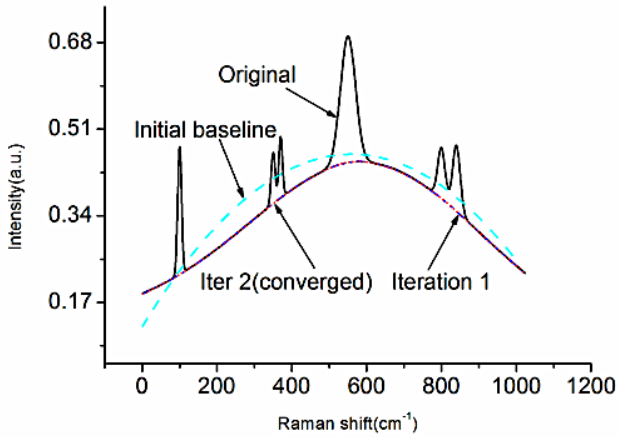
Een complete integratie van de eerste afgeleide in het benaderen van de basislijn, wordt gevonden bij de IAsLS-methode. De wiskunde hierachter is relatief ingewikkeld. De algoritmes zijn ook gebaseerd op de Whittaker smoother. Deze smoother is een verbetering van de reeds genoemde SG-filter. Waar de Whittaker smoother als voordeel heeft dat het een snelle en flexibele methode is, steekt het de SG-filter daardoor voorbij als performante techniek. De implementatie van dergelijke smoother kan gebeuren met Python-code. (55,56)

De rode draad in deze afleidingen is een iteratie waarbij gezocht wordt naar de best mogelijke fit. Waar de Whittaker smoother convergentie bereikt na vijf tot tien iteraties, kan de IAsLS reeds na twee iteraties convergentie bereiken. Uit het artikel *Baseline correction for Raman spectra using an improved asymmetric least squares method*, blijkt dat IAsLS het krachtigste is, gevolgd door respectievelijk JAsLS en AsLS. De RMSE-waarde is dan ook het grootste voor de AsLS-methode. De tijd die nodig is om de basislijn te berekenen, is in de drie gevallen steeds kleiner dan 1 seconde waarbij AsLS de snelste is. Zonder verder in te gaan op de wiskunde achter de constructie van de basislijn, toont Figuur 2.23 de resultaten van de verschillende AsLS-methode. (55,56)



Figuur 2.23: Vergelijking van de AsLS-, JAsLS- en IAsLS-methode (55)

Figuur 2.23 toont de best mogelijke iteratieve benaderingen van de basislijn met alle drie de AsLS-methoden. Daaruit blijkt dat IAsLS de basislijn accurater kan benaderen dan JAsLS en AsLS. Ook het verschil tussen AsLS en JAsLS is relatief groot. Waar AsLS de piek rond 600 cm^{-1} duidelijk hoger inschat dan JAsLS, blijkt dat JAsLS beter aansluit aan de werkelijke basislijn en aan IAsLS. AsLS zal in vergelijking met de andere methoden mogelijks aanleiding geven tot overfitting. De IAsLS-correctie is krachtig genoeg om reeds na twee iteraties de basislijn te benaderen. Dit wordt gedemonstreerd in Figuur 2.24. (55)



Figuur 2.24: Resultaat van de basislijnbenadering na 1 en 2 iteraties met de IAsLS-methode (55)

Aangezien het de bedoeling is om een basislijncorrectie uit te voeren op spectrale data via een Python-programma, is de implementatie van de algoritmen naar Python-code een belangrijk aspect. Het blijkt dat de IAsLS en JAsLS moeilijk te schrijven zijn voor een niet-getraind programmeur. Daarom zal in de praktische verwerking van de spectrale data gewerkt worden met de AsLS-techniek die wel relatief eenvoudig te implementeren is in Python.

2.3.1.3 Cubic spline basislijncorrectie

Genetic algorithm cubic spline baseline correction (GaCspline) maakt gebruik van een genetisch algoritme. Dit maakt dat golfgetallen die in de achtergrond voor storingen zorgen, waardoor de basislijn verschuift, gedetecteerd en verwijderd worden. Deze verwijdering is erop gebaseerd dat in een eerste stadium het algoritme op zoek gaat naar afwijkende golfgetallen. Dit zijn golfgetallen waarbij er een verschuiving is van de basislijn. Daarna worden deze punten afgetrokken van het spectrum waardoor een correctie voltrokken wordt. Het genetisch algoritme kan zelfstandig inschattingen maken en onvolkomenheden opmerken in het spectrum. Ook hier is er het probleem van de vertaling naar Python-code. Voor een niet-geoefend programmeur is de implementatie niet vanzelfsprekend. Daarom zal deze correctiemethode in het praktische gedeelte van deze thesis niet aan bod komen. (57)

2.3.1.4 Extended range penalized least squares

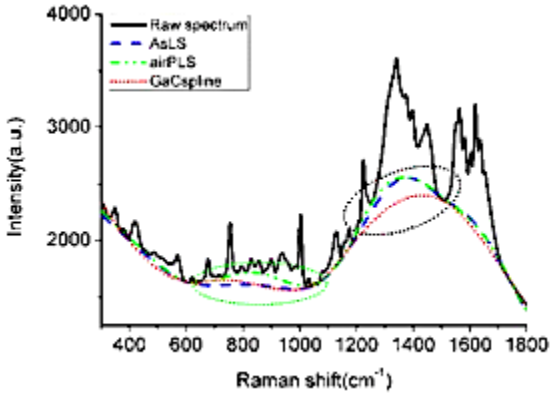
Deze methode is hoofdzakelijk gebaseerd op de least squares benadering. Extended range penalized least squares ofwel erPLS, maakt gebruik van een automatische inschatting van parameters. Er is hier sprake van een automatische basislijncorrectie. De erPLS-techniek is gebaseerd op de AsLS-techniek en kan aanzien worden als een verbetering ervan. Daar AsLS-methoden vooral performant zijn bij grote verschuivingen, zal de performantie dalen bij kleine basislijnverschuivingen. Dit heeft een invloed op de kwantitatieve verwerking. Een smoothing kan dit gedeeltelijk wegwerken maar er is daarvoor een extra bewerking nodig. De wiskundige techniek van ErPLS berust op een uitbreiding van het spectrum. Dit moet eveneens geprogrammeerd worden. Dit is te moeilijk om in een eerste aanleg te vertalen naar Python-code. Daarom wordt deze techniek enkel als theoretisch voorbeeld behandeld. (59)

2.3.1.5 Adaptive iteratively reweighted penalized least squares

Adaptive iteratively reweighted penalized least squares (AirPLS) is een variant van erPLS en gaat eveneens een basislijncorrectie doorvoeren op een ingevoerd IR-spectrum. Daarenboven wordt hierbij ook de Whittaker smoother gebruikt. De techniek berust op het veranderen van

de gewichten gekoppeld aan het spectrum. Dit is eveneens een vorm van smoothing. Een nadeel verbonden aan AirPLS is de inschatting van twee parameters, namelijk de asymmetrie-parameter en de smoothingcoëfficient. Dit kan deze methode omslachtiger maken dan bijvoorbeeld AsLS. De kleinste kwadraten methode wordt gebruikt om de iteratie door te voeren en te evalueren. Bij elke iteratie wordt een nieuw gewicht toegekend aan een vector die geassocieerd is aan het spectrum en wordt vergeleken ten opzichte van een vorig punt. Wanneer convergentie bereikt is en dus een minimale waarde bij de evaluatie gevonden is, wordt besloten dat de best mogelijke basislijn geconstrueerd is. (60)

Figuur 2.25 geeft in een groene stippellijn de basislijn weer die gevonden werd met AirPLS. Daarbij wordt opgemerkt dat de basislijn minder goed aansluit dan deze geconstrueerd met GaCspline (voorgesteld door de rode stippellijn) maar wel vergelijkbaar is met de basislijn die geconstrueerd is met AsLS (voorgesteld door de blauwe stippellijn). (60)



Figuur 2.25: Visuele weergave van de GaCspline-, AsLS- en AirPLS-techniek (respectievelijk in rood, blauw en groen aangeduid) (57)

Er kan opgemerkt worden dat wanneer de coëfficiënten geoptimaliseerd worden, de basislijn even goed benaderd wordt zoals met GaCspline. Dit is althans een van de besluiten die getrokken werd uit het artikel *Baseline correction using adaptive iteratively reweighted penalized least squares*. Er werd eveneens aangegeven dat deze techniek gekenmerkt wordt door een simpel doch efficiënt algoritme waarmee analytisch bruikbare resultaten worden bekomen. Daarenboven is dit algoritme sneller en flexibeler in vergelijking met GaCspline of AsLS. Ook kan door een eenvoudige aanpassing van de coëfficiënten een ander spectrum geoptimaliseerd worden. Echter dient wel eerst vastgesteld te worden wat de optimale waarde van deze coëfficiënten is door een trial-and-error operatie. (60)

Aangezien dat deze basislijncorrectie op een relatief eenvoudige manier in Python-code kan vertaald worden en vergelijkbaar is met AsLS, zal deze techniek in het praktische gedeelte van deze thesis onderzocht worden.

2.3.2 Tussentijdse conclusie omtrent kwantificeren met ATR-FTIR

Het gebruik van ATR-FTIR blijkt interessant te zijn op vele vlakken. De techniek is niet-destructief, snel en het gebruik van schadelijke solventen en chemicaliën wordt verminderd. De kwantificatie gebeurt echter niet zonder enige restrictie. Het gebruik van de wet van Lambert-Beer zal niet volstaan om een precieze bepaling van een concentratie te volbrengen door de veronderstellingen die hierbij al dan niet impliciet worden gemaakt. Daaronder valt de veronderstelling dat er over het hele concentratiebereik een lineair verband heerst tussen de absorbantiewaarden en de concentraties. Ook wordt de kwantificatie

bemoeilijkt door verschuivingen van pieken en storingen in het achtergrondsignaal. Daarnaast is er het belangrijke aspect van de piekselectie. Eén enkele piek kan onmogelijk dienen ter referentie. Een serie pieken daarentegen leent zich hier wel toe. Verscheidene methoden om de selectie te voltrekken zijn hiervoor geschikt. Onder andere PCA en PLS zijn hiervoor aangewezen. Om een gladder signaal en minder ruis te verkrijgen, kan gebruik gemaakt worden van een bandfilter. Een voorbeeld hiervan is de SG-filter. (34,42,45,46,48–50)

Uit de praktijk blijkt basislijnverschuiving en daarmee gepaard gaande reproduceerbaarheid een probleem te zijn. De basislijn is de referentie voor de piekhoogte en het is belangrijk dat deze accuraat wordt bepaald om de kwantificatie te voltrekken. Verschillende technieken kunnen hiervoor gebruikt worden. Uit de literatuur blijkt een iteratieve methode, toegepast door een algoritme het meest aangewezen. Daarvoor zal gebruik gemaakt worden van AsLS en AirPLS in het praktische gedeelte van deze thesis. (55–57,59,60)

2.3.3 Pre-processing van data

Om aan deep learning te doen, is pre-processing een cruciale stap. Vaak wordt deze om verschillende redenen gedaan. Een reden kan zijn om de grootte van de dataset te verkleinen en enkel te werken met relevante data. Het kan ook zijn dat er bepaalde delen van de dataset moeten gescheiden worden van de andere voor de verwerking. Bijvoorbeeld om aan patroonherkenning te doen, moeten uitschieters verwijderd worden. Pre-processingsmethoden behoren tot de tak van de chemometrie. (61)

Spectrale data bevatten vaak te veel meetpunten om verwerkt te worden door een deep learning (DL) algoritme. Daartoe moet ten eerste een selectie gemaakt worden van relevante meetpunten. Deze spectrale data leent zich uitstekend voor een chemometrische verwerking. Daarom wordt speciale aandacht gevestigd op multivariaatmethoden. (61)

In de volgende paragrafen zal dieper ingegaan worden op de techniek van data-scaling, PLS en PCA daar deze uit de literatuur, omtrent kwantitatieve verwerking, het meest interessant bleken te zijn. Ook standard normal variate (SNV) is een mogelijke pre-processingsmethode en wordt in wat volgt verder toegelicht. In Tabel 2.10 wordt een overzicht gegeven van de technieken die zullen worden besproken, alsook van de belangrijkste voor- en nadelen. (61)

Tabel 2.10: Overzicht van de pre-processingstechnieken en de daarbij horende belangrijkste voor- en nadelen (61–65)

Pre-processing	Voordelen	Nadelen
Data-scaling	Eenvoudig toe te passen	Mogelijk verlies aan informatie
PCA (Principale component analyse)	Datareductie: snellere en eenvoudigere verwerking voor het DL-model	Mogelijk verlies aan informatie, instellen van het aantal principale componenten
PLS (Partial least squares)	Datareductie: snellere en eenvoudigere verwerking voor het DL-model	Instellen van het aantal latente variabelen
SNV (Standard normal variate)	Eenvoudige implementatie in Python, eliminatie van ruis	Inschatten van parameters

2.3.3.1 Voorbehandeling van de data

Om data vlot te verwerken, dient deze eerst een zogenoemde voorbehandeling te ondergaan. Wanneer deep learning wordt gehanteerd, wordt dit aangeduid met de term data-scaling. Er zijn verscheidene manieren om deze voorbehandeling uit te voeren. De data zal steeds aanzien worden als een matrix met kolommen en rijen waarin de data geordend is. Deze matrix zal in deze thesis steeds worden voorgesteld door de notatie X. Deze matrix X heeft als dimensie *m* kolommen en *n* rijen. (61)

In Tabel 2.11 wordt een overzicht gegeven van de formules (Vergelijkingen 2.3 tot en met 2.5) en de verklaringen van de symbolen om de voorbehandelingsstappen te voltrekken. Daartoe behoren centering, standaardisatie en normalisatie. (61)

Tabel 2.11: Formules om data-scaling uit te voeren (Vergelijking 2.3 tot en met Vergelijking 2.5) (61)

Techniek	Formule	Betekenis
Centering	$x_{ij}' = x_{ij} - \overline{x_j}$ **(2.3)**	Met x_{ij}' het gecentreerde element, x_{ij} het oorspronkelijke matrix-element en $\overline{x_j}$ het kolomgemiddelde
Standaardisatie	$x_{ij}'' = \frac{x_{ij} - \overline{x_j}}{s_j}$ **(2.4)**	Met x_{ij}'' het gestandaardiseerde element, x_{ij} het oorspronkelijke matrix-element, $\overline{x_j}$ het kolomgemiddelde en s_j voor de standaarddeviatie van de kolom
Normalisatie	$x_{ij}''' = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}$ **(2.5)**	Met x_{ij}''' genormaliseerde element, x_{ij} het oorspronkelijke matrix-element, $\min(x_j)$ voor het kolomminimum en $\max(x_j)$ voor het kolommaximum

Centering

Wanneer aan centering wordt gedaan, wordt een gemiddelde genomen per kolom en dit afgetrokken van elk element uit de matrix X (Vergelijk 2.3 in Tabel 2.11). Dit wordt toegepast wanneer relatief gezien de meetpunten onderling dezelfde verschillen moeten behouden maar absoluut gezien moeten verkleinen qua waarde. Door te centreren rond het gemiddelde wordt dit voltrokken, waarbij het kolomgemiddelde het centrum vormt van deze spreiding. Waarden kleiner dan het gemiddelde worden zo negatief geschaald en waarden groter dan het gemiddelde krijgen een positieve schaling. (61)

In sommige situaties leidt het toepassen van centering tot een verlies aan informatie. (61)

Standaardisatie

Standaardisatie heeft een andere manier van werken dan centering. Daarbij wordt van elk matrixelement het kolomgemiddelde afgetrokken en de verhouding wordt vervolgens genomen ten opzichte van de standaarddeviatie van de kolom (Vergelijking 2.4 in Tabel 2.11). Dit wordt toegepast om de samenhang tussen de waarden van een dataset te vergroten. Door telkens te delen door de standaardafwijking, worden de meetpunten dichter bij elkaar

geschaald. Dit is interessant wanneer in de verwerking relatief weinig onderscheid mag gemaakt worden tussen de meetpunten onderling. (61)

Net zoals centering kan standaardisatie leiden tot een verlies aan informatie. Sommige statistische methoden vereisen echter dat dit gebeurt. (61)

Normalisatie

Tot slot kan ook normalisatie worden gebruikt. Deze heeft als resultaat dat alles geschaald wordt met behulp van het maximum en het minimum van de kolom (Vergelijk 2.5 in Tabel 2.11). Deze schaling heeft als grenzen 0 en 1. Eventueel kunnen de waarden nadien vermenigvuldigd worden met een vermenigvuldigingsfactor om de grenzen aan te passen. Normalisatie wordt gebruikt wanneer duplicaten uit een dataset moeten worden verwijderd. Ook wordt normalisatie gehanteerd wanneer een dataset moet worden ingedeeld in kleinere datasets met gelijkaardige kenmerken. (61,66)

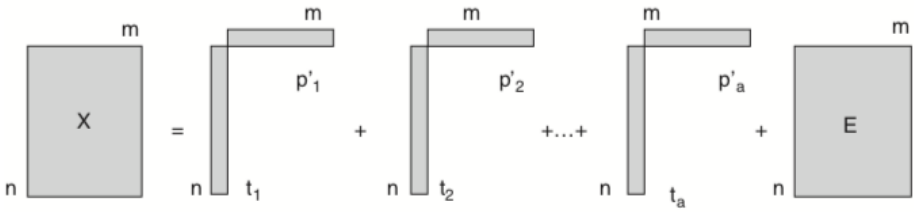
2.3.3.2 Multivariaatmethoden

Principale component analyse

Een welbekende multivariaatmethode is principale component analyse (PCA). Dit is een projectiemethode en berust erin om relevante informatie te selecteren uit een grote dataset en deze te benoemen als principale componenten (PC's). De eerste PC bevat meer informatie in termen van variantie dan de tweede PC, die op zijn beurt meer info bevat dan de derde PC, enzovoort. Dit gaat door tot alle informatie onderverdeeld is in PC's. (61)

Wanneer dit wiskundig bekeken wordt, is PCA een projectie van een matrix X naar een nieuwe matrix met een kleinere dimensie. Er wordt daarbij dus aan datareductie gedaan. De matrix X kan bekeken worden als een m-dimensionale ruimte met n punten. PCA wordt typisch gebruikt voor unsupervised learning en voor patroonherkenning. (61)

Figuur 2.26 toont hoe de PCA-techniek de matrix X opdeelt in verschillende PC's. P'_1 is de eerste PC en deze bevat de meeste variantie en informatie. Voor de andere PC's tot en met P'_a geldt hetzelfde waarbij opeenvolgend steeds minder variabiliteit in de PC zit. De matrix E bevat de rest van de waarden die niet onderverdeeld zijn in vorige PC's en wordt de error-matrix genoemd. (61)



Figuur 2.26: Werkingsprincipe van de PCA-techniek (61)

Partial least squares

Partial least squares projection of latent structures is de volledige benaming van de afkorting PLS. De term PLS-R wordt gehanteerd wanneer er met behulp van PLS aan regressie wordt gedaan (vandaar 'R'). Deze methode wordt gebruikt om aan voorspellingen te doen en wordt bovendien ingedeeld in het veld van de supervised learning. Er wordt getracht een relatie te vinden tussen de response variabelen (bijvoorbeeld de concentratie) en de predictoren (bijvoorbeeld de piekhoogte). PLS werkt des te beter naarmate er veel data voor handen is en wanneer er een grote mate van correlatie is tussen de predictoren. (61)

Hetzelfde gedachtegoed zoals aangetroffen bij PCA wordt hier gehanteerd. Namelijk de originele informatie van de matrix X wordt geprojecteerd op een kleiner aantal latente variabelen die zo veel mogelijk informatie dragen. De eerste variabele bevat opnieuw meer informatie dan de volgende, enzovoort. Latent verwijst naar het feit dat deze variabelen uit een wiskundig model volgen. Om uiteindelijk een voorspelling te kunnen doen, moet een verband gezocht worden tussen de afhankelijke variabelen en de predictoren. Dit gaat des te makkelijker naarmate er minder latente variabelen zijn of nog anders gesteld, naarmate de informatie meer geconcentreerd is. (61)

Uit het artikel *Principle Component Analysis and Partial Least Squares: Two Dimension Reduction Techniques for Regression*, wordt geconcludeerd dat PLS efficiënter is dan PCA wanneer de afhankelijke variabele het belangrijkste is. Zo zal in een FTIR-spectrum de afhankelijke variabele de absorbantiewaarde zijn daar deze afhangt van de concentratie. De concentratie is de onafhankelijke variabele. In deze thesis is de concentratie de belangrijkste parameter en dus niet de afhankelijke variabele. (61,64,65)

In het kader van de thesis, zal PLS niet worden toegepast bij de verwerking van de spectrale data. Voor PLS wordt namelijk een algoritme opgebouwd dat supervised learning vereist terwijl het in deze thesis de bedoeling is om aan unsupervised learning te doen. Daarnaast is zoals net aangehaald, in dit onderzoek vooral de onafhankelijke variabele het belangrijkste (namelijk de concentratie). Uit de literatuur blijken PCA en PLS dan even efficiënt te zijn waardoor er evenveel voordeel gehaald wordt met PCA dan met PLS. (65)

Standard normal variate

Standard normal variate ofwel SNV, is eveneens een pre-processingsmethode. Deze methode wordt vaak gebruikt in combinatie met spectrale data. De techniek is een soort van normalisatie waarbij er een zogenoemde standaardscore voor elk meetpunt wordt berekend. Wanneer SNV gebruikt wordt als pre-processingsmethode, zal de standaarddeviatie verkleinen. Hierdoor worden het signaal en de basislijn bevrijd van ongewenste variaties. Daardoor kan een kwantificatie met grotere nauwkeurigheid gebeuren. (63)

Er zijn verschillende soorten SNV-technieken, namelijk dynamisch gelokaliseerde piek SNV, piek SNV en partiële piek SNV. In elke variant is SNV een hulpmiddel door de simpliciteit van het principe en de betrekking van het hele spectrum in een correctie van een individueel signaal. Dit kan verklaard worden door de kijken naar Vergelijking 2.6: de standaarddeviatie en het gemiddelde worden immers betrokken bij iedere correctie. (62,63)

$$I_{SNV}(\lambda) = \frac{|I_{Ruw}(\lambda) - I_{BL}(\lambda)|}{\sigma} * \mu \qquad \textbf{(2.6)}$$

Waarbij: $I_{SNV}(\lambda)$ de intensiteit van het signaal in functie van de golflengte na SNV-correctie (-),

$I_{Ruw}(\lambda)$ de intensiteit van het oorspronkelijk signaal in functie van de golflengte (-),

$I_{BL}(\lambda)$ de intensiteit van de basislijn in functie van de golflengte (-),

λ de golflengte (cm),

μ het gemiddelde van de netto-intensiteiten (-) en

σ de standaarddeviatie van de netto-intensiteiten (-)

Er wordt opgemerkt dat met betrekking tot Vergelijking 2.6, de netto-intensiteiten staan voor de uitkomst van de teller en dus voor het (absolute) verschil van $I_{Ruw}(\lambda)$ en $I_{BL}(\lambda)$. (63)

Bij dynamisch gelokaliseerde piek SNV wordt enkel gewerkt met gedeelten van het spectrum (standaarddeviatie en gemiddelde worden eveneens over dit gedeelte genomen). Bij piek SNV en partiële piek SNV wordt gekeken naar respectievelijk de golflengten waarbij een piekmaximum voorkomt of naar de golflengten op halve hoogte van de piek. (62,63)

Uit onderzoek blijkt dat alle varianten van de SNV-techniek de variatie reduceerden en dat partiële piek SNV de meest aangewezen methode was om de standaarddeviatie te verkleinen. De ruis werd eveneens verkleind. Als dit specifiek de bedoeling is, wordt er het beste gewerkt met het gehele spectrum in plaats van met delen van het spectrum. Er zijn bij de SNV-techniek verschillende parameters die moeten worden ingesteld. Bijvoorbeeld bij dynamisch gelokaliseerde piek SNV moet het golflengtegebied ingesteld worden. (62,63)

In deze thesis zal deze pre-processingsmethode worden gehanteerd en vergeleken met de PCA-techniek. Wanneer SNV wordt toegepast, zal eveneens een standaardisatie worden geïmplementeerd. Door de standaardisatie worden de gewichten van de respectievelijke pieken gereduceerd. Hierdoor gaan de pieken van de stalen onderling relatief gezien gelijk behandeld worden. Daarenboven zal het algoritme dat geschreven wordt, toepassing hebben op piek-SNV. De reden hiervoor is de simpliciteit van de implementatie in een Python-programma alsook omdat de kwantitatieve verwerking steunt op de absorbantiewaarden van de piekmaxima. Wanneer een grafische vergelijking wordt gedaan tussen een spectrum die behandeld is met een SNV pre-processing en een spectrum die zonder die pre-processing, is op het eerste gezicht geen verschil waar te nemen. De verandering van de absorbantiewaarden door de SNV pre-processing zijn daarvoor niet groot genoeg. (67)

2.4 Deep learning en neurale netwerken

De verwerking van data wordt heden ten dage bemoeilijkt door de steeds groeiende omvang en toenemende complexiteit van de datasets. Dit fenomeen wordt aangeduid met de term ‘big data’. Het blijkt wel dat deze big data een fenomeen is dat de economie doet pieken. Nieuwe arbeids- en afzetmarkten worden gecreëerd alsook nieuwe soorten economieën, waaronder robotica, worden verder ontwikkeld. Maar deze big data maakt dat manuele verwerking steeds arbeidsintensiever en ingewikkelder wordt. Dit heeft tot gevolg dat neurale netwerken, die een toepassing zijn van artificiële intelligentie (AI), steeds meer aan belang winnen. Dergelijke neurale netwerken lenen zich goed tot het verwerken van grote datasets en tot simulatie. Deze laatste eigenschap is belangrijk in het kader van deze thesis, daar het inschatten van concentraties onderzocht wordt. Het inschatten gebeurt overigens sneller met een neuraal netwerk dan dat dit manueel zou gebeuren. (68–70)

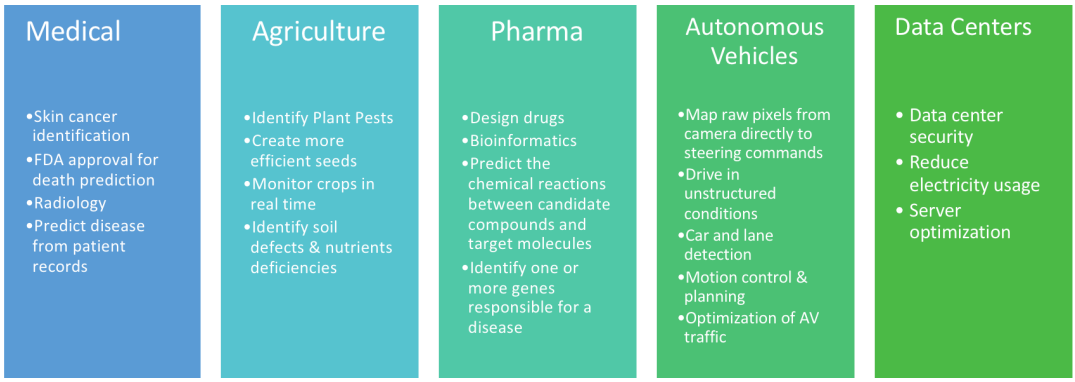
Hoewel de neurale netwerken snel een grote hoeveelheid aan data kunnen verwerken, gaan te grote datasets leiden tot een overmaat aan input. Dergelijke situaties dienen vermeden te worden aangezien de snelheid en de kracht van het netwerk daardoor afnemen. Door gebruik te maken van geschikte pre-processingsmethoden zal er een selectie worden gemaakt van relevante data. Dit maakt machine learning (ML) en in het bijzonder deep learning (DL) mogelijk. Het kan ook de bedoeling zijn om onderliggende verbanden op te merken die op het eerste gezicht niet zichtbaar zijn, althans niet voor een menselijke programmeur. Een ML- of DL-algoritme kan dit wel opmerken. Dit wordt ‘data mining’ genoemd. (68–70)

Er wordt in wat volgt ingegaan op de verschillende types neurale netwerken, de betekenis en het verschil in DL en ML, alsook op de achtergrond van de technieken.

2.4.1 Artificiële intelligentie

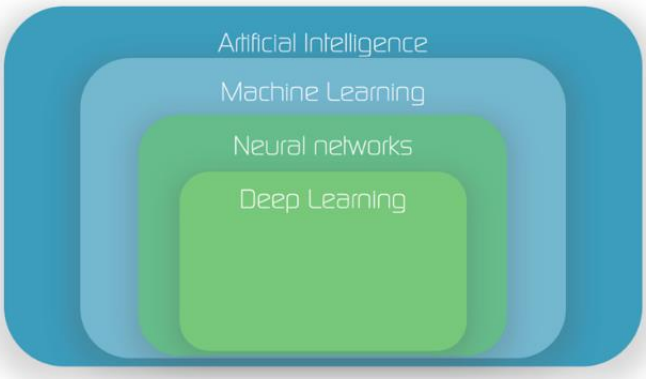
Artificiële intelligentie (AI) is alom vertegenwoordigd in de menselijke leefwereld. Van een mobiele telefoon tot een zelfrijdende auto; AI is geïntegreerd in het dagelijkse leven. AI wordt gedefinieerd als de simulatie van menselijk denken en intelligentie door een computersysteem. De doelstellingen van AI kunnen uiteenlopend zijn: menselijk denken nabootsen, menselijke acties inschatten, probleemoplossend denken, … Het komt erop neer dat de menselijke karakteristieken worden geïmiteerd door een computeralgoritme. Heden ten dage wordt de doelstelling nog scherper. Er wordt getracht het menselijk denken en ageren te overstijgen. Naast het feit dat dit een ethische discussie kan opwerpen, wordt er in deze thesis louter objectief naar AI gekeken. (71)

AI gaat verder dan het standaardbeeld van een robot, veelal worden de taken uitgevoerd door een getraind algoritme. Naargelang de manier van trainen, wordt er gesproken van machine learning (ML) en deep learning (DL). Een ML-algoritme wordt getraind door een mens maar zal zelfstandig leren uit deze training. Bij DL daarentegen, leert het algoritme uit grote datasets zonder enige menselijke tussenkomst. Wat er geleerd wordt, zal gestockeerd worden in zogenoemde neurale netwerken. Figuur 2.27 geeft een globaal overzicht van de toepassingen van AI in verschillende takken van de industrie. (71)



Figuur 2.27: Overzicht van de toepassingen van AI in verschillende industriële takken (72)

Het wordt al snel duidelijk dat de mogelijkheden van AI bijna onbeperkt zijn. Hoe AI, ML, DL en neurale netwerken exact met elkaar verweven zijn, wordt gegeven in Figuur 2.28. Hierin wordt getoond dat AI het overkoepelend orgaan is van ML, DL en neurale netwerken. Ook geeft Figuur 2.28 weer dat DL eigenlijk een toepassing is van de tak van ML en dat neurale netwerken worden opgebouwd door DL. (69,71,72)



Figuur 2.28: Weergave van de verhouding van artificiële intelligentie, machine learning, deep learning en neurale netwerken tegenover elkaar (73)

Hoe ML geëvolueerd is uit AI wordt uitgelegd aan de hand van een standaard toepassing van AI, namelijk een schaakprogramma. Oorspronkelijk werd het schaak-algoritme geprogrammeerd via een grote set aan regels. Er heerste een overtuiging dat wanneer de set groot genoeg was, de menselijke acties in alle omstandigheden zouden worden nagebootst. Echter bleken schaakzetten die niet voorgeprogrammeerd waren, een manier om het algoritme te verslaan. Snel werd duidelijk dat alles voorprogrammeren onmogelijk zou worden. Het algoritme diende met andere woorden zelf te kunnen simuleren. Het idee van een zelfstandig lerend algoritme was daarmee ontstaan. Dit algoritme zou niet voorgeprogrammeerd worden maar getraind worden. Hierop wordt dieper ingegaan in de volgende paragrafen. (69)

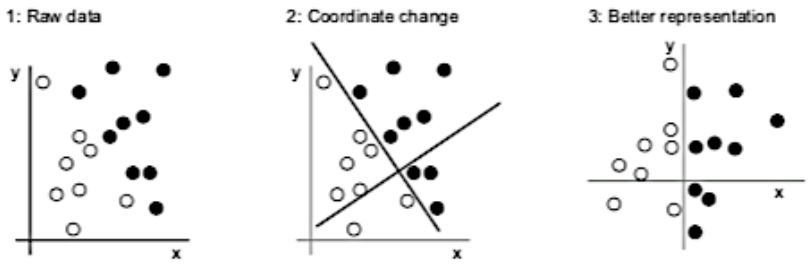
2.4.2 Machine learning en deep learning

Dagelijks komen mensen met AI in contact zonder het te beseffen. Een voorbeeld uit het dagelijkse leven is bijvoorbeeld hoe Netflix een nieuwe serie voorstelt op basis van de vorig bekeken series. Er zijn dagelijks nieuwe ontwikkelingen die ervoor zorgen dat AI meer kan en dat overigens steeds sneller kan. (69)

2.4.2.1 Het leermechanisme van machine learning en deep learning

Eerst is het belangrijk om te kijken waarvoor 'learning' precies staat. De term doet vermoeden dat er iets aangeleerd wordt. In het boek *Deep Learning with Python*, wordt dit leermechanisme verduidelijkt. Er zijn om te beginnen drie verschillende factoren nodig om het leren mogelijk te maken: input van datapunten, voorbeelden van verwachte output en een boolean-operatie. Deze laatste heeft als doeleinde om in te schatten of het algoritme een juiste waarde heeft gegenereerd. Een boolean is een operator die aangeeft of iets juist al dan niet fout is, de zogenoemde 'True' en 'False' in courante programmeertalen. Input wordt door middel van een bepaalde transformatie omgezet in output. Hoe die omzetting gebeurt, wordt aangeleerd door reeds gekende output te voeden aan het algoritme. Hiervan is bekend dat deze juist is en deze wordt dan als een referentie genomen. De input dient gecodeerd te worden in een fase die de encoding-fase wordt genoemd en toont hoe de input naar output moet worden omgezet. (69)

Algemeen kan gesteld worden dat ML- en bij uitbreiding DL-algoritmen, gebaseerd zijn op een gepaste transformatie te vinden tussen input en output opdat er een voorspelling kan gedaan worden. Een meer algemeen voorbeeld wordt gegeven in Figuur 2.29. Hierbij wordt het principe van classificatie gedemonstreerd. Classificatie gaat de datapunten opdelen in gedefinieerde klassen.



Figuur 2.29: Binair classificatie-leersysteem toegepast op zwarte en witte stippen (69)

Op Figuur 2.29 wordt de input ('1: Raw data') ingedeeld in zwarte en witte stippen zodanig dat deze twee aparte klassen vormen. Dit is met andere woorden een binair probleem. Hierbij wordt de eenvoudige classificatieregel (en tevens de encoding) vooropgesteld: Als de stip wit is, wordt deze geplaatst in de zone $x < 0$. Het omgekeerde geldt voor de zwarte stippen. De transformatie wordt centraal weergegeven in Figuur 2.29, namelijk de 'coordinate change' ofwel coördinaatverandering en wordt intern door het algoritme uitgevoerd. Als output wordt het rechtse coördinatenstelsel gegeven waarbij de classificatieregel toegepast werd. Hierbij worden duidelijk de twee klassen apart waargenomen. (69)

2.4.2.2 Verschillende types van leersystemen

Afhankelijk van de manier van programmeren zijn verschillende leersystemen een optie. Naargelang dat de training al dan niet plaatsvindt onder leiding van een persoon, wordt er gesproken van: supervised, semi-supervised, unsupervised en reinforcement learning. Ook wordt er een verschil gemaakt tussen detecteren van patronen of vergelijken ten opzichte van gekende datapunten. Verder is er ook een onderscheid tussen online en batch learning. Tabel 2.12 geeft weer waarvoor deze termen staan. (68)

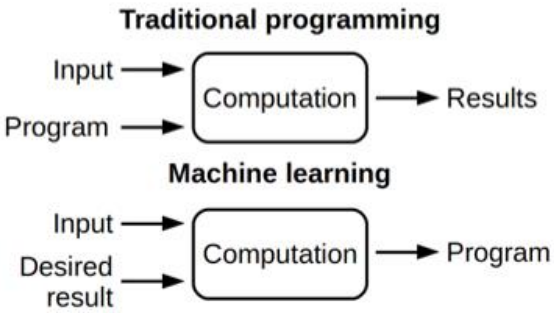
Tabel 2.12: Verschillende types van leersystemen (68)

Onderverdeling	Type	Betekenis
Hoe wordt er getraind?	Supervised	De oplossingen worden meegegeven als input en een programmeur waakt over het correct toepassen.
	Unsupervised	Er worden geen oplossingen meegegeven met de input en het algoritme ageert volledig zelfstandig.
	Semi-supervised	Slechts een gedeelte van de oplossingen wordt meegegeven als input.
	Reinforcement learning (robotica)	Dit is een leersysteem waarbij er bonus- en strafpunten worden toegekend. Op het einde wordt een rapport opgesteld.
Hoe wordt een voorspelling gemaakt?	Model-based (patroonherkenning)	Hierbij wordt er getracht een model te ontwikkelen of een patroon te herkennen in de dataset. Het model wordt daarna toegepast op een nieuw datapunt.
	Instance-based	Het model leert op basis van gelijkheid. Er wordt vergeleken met punten die reeds in het systeem zitten. Bij similariteit worden punten naast elkaar gezet. Hierbij wordt een soort 'van-buiten-leren' toegepast.
Hoe wordt de data gevoed?	Online learning	Er wordt getraind met incrementele datasets. Er wordt telkens data toegevoegd om het algoritme te trainen zodanig dat het nieuwe inzichten verwerft. Een volledig getraind algoritme kan op deze manier opnieuw in de trainingsfase gezet worden. Deze manier van leren is interessant om grote datasets te verwerken aangezien met dit soort leersysteem overbelasting vermeden wordt.
	Batch learning	Het systeem wordt getraind met alle beschikbare data tegelijkertijd. Hier zijn duidelijk twee fasen te herkennen: training en applicatie. Eens het algoritme getraind is, zal het algoritme de opgedragen taak uitvoeren zonder dat het nadien nog getraind wordt. Wanneer er nieuwe input-data beschikbaar wordt als training, dient een nieuwe versie gemaakt te worden.

In deze thesis worden de algoritmen geprogrammeerd als een unsupervised algoritme waarbij gelabelde input gevoed wordt. Er wordt aan model-based, batch learning gedaan. In paragraaf 3 omtrent Materialen en Methoden wordt hier dieper op ingegaan.

2.4.2.3 Klassiek programmeren versus machine learning

Om het verschil tussen machine learning (ML) en deep learning (DL) te verduidelijken, wordt er eerst gekeken naar de klassieke manier van programmeren. Deze wijze van programmeren wordt vergelijken met ML. Dit wordt weergegeven in Figuur 2.30.

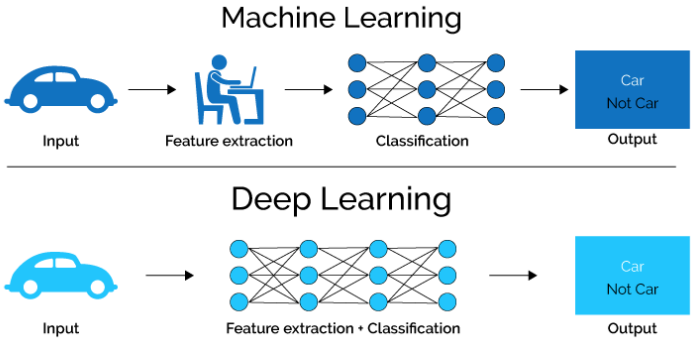


Figuur 2.30: Verschil tussen klassiek programmeren en machine learning (74)

Uit Figuur 2.30 volgt het denkpatroon van ML (en bij uitbreiding voor DL). Bij ML zal het antwoord dienen ter vorming van regels (desired results op Figuur 2.30). Bij klassiek programmeren zorgt het programma voor resultaten. Dit is dus het tegenovergestelde dan bij ML. Door een ML-benadering zal het algoritme sterker worden en kunnen anticiperen op nieuwe input. Het leerproces van een ML-systeem dient echter gestuurd te worden door de mens. (69)

2.4.2.4 Machine learning versus deep learning

Machine learning verschilt van deep learning op vlak van de manier waarop het algoritme geprogrammeerd wordt. ML-algoritmen zijn geprogrammeerd om specifiek een bepaalde taak uit te voeren. Hierbij wordt er data verwerkt en uit deze verwerking volgt een bepaalde actie of beslissing. Dit is een statische manier van programmeren. Bij een incorrecte output zal de programmeur zelf aangeven wat er fout was. De volgende keer zal het algoritme deze fout eruit halen. Echter zal hetzelfde proces zich voordoen bij een andere onbekende fout. Om verder het verschil tussen ML en DL duidelijk te maken, wordt Figuur 2.31 gebruikt. (69,72,75)



Figuur 2.31: Machine learning tegenover deep learning (72)

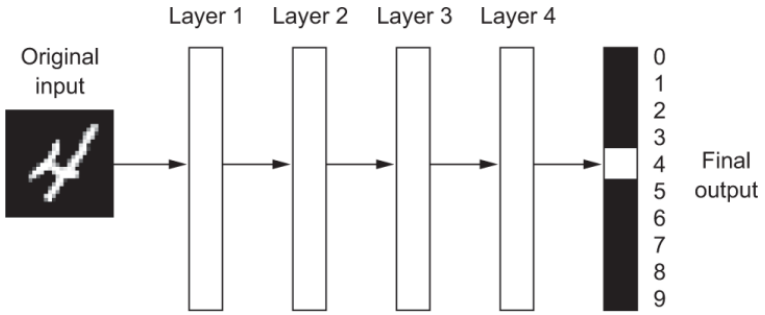
Zoals aangegeven wordt in Figuur 2.31, is er menselijke tussenkomst nodig wanneer ML gebruikt wordt. Deze tussenkomst vertaalt zich in het voorprogrammeren van een hypothese set. Deze set stelt een ML-algoritme in staat om te denken. Door de statische manier van programmeren, is een ML-logaritme niet altijd in staat om een onderliggend verband te vinden. Hierdoor wordt het leermechanisme bemoeilijkt. Door reeds aan te geven wat een set van mogelijk juiste waarden zijn (de hypothese set), kan het verband wel gevonden worden. (69)

Bijvoorbeeld een programma waarbij landen ingedeeld worden in de verschillende werelddelen. Een mogelijke hypothese set is: *België, Frankrijk en Duitsland behoren tot Europa* en *Duitsland, Luxemburg en Nederland zijn buurlanden van België*. Wanneer het algoritme het land Luxemburg tegenkomt in de dataset en weet dat het een buurland is van België, deelt het algoritme dit in onder het werelddeel Europa. Er zullen ook landen zijn die niet in de hypothese set zitten, bijvoorbeeld Spanje, waarbij het algoritme dan een berekende

gok maakt om het land in te delen in een werelddeel. Ook voor landen die op de grens liggen van twee werelddelen, bijvoorbeeld Turkije (ligt op de grens tussen Europa en Azië), wordt gegokt tot welk werelddeel dit behoort. Hierbij zal een programmeur moeten aangeven wat het juiste antwoord is. (69)

Bij deep learning is het de bedoeling dat de algoritmen zodanig ontwikkeld zijn opdat een matrix ontstaat dat op zijn beurt bestaat uit verschillende gekoppelde lagen. Uit elke koppeling dient het algoritme sterker en slimmer te worden. Zodoende wordt de output van een laag, de input van een volgende en ontstaan lange ketens aan informatie. Dit vormt de basis van zogenoemde artificiële neurale netwerken (ANNs). (69,72,75)

Er zijn steeds twee buitenste lagen (input en output). De andere lagen zijn zogenoemde hidden layers of verborgen lagen. Het opgebouwde neurale netwerk stelt een DL-algoritme in staat om zelfstandig te oordelen of een output al dan niet correct is. Dit vormt een significant verschil tussen DL en ML. Wanneer in het algemeen een deep learning proces voorgesteld wordt, zal altijd een neuraal netwerk mee afgebeeld worden. Zonder verder op het type neuraal netwerk in te gaan, geeft Figuur 2.32 een deep learning proces weer. De output is louter een verzameling van cijfers vertrekkende van een afbeelding als input. (69,72,75)

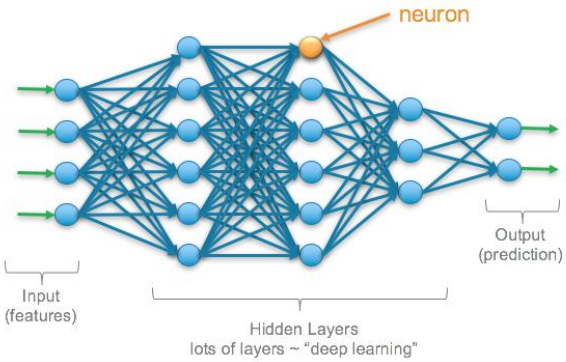


Figuur 2.32: Eenvoudige weergave van een deep learning proces (69)

2.4.3 Neurale netwerken

Neurale netwerken zijn inherente toepassingen van het veld van de artificiële intelligentie en zijn de basis van zowel deep als machine learning. Dit werd reeds aangegeven in Figuur 2.28. Een DL-model wordt opgebouwd om een bepaald patroon te herkennen, bepaalde output te genereren of een actie te kunnen ondernemen op basis van reeds verworven kennis. Deze kennis wordt opgeslagen in een netwerk dat bestaat uit verschillende lagen. Het geheel van lagen is met elkaar verbonden. De term neuraal netwerk is ontleend aan de biologie, waarbij het ‘neuraal’ verwijst naar neuronen die zich in de hersenen bevinden. (68)

Er wordt een onderscheid gemaakt tussen artificiële en gesimuleerde neurale netwerken (respectievelijk ANNs en SNNs). Het zijn de ANNs die het hart vormen van deep learning. Belangrijk is dat input hierbij dient om het netwerk slimmer te maken. De input-laag zal altijd bereikbaar zijn. Bereikbaarheid verwijst naar het feit dat sommige lagen dat niet meer zijn. Deze worden dan verborgen lagen genoemd. Deze bevinden zich tussen de input- en de output-laag. Figuur 2.33 geeft hiervan een visuele weergave. (76)



Figuur 2.33: Schematische voorstelling van een neuraal netwerk (77)

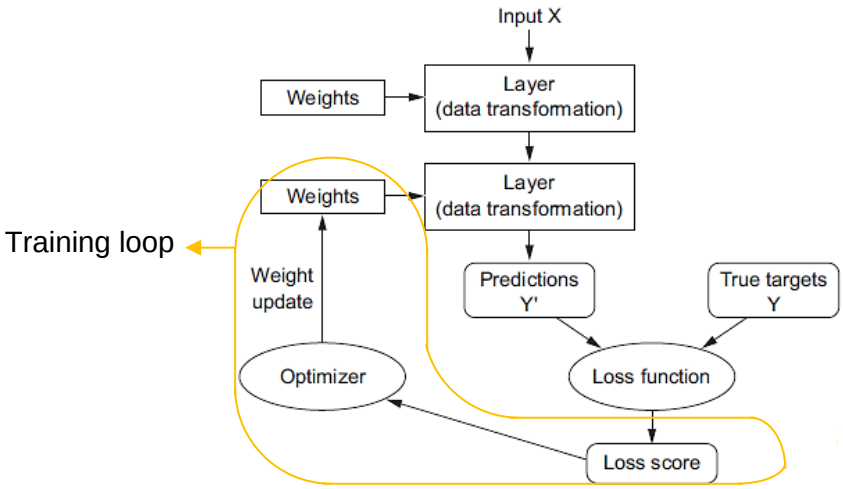
Wanneer ingezoomd wordt op de werking van dit netwerk kan het snel ingewikkeld worden. Het komt erop neer dat elk neuron, ook wel een node genoemd, een bepaald geassocieerd gewicht en drempelwaarde heeft. Zodra de input ervoor zorgt dat een drempelwaarde wordt overschreden, wordt de node geactiveerd. Activatie betekent dat het signaal aan de volgende reeks noden wordt doorgegeven. Is er dan opnieuw een node waarvan de drempelwaarde overschreden wordt, dan zal ook hier het signaal worden doorgegeven. Dit gaat door tot de output layer bereikt is (en resulteert in een output signaal) of dit stopt wanneer er geen enkele node het signaal meer doorgeeft. (76)

Dit neuraal netwerk wordt intelligent, sterk en robuust indien dit veelvuldig getraind wordt door middel van de verwerking van nieuwe data. Een voorbeeld van een neuraal netwerk dat (bijna oneindig) veel getraind wordt, is de Google-zoekmachine. Elke nieuwe zoekopdracht maakt dit systeem robuuster. (76)

2.4.3.1 Werking van een deep learning door opbouw van een neuraal netwerk

Het deep learning proces staat of valt met het neuraal netwerk dat opgebouwd wordt. Een robuust neuraal netwerk laat het DL-proces toe om accurate voorspellingen te doen. De opbouw van het netwerk door en voor de werking van het DL-proces, wordt uitgewerkt met behulp van de schema's uit het boek *Deep Learning with Python*. (69)

Een eerste verborgen laag wordt gevormd door de gewichten van de input te nemen. Elke input wordt immers gezien als een vector met een bepaald gewicht (bijvoorbeeld de scalaire grootte van de vector). Deze gewichten worden opgeslagen in deze eerste laag. Figuur 2.34 geeft dit weer.



Figuur 2.34: Opbouw van een neuraal netwerk en aanduiding van de training loop (oranje) (78)

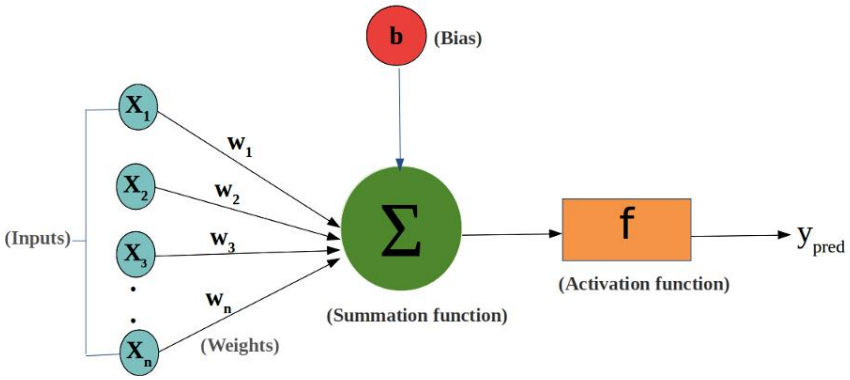
Op Figuur 2.34 is ook is te zien hoe de gewichten verder worden getransformeerd naar een nieuwe set gewichten in de daaropvolgende laag. Beter is het om te zeggen dat de vectoren getransformeerd worden en de daarbij horende gewichten veranderen. Zodoende wordt een output Y' gevormd die een set van gemaakte voorspellingen is. Ook dit wordt weergegeven in Figuur 2.34. De keten van input naar output wordt een perceptron genoemd. (69,79)

Deze eerste benadering van een neuraal netwerk toont een heel erg belangrijk aspect niet, namelijk de feedback-koppeling. Zoals Figuur 2.30 deed vermoeden, moet het algoritme gevoed worden met een vorm van feedback, namelijk de 'gewenste uitkomst'. De gewenste uitkomst kan vrij letterlijk de gewenste uitkomst zijn, maar veel vaker komt het bij deep learning voor dat een verliesfunctie als feedback wordt gegeven. De verliesfunctie is niets meer dan de uitkomst van een validatie van de output Y' tegenover de werkelijke waarden Y. Het verschil ertussen wordt het verlies genoemd en wordt gedefinieerd in de verliesfunctie. De 'Loss score' op Figuur 2.34, is niets anders dan het verlies berekend tussen Y en Y'. (69)

Dit verlies wordt als feedbacksignaal teruggegeven aan de laag die instaat voor de transformatie naar de voorspelling Y'. De lus die zo gevormd wordt, krijgt de benaming 'training loop' (in het oranje aangeduid op Figuur 2.34). Dit verlies wordt eerst doorgegeven aan een 'Weight optimizer' vooraleer aan het netwerk teruggegeven te worden. Zo zal de inschatting van de gewichten accurater zijn. Een optimizer is een algoritme dat zorgt voor de correctie van kritische parameters waaronder de leersnelheid. Dit wordt verder uitgediept in paragraaf 2.4.3.3. De terugkoppeling via de training loop, wordt aangeduid met de term backpropagation of terugpropagatie. Hoe accurater de gewichten ingeschat worden, des te meer de voorspelde waarden zullen aanleunen bij de reële waarden. Ook dit is te zien op Figuur 2.34. (69,80)

2.4.3.2 Beknopte wiskundige achtergrond

De opbouw van een deep learning neuraal netwerk gaat gepaard met complexe wiskunde. Figuur 2.34 gaf het principe weer van DL zonder in te gaan op de wiskundige achtergrond. Figuur 2.35 doet dit op een relatief eenvoudige manier en toont hoe er wiskundig wordt omgegaan met de inputwaarden.

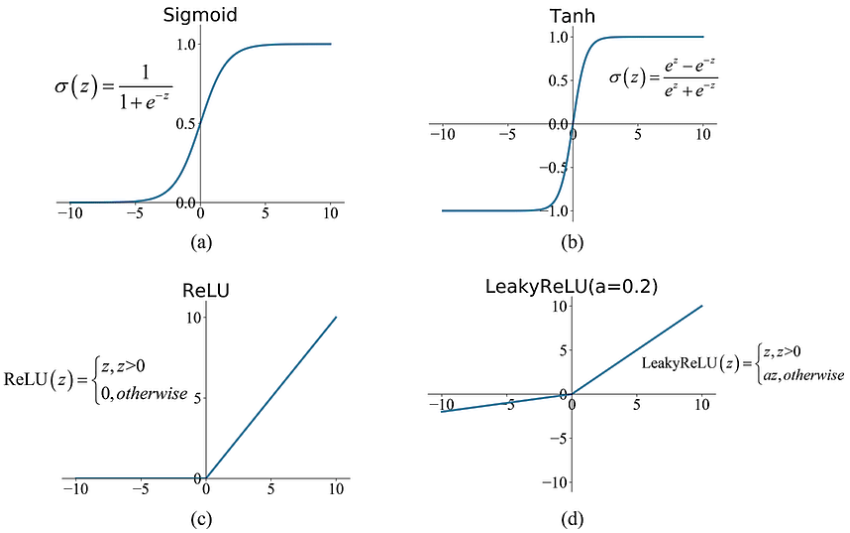


Figuur 2.35: Wiskundige benadering van een voorwaarts gepropageerd perceptron (81)

Zoals weergegeven op Figuur 2.35 wordt de input X gevoed aan een transformatie-laag. Deze transformatie-laag neemt de som van de producten van de input en het geassocieerde gewicht. Dit gewicht geeft ook aan wat het relatieve belang is van een bepaalde inputwaarde. Verder wordt er getracht een systeem te ontwikkelen dat niet louter met lineaire verbanden kan werken maar ook met niet-lineaire systemen. Daartoe wordt een niet-lineaire activatiefunctie f ingebouwd. (69,80)

De activatiefunctie f kan naargelang de input variëren. Bijvoorbeeld indien een verzameling van x-waarden enkel positieve waarden kent, zal de activatiefunctie hierop betrekking hebben en bijvoorbeeld enkel rekening houden met het positieve deel van de x-as. Wanneer er dan een set negatieve waarden als input wordt toegevoegd, zal deze activatiefunctie niet langer het gewenste effect hebben. Deze zou in deze situatie deels moeten verschoven worden naar het negatieve deel van de x-as. Daartoe dient de bias b. Onafhankelijk van de input x-waarden zal de activatiefunctie het gewenste effect hebben door een mogelijke verschuiving naar 'links' of 'rechts'. Deze geeft met andere woorden een extra dimensie aan de input. Ook zorgt deze ervoor dat nulwaarden als input steeds een output krijgen. (69,80)

Mogelijke activeringsfuncties worden gegeven in Figuur 2.36. Er zal vaak gewerkt worden met de Sigmoïd-functie, daar voldoende variantie wordt behouden binnen wel gedefinieerde grenzen van y (namelijk 0 en 1). Ook de rectified linear activation function ofwel ReLU behoudt de oorspronkelijk aanwezige variantie van de input. Daarenboven worden dan geen negatieve waarden van y getolereerd en is er een lineair verband nodig. Eens de activatiefunctie gedefinieerd is, kan de output worden gegenereerd. (69,80)

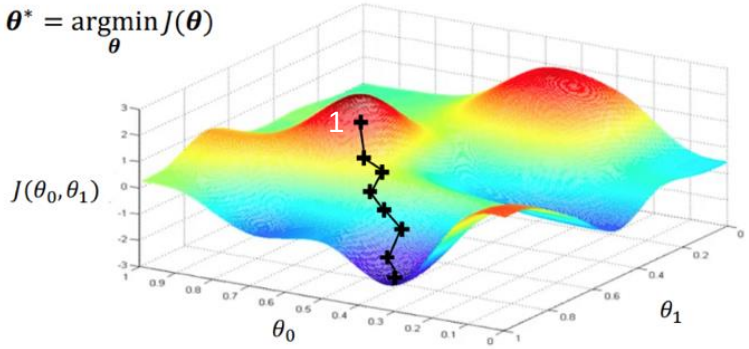


Figuur 2.36: Mogelijke activatiefuncties: a) Sigmoid, b) Tanh, c) ReLU en d) LeakyReLu (82)

2.4.3.3 Trainen van neurale netwerken, gradiënt daling en terugpropagatie

Het trainen en het leren verschillen op zich niet veel van elkaar qua aanpak. Het verschil zit louter in de doelstelling. Bij het leermechanisme is het de bedoeling dat het algoritme op eigen basis data verwerkt. Bij het trainen van een neuraal netwerk, wordt data ingevoerd door de programmeur met als doel het algoritme iets aan te leren. De klemtoon ligt niet op de zelfstandigheid maar eerder op het verwerven van kennis. Het trainen van een neuraal netwerk is niets anders dan het algoritme leren hoe de gewichten van de input moeten worden ingeschat en vertaald naar output. Dit vergt een grote hoeveelheid aan data. (80)

In een eerste benadering kan er getracht worden het verlies te minimaliseren. Het verlies is ook in deze context het verschil tussen de ingeschatte waarde en de werkelijke waarde. Dit verlies wordt gedefinieerd door de verliesfunctie. Als deze minimaal is, zal de inschatting maximaal zijn. Dit wordt voorgesteld in Figuur 2.37. In de linkse bovenhoek wordt de geminimaliseerde functie voorgesteld door θ^*. Stel dat er slechts twee input waarden zijn, θ_0 en θ_1, dan kan de verliesfunctie grafisch geplot worden zoals voorgesteld in Figuur 2.37. (80)



Figuur 2.37: Verliesoptimalisatie door minimalisatie van de verliesfunctie (80)

Een eerste inschatting van de gewichten leidt tot een eerste punt aangeduid met een '1' op Figuur 2.37. Vervolgens wordt de gradiënt berekend die hoort bij deze eerste inschatting (op dit eerste punt) waarbij rekening gehouden wordt met de gewichten. Deze gradiënt geeft informatie over het stijgen en dalen van de functie. Is de helling negatief, dan kan er een kleiner verlies gevonden worden. Het tweede punt wordt dan gekozen zodanig dat de dalende richting aangehouden wordt. Dit wordt doorgevoerd tot een minimum in verlieswaarde bereikt is. Er wordt iteratief naar het minimum gezocht. De convergentie wordt bekomen door Vergelijking 2.7 toe te passen. De voorstelling in Figuur 2.37 is in veel gevallen te eenvoudig aangezien het aantal inputwaarden vaak meer dan twee bedraagt. Daardoor wordt de meerdimensionale voorstelling al snel veel ingewikkelder. (80)

$$\theta - \eta \frac{\partial j(\theta)}{\partial \theta} \rightarrow \theta \qquad \textbf{(2.7)}$$

Met θ het gewicht,

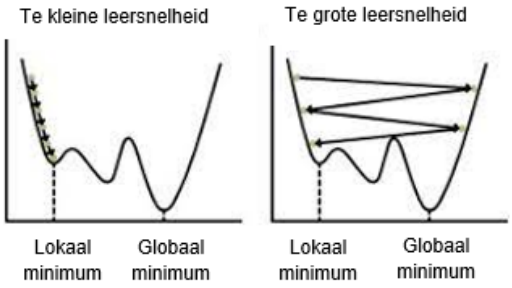
η de leersnelheid en

$j(\theta)$ de verliesfunctie

Door telkens het gewicht aan te passen naar de juiste richting, wordt geconvergeerd naar een minimum. De factor η is nieuw geïntroduceerd in Vergelijking 2.7 en is een maat voor de snelheid van een stap in dalende zin te nemen. Deze 'snelheid' staat voor de afstand tussen de twee punten in Figuur 2.37 en wordt de leersnelheid genoemd. Deze manier van minimaliseren wordt aangeduid met de term stochastic gradient descent ofwel stochastische gradiënt daling (SGD). SGD is een optimizer die gebruikt wordt bij de opbouw van neurale netwerken. (80,83)

De leersnelheid, of de snelheid waarmee stappen worden genomen richting het minimum, is een parameter die een belangrijke invloed heeft op het leerproces. Hoe groter de leersnelheid wordt ingesteld, hoe groter de stappen zijn en omgekeerd. Het proces verliest dan aan nauwkeurigheid, doch zijn er minder stappen en dus tijd nodig. In deze situatie zal het systeem zich snel aanpassen aan nieuwe data maar wordt de oudere data eveneens sneller vergeten. Onderliggende verbanden kunnen zo over het hoofd worden gezien door de grote fluctuaties. Wordt de leersnelheid te klein ingeschat, wordt een lokaal minima aanzien als het minimum van de verliesfunctie. Terdege is het systeem robuuster maar tegelijkertijd ook gevoeliger aan ruis. De te kleine parameter zorgt eveneens voor een vertraging van het proces. (68,79)

Figuur 2.38 geeft een weergave van de gevolgen van een fout inschatting van de leersnelheid. (84)



Figuur 2.38: Effect van een te grote en te kleine inschatting van de leersnelheid op de verliesfunctie (84)

Het instellen van een te grote leersnelheid zal resulteren in een kleiner aantal training loops in tegenstelling tot een kleine leersnelheid. Deze training loop werd voorgesteld in Figuur 2.34. De minimalisatie van de verliesscore gaat immers sneller waardoor er minder training loops nodig zijn. Echter kan het zijn dat de nauwkeurigheid daardoor kleiner is, wat zich vertaalt in een groter verlies (Loss score op Figuur 2.34). De leersnelheid zou kunnen begrepen worden als: “Hoe snel moet het verlies geminimaliseerd worden?”.

Een goed alternatief is een zelfregulerende leersnelheid. Hierbij wordt gekeken of de gradiënt klein of groot is en naargelang deze uitkomst, wordt de leersnelheid vergroot of verkleind. De optimizer RMSPROP is hiervan een voorbeeld. (85)

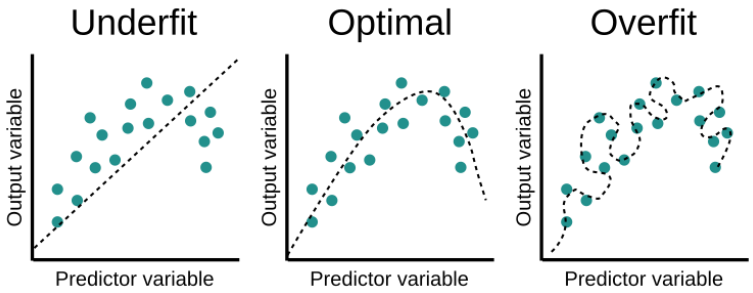
Aangezien er verschillende lagen aanwezig zijn in een deep neuraal netwerk en gewichten moeten geüpdatet worden voor elke laag bij een terugkoppeling, dient een mechanisme voorzien te worden om vanuit de achterste lagen terug te koppelen naar de eerste lagen. Deze manier van werken wordt backpropagation of terugpropagatie genoemd en is een toepassing van de kettingregel voor afgeleiden. (79)

Zoals reeds vermeld, dient een systeem getraind en getest te worden. Daartoe is zowel trainings- als testdata nodig. Trainingsdata wordt gebruikt om het algoritme te trainen. Om te kijken of het algoritme goed presteert, wordt testdata gebruikt. De training zal altijd intensiever zijn dan het testen en daarvoor is meer data nodig is. Typisch zal 80 % van de data gebruikt worden voor de training en de overige 20 % voor het testen van het model. Nadat het model getraind en getest is, kan het bijkomend gevalideerd worden (in de validatiefase). Daartoe wordt nieuwe data gebruikt die het model nooit gezien of verwerkt heeft. Zo kan getest worden of het model al dan niet in staat is om nieuwe data te verwerken. (68)

2.4.3.4 Probleem van underfitting en overfitting

Het algoritme is in het geval van underfitting niet in staat om onderliggende verbanden bloot te leggen en nieuwe data te verwerken. Het probleem van underfitting wordt duidelijk wanneer het gesimuleerde model, de complexiteit van het probleem niet aankan. Bij overfitting doet zich net het omgekeerde voor. Het algoritme neemt dan een trend waar en gaat deze trend inbouwen in de lagen waardoor de data gememoriseerd wordt. De complexiteit van het algoritme is in dit geval te hoog. (68,86)

Om underfitting en overfitting te vermijden, is een middenweg nodig waarbij het algoritme complex genoeg is om onderliggende structuren en verbanden op te merken, zonder trend daadwerkelijk op te slaan. Bij overfitting dient de complexiteit van het model verlaagd te worden, terwijl bij underfitting net het tegenovergestelde noodzakelijk is. Wat in beide gevallen helpt, is de training uitbreiden door een grotere trainingset te voorzien. Figuur 2.39 geeft deze fenomenen weer. (68,86)



Figuur 2.39: Het probleem van underfitting en overfitting (87)

Regularisatie wordt toegepast bij het training van een neuraal netwerk. Hierbij wordt er 'te veel' data genomen dan nodig is om een geoptimaliseerd model te maken. Dit aangezien het vergroten van de hoeveelheid data een goede manier is om underfitting en overfitting te voorkomen. Door het toepassen van regularisatie, wordt getracht underfitting en overfitting te vermijden. De mate waarin regularisatie wordt toegepast, wordt ingesteld door middel van de grootte van de hyperparameters. (68,86,88)

Hyperparameters zijn parameters van het leermechanisme op zich en niet van het model. Daaronder worden bijvoorbeeld het aantal epochs en de leersnelheid verstaan. Een epoch is een iteratie over de hele dataset. Dit komt neer op het doorlopen van één training loop zoals getoond op Figuur 2.34. De bepaling van de grootte van de hyperparameters is een trial-and-error operatie. Deze worden bepaald alvorens het leerproces te starten. Wordt een hyperparameter te groot ingesteld, zal er geen overfitting kunnen plaatsvinden en omgekeerd. Het aantal verborgen eenheden en het aantal lagen worden daarentegen beschouwd als 'gewone' parameters aangezien deze eigen zijn aan het model en niet aan het leermechanisme. (68,86,88)

Ook de batch-grootte (batch-size) is een hyperparameter. Dit staat voor het aantal datawaarden die simultaan worden verwerkt door het algoritme. Door de batch-grootte groter te nemen dan 1, wordt de dataset opgedeeld in groepen. Een batch-grootte van bijvoorbeeld 5 staat voor: "Vijf opeenvolgende waarden worden verwerkt in eenzelfde iteratie". (68,86,88)

Er zijn nog bijkomende manieren om underfitting en overfitting te vermijden. Onder meer drop-out is een veel gebruikte techniek. Daarbij wordt een bepaald percentage (vaak 50 %) van de noden uitgeschakeld. Dit getal wordt de drop-out-parameter genoemd. Daardoor wordt overfitting vermeden en wordt het model robuuster. Bij een volgende iteratie wordt een andere 50 % uitgeschakeld at random, zodanig dat er geen noden een voorkeur krijgen. (68,86)

2.4.3.5 Soorten neurale netwerken

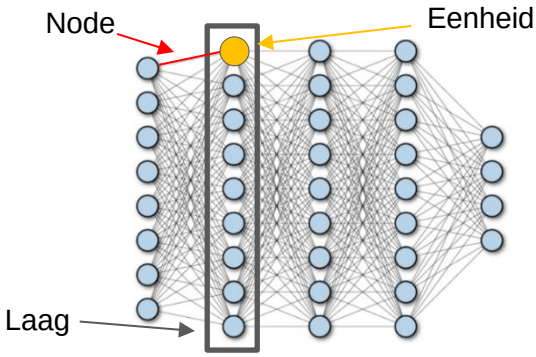
Er bestaan verschillende soorten neurale netwerken naargelang de verbindingen en de operaties die tussen de lagen worden uitgevoerd. Er wordt kort belicht welke soorten netwerken er bestaan. Tabel 2.13 geeft een overzicht van de mogelijke neurale netwerken en de daaraan verbonden toepassingen.

Tabel 2.13: Overzicht van de verschillende neurale netwerken en toepassingen (89–91)

Neuraal netwerk	Toepassing
Fully connected	Breed toepassingsveld
Recurrent	Afbeeldingsherkenning en -simulatie
Convolutional	Tekstherkenning en -voorspelling

Fully connected network

Een eerste soort neuraal netwerk is een fully connected network. Hierbij wordt elke laag verbonden met elke andere laag en elke eenheid met elke eenheid. Er ontstaat op dergelijke wijze een complexe kaart met veel noden. Figuur 2.40 geeft een voorbeeld van dergelijk netwerk. Hierop wordt voor alle duidelijkheid een eenheid, een node en een laag aangeduid. (90)



Figuur 2.40: Een fully connected network (90)

Het voordeel van een fully connected netwerk is dat er geen aannames dienen gemaakt te worden, immers is alle informatie bereikbaar door de connecties tussen de lagen onderling. Dit soort netwerken zijn het meest algemeen en kunnen voor verscheidene taken ingezet worden. Dit voordeel is meteen ook het nadeel van dergelijk netwerk; er is namelijk geen gespecialiseerde eigenschap. Soms is het gemakkelijker om te werken met een netwerk die specifiek gemaakt is voor een bepaalde taak. Wanneer dit niet nodig is, wordt het best gekozen voor een fully connected network. Vanuit dat perspectief zal ook in het kader van deze thesis gewerkt worden met een fully connected neural network. (90)

Convolutional neural network

Een convolutional neural network (CNN) is een geheel ander soort neuraal netwerk dan een fully connected network. Dit soort netwerken wordt gebruikt bij beeldverwerking. De input van een CNN is steeds een afbeelding. Deze netwerken worden minder gebruikt voor spectrale dataverwerking en daarom wordt ook niet verder ingegaan de implementatie van dergelijk netwerk. (89)

Recurrent neural network

Een ander belangrijk type neuraal netwerk is het recurrent neural network (RNN). Deze netwerken worden ontworpen om patronen in sequenties te ontdekken. Ook de voorspellende kracht van deze netwerken is groot. Deze netwerken zijn gespecialiseerd in tekstherkenning, bijvoorbeeld woordvoorspelling op basis van een getypte zin. Oorspronkelijk was de Long Short Term Memory (LSTM) de belangrijkste ontwikkeling binnen dit domein. Daarbij wordt een stuk van de zin onthouden (korte termijn geheugen) of een stuk in een van de vorige zinnen (lange termijn geheugen). Een recurrent neural network, wordt zoals de naam doet vermoeden, gekenmerkt door een steeds terugkerend deel. (91)

3 MATERIALEN EN METHODE

De verwerking van FTIR-datasets ter inschatting van een concentratie van een ternair mengsel van stabilisatoren, zal in deze thesis gebeuren door middel van een neuraal netwerk. Er zal daarbij aan deep learning worden gedaan. Het netwerk wordt opgebouwd en geoptimaliseerd zodanig dat de inschattingen van de concentraties steeds nauwkeuriger worden. Daartoe worden verschillende stalen met variërende concentraties aan glycerol, sorbitol en MPG aangemaakt en geanalyseerd door middel van FTIR. Er worden twee soorten deep learning netwerken geconstrueerd: een neuraal netwerk om binaire mengsels van stabilisatoren te verwerken alsook een netwerk voor ternaire mengsels te behandelen.

3.1 Materialen

3.1.1 Binaire en ternaire mengsels van stabilisatoren

In dit onderzoek wordt er gebruik gemaakt van twee types van binaire mengsels namelijk MPG en glycerol alsook MPG en sorbitol. De glycerol-standaard is 99,71 % zuiver en de MPG-standaard is voor 99,59 % zuiver. De sorbitol-standaard bevat 70 % sorbitol en 30 % water. Wanneer in deze thesis over sorbitol gesproken wordt, zal steeds op deze standaard als grondstof gedoeld worden. Deze twee binaire mengsels worden in verschillende verhoudingen aangemaakt. De concentraties van de stalen die gebruikt zijn bij de training, het testen en de validatie worden verder toegelicht in paragraaf 4.1.

Om deze binaire stalen aan te maken, wordt een analytische balans van het type AB204-S (Mettler Toledo) gebruikt. Deze stalen worden vervolgens geanalyseerd met behulp van een FTIR-spectrofotometer van het type IRAffinity-1S (Shimadzu). Zodoende worden er na de analyse van de stalen verschillende datasets verkregen. Deze datasets zullen dienen ter training, ter testen en ter validatie van het opgebouwde binaire model. Dit model wordt geconstrueerd door aan deep learning te doen, waarbij een neuraal netwerk ontwikkeld wordt. Dit neuraal netwerk vormt het hart van het binaire model.

Na de experimenten met de binaire modellen, worden ook ternaire modellen opgebouwd en gevalideerd. Daartoe worden ternaire mengsels aangemaakt van de stabilisatoren glycerol, MPG en sorbitol. Deze worden bereid met behulp van dezelfde analytische balans AB204-S (Mettler Toledo). De analyse van de stalen wordt ook hier voltrokken met de FTIR-spectrofotometer van het type IRAffinity-1S (Shimadzu).

Bij de ternaire experimenten wordt er eerst een Design of Experiments (DoE) opgesteld opdat de verhoudingen van de stabilisatoren zouden leiden tot een zo robuust mogelijk model. Dit werd gedaan door middel van Design-Expert-software. De bedoeling van deze DoE is om een idee te krijgen welke concentraties inzake de ternaire mengsels uit het DoE-model volgen. Hierbij worden dan ook louter verder gewerkt met deze ternaire concentraties.

In een later experiment zullen deze aangemaakte binaire en ternaire stalen aangevuld worden met RO-water (Reversed Osmosis) zodanig dat de concentratie aan water ongeveer 50 % bedraagt. Dit gebeurt ook via gewichtsmetingen op de analytische balans AB204-S

(Mettler Toledo). De overstap naar een ternair mengsel van stabilisatoren in de aanwezigheid van water, is een stap in de opbouw naar de uiteindelijke analyse van enzympreparaten. Deze bevatten namelijk vaak een mengsel van polyolen en water. Tot deze polyolen, wat staat voor een verbinding met verschillende hydroxylgroepen, behoren ook sorbitol, glycerine en MPG. Het toevoegen van water aan deze stalen wordt dus gedaan opdat de commercieel verkrijgbare enzympreparaten zouden worden benaderd qua samenstelling. Deze stalen worden vervolgens geanalyseerd met de spectrofotometer IRAffinity-1S (Shimadzu). Op basis van deze data wordt eveneens een neuraal netwerk opgebouwd.

Om het watergehalte van sommige stalen en enzympreparaten te bepalen, wordt gebruik gemaakt van een Karl Fischer titratie. Dit wordt uitgevoerd met een 915 KF Ti-Touch (Metrohm) voorzien van een 6.0338.100 dubbel platina elektrode (Metrohm). Als reagentia wordt gebruik gemaakt van Methanol Quick (VWR) en Reagent CombiNORM 5 (VWR).

3.1.2 Neuraal netwerk

De inschattingen van de concentraties in een binair of ternair mengsel van stabilisatoren, wordt uitgevoerd door middel van deep learning en de opbouw van een neuraal netwerk. Het binair mengsel bestaat uit MPG en glycerine of MPG en sorbitol. In het ternaire mengsel van stabilisatoren zijn zowel MPG, glycerine als sorbitol aanwezig. Er wordt in het kader van deze thesis gekozen voor een fully connected netwerk, zodanig dat alle informatie in elke laag steeds beschikbaar is voor de volgende lagen. Deze optie is geschikt daar de inschatting van een concentratie geen specifiek neuraal netwerk vereist. Dit netwerk wordt opgebouwd voor zowel twee (binair mengsel), drie (ternair mengsel) als vier (ternair mengsel en water) componenten. De architectuur van dergelijke netwerken is echter verschillend voor de inschatting van twee componenten in vergelijking met die voor meerdere componenten. Daar wordt in paragraaf 3.2.5 dieper op ingegaan. (90)

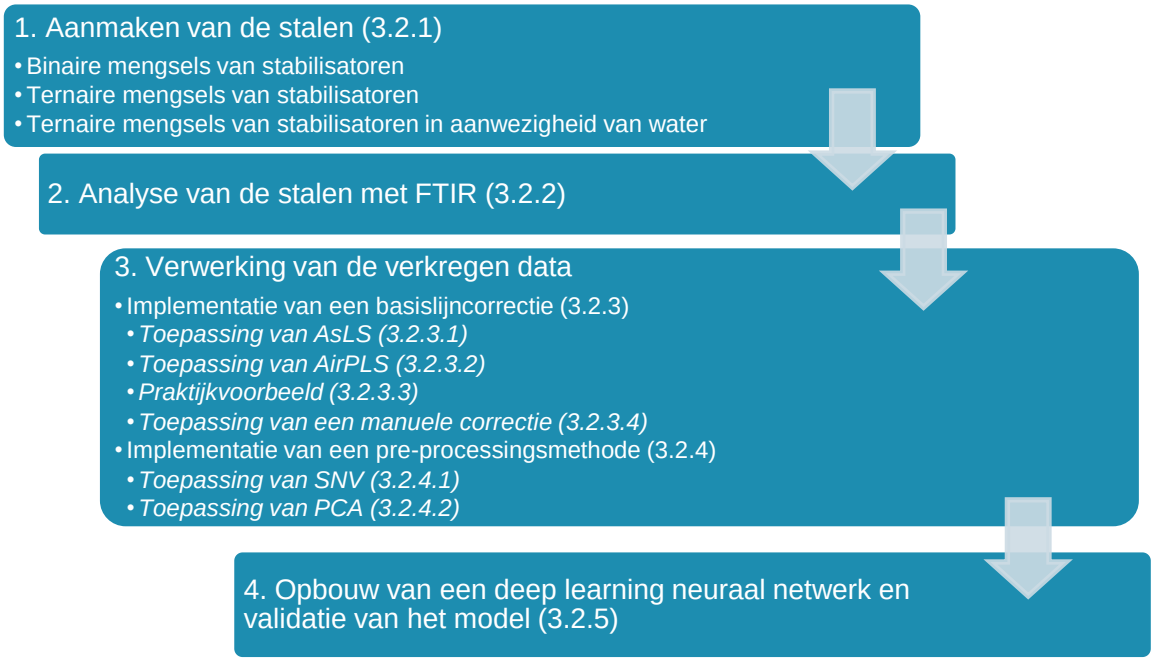
De netwerken worden geprogrammeerd in de programmeertaal Python. Dit aangezien er tal van pakketten zijn die de opbouw van het neuraal netwerk gemakkelijker maken en omdat er daarenboven een handleiding gebruikt wordt waarin ook voor Python geopteerd wordt. Deze handleiding is het boek *Deep learning with Python*. (69)

Het Pandas-, NumPy-, Matplot-, Tensor- en TensorFlow-pakket worden bij de opbouw van een neuraal netwerk quasi altijd gebruikt. (92–95)

- Pandas maakt het gebruik van dataframes mogelijk. Dataframes structureren data in een matrixvorm. De gestructureerde datasets kunnen vervolgens doorgegeven worden aan het neuraal netwerk. Daarenboven zal daardoor datamanipulatie (bijvoorbeeld het wisselen van rijen, transponeren, ...) mogelijk worden. (95)
- Het Numpy-pakket maakt wiskundige operaties en rij-operaties mogelijk. Dit kan van pas komen wanneer een rij van een dataframe moet bewerkt worden. (92)
- Matplot wordt gebruikt voor de visuele weergave van data en dus voor het construeren van grafieken en figuren. (93)
- Het Tensor-pakket is eveneens belangrijk bij het programmeren van een neuraal netwerk. Tensor en TensorFlow zijn beide deelpakketen van het overkoepelende pakket Keras. Dit pakket bevat specifieke onderdelen om een neuraal netwerk op te bouwen zoals ‘Layers’, ‘Model’ en ‘Optimizer’. (94)

3.2 Methode

Deze paragraaf omtrent de gehanteerde methode bevat een opeenvolging van stappen nodig om de kwantificatie van een ternair mengsel van stabilisatoren te volbrengen. Om het overzicht te bewaren, worden de aspecten die besproken worden in dit deelhoofdstuk weergegeven in Figuur 3.1.



Figuur 3.1: Schematische weergave van de gevolgde methode in de experimentenreeksen

3.2.1 Bereiding van de stalen

De bereiding van de binaire en ternaire stalen gebeurt door middel van standaarden van sorbitol, MPG en glycerol. De binaire en ternaire stalen worden in verschillende concentraties aangemaakt tussen 0 % en 100 % van de verschillende stabilisatoren. Om deze concentraties te bekomen, worden gewichtsmetingen uitgevoerd op de analytische balans AB204-S (Mettler Toledo). De stabilisatoren worden afgewogen in een glazen container. Het aantal binaire stalen dat op dergelijke wijze aangemaakt werden, bedraagt 153 voor het MPG-glycerol mengsel en 155 voor het MPG-sorbitol mengsel. De concentraties van deze stalen worden besproken in paragraaf 4.1.

Om ervoor te zorgen dat er een meer gelijke spreiding is tussen de ternaire stalen onderling, wordt ervoor geopteerd een DoE op te stellen door het gebruik van Design-Expert-software. Er werd gekozen voor de optie ‘Chemical mixtures’. Door deze DoE zullen de concentraties van de stalen zodanig gekozen zijn dat een optimaal model wordt geconstrueerd. Er wordt louter verder gewerkt met de ternaire stalen die dit model aanreikt.

In een daarop volgend experiment zal het concentratiebereik voor de eventueel nieuwe ternaire stalen afhangen van een optimalisatie die wordt doorgevoerd. Bijvoorbeeld als de regio tussen 40 % en 60 % sorbitol slecht scoort, zullen de stalen in een volgend experiment concentraties hebben die voornamelijk op deze regio betrekking hebben. Het totaal aantal ternair gemaakte stalen bedraagt 193. De concentraties van deze stalen worden per experiment besproken in paragraaf 4.2.

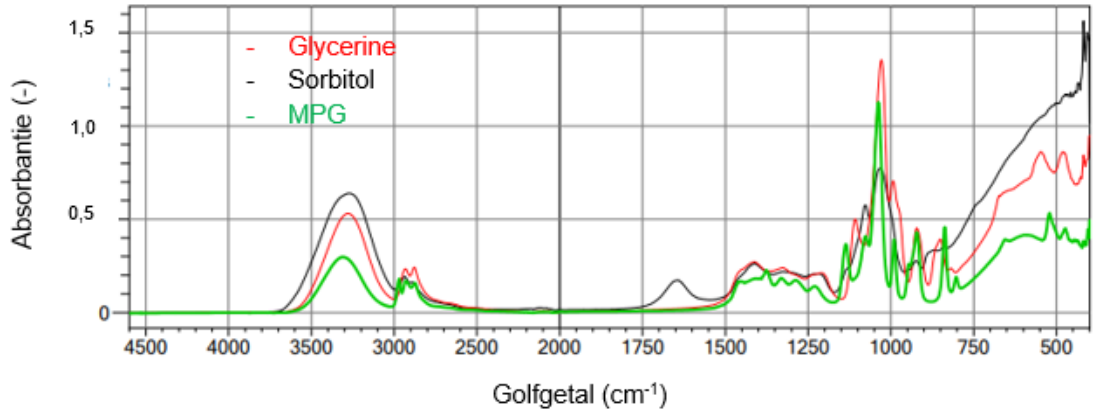
Vervolgens worden de binaire en ternaire stalen in een later experiment aangevuld met RO-water tot wanneer de concentratie van het water ongeveer 50 % bedraagt. Dit wordt gedaan opdat de commercieel verkrijgbare enzympreparaten zouden worden benaderd qua samenstelling. Op dergelijke wijze werden er zo 271 stalen bereid.

3.2.2 Uitvoeren van een FTIR-analyse

Eens de stalen bereid zijn, worden ze geanalyseerd op de FTIR-spectrofotometer van het type IRAffinity-1S (Shimadzu). De analyse zelf wordt voltrokken door een druppel te laten vallen op het kristal waarna de FTIR-opname gebeurt. De IR-spectrofotometer wordt zodanig ingesteld dat er 45 $^{scans}/_{min}$ worden voltrokken. De resolutie bedraagt in elke opname steeds 4 cm^{-1} en er wordt gemeten in het golfgebied van 400 cm^{-1} tot 4600 cm^{-1}.

Het spectrum wordt verwerkt door middel van de LabSolutions IR software. Deze software laat toe verschillende spectra over elkaar te leggen en zorgt ook voor de vertaling naar een txt-bestand. Dit bestand wordt vervolgens omgezet naar een xlsx-formaat (Excel-bestand). De spectra worden in deze vorm gevoed aan het neuraal netwerk

Figuur 3.2 toont een overzicht van de spectra van sorbitol, glycerine en MPG. Op de individuele spectra werd reeds in paragraaf 2.2.3 ingegaan. De bedoeling van Figuur 3.2 is om de verschillen te tonen in de spectra van de drie componenten.

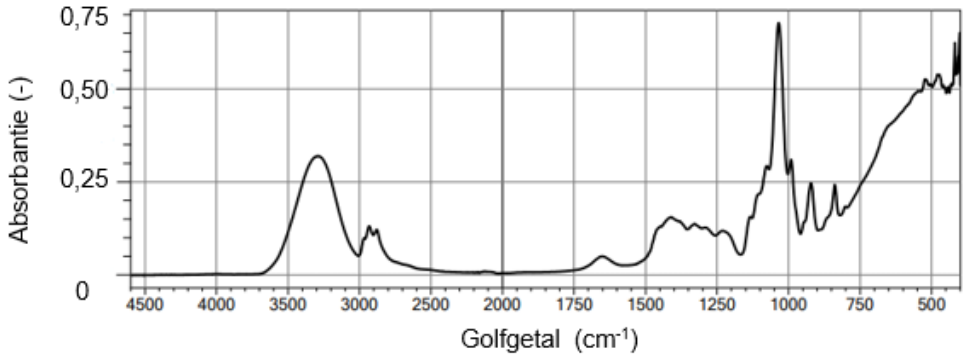


Figuur 3.2: Overzicht van de spectra van glycerine (rood), sorbitol (zwart) en MPG (groen)

Figuur 3.2 toont dat grotendeels dezelfde pieken voorkomen voor glycerine, sorbitol en MPG. Vooral de intensiteiten van de pieken zijn anders. Echter zal daarop niet gedifferentieerd kunnen worden omdat bij mengsels van de stabilisatoren de intensiteiten van overeenkomstige pieken niet aan één component specifiek kunnen worden toegeschreven. Deze komen dan namelijk voor als een grote piek.

Voor MPG valt een specifieke piek op, namelijk rond 2970 cm^{-1}. Deze is afkomstig van de methylgroep (buigvibratie). Voor sorbitol is er ook een specifieke piek, namelijk die rond 1647 cm^{-1}. Deze piek is afkomstig van het aanwezige water in de grondstof en dus in de regel niet van sorbitol zelf. Het is wel mogelijk dat het neuraal netwerk hierop zal differentiëren.

Figuur 3.3 toont een IR-spectrum van een mengsel dat bestaat uit 33,77 % glycerine, 32,63 % sorbitol en 33,60 % MPG (dus ongeveer elk evenveel vertegenwoordigd).



Figuur 3.3: IR-spectrum waarbij glycerine, sorbitol en MPG gelijk vertegenwoordigd zijn qua concentratie

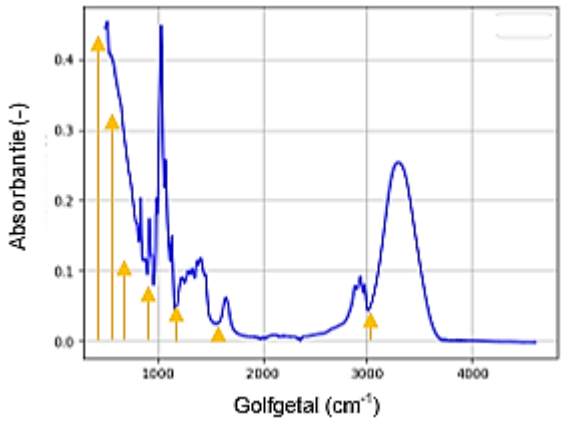
Figuur 3.3 toont dat door de overlappende pieken van de individuele spectra (Figuur 3.2 en paragraaf 2.2.3) het voor het menselijk oog quasi onmogelijk is om te differentiëren op specifieke pieken die rechtstreeks gelinkt zijn aan de componenten. Het is moeilijk om aan de hand van het spectrum getoond in Figuur 3.3, een kwalitatieve analyse te doen en dus de aanwezigheid van een specifieke stabilisator aan te tonen. Een kwantitatieve analyse uitvoeren, lijkt dan ook niet vanzelfsprekend door menselijke interpretatie.

Door het toepassen van deep learning, kan er wel een kwantitatieve analyse uitgevoerd worden. Een computeralgoritme ziet ook kleinere pieken. Deze zitten mogelijks verstopt achter andere pieken of zijn te klein om op te merken en kunnen wel specifiek gelinkt worden aan een individuele stabilisator. Door deze kennis kan een algoritme wél differentiëren op basis van het spectrum van een mengsel. Vanuit dit standpunt is het dus interessant om een kwantificatie uit te voeren met een deep learning model daar dit model meer inzicht heeft in een IR-spectrum dan het inzicht verkregen door menselijke interpretatie.

Tijdens het leerproces van een netwerk, analyseert het algoritme variaties in specifieke pieken gelinkt aan een bepaalde componenten (glycerine, MPG of sorbitol). Op basis van deze variaties, leert het algoritme welke piekhoogte (absorbantiewaarde) gelinkt is aan een welke concentratie. Een onbekende staal met dezelfde bestanddelen (glycerine, MPG en sorbitol) en dus pieken, kan verwerkt worden door het model waaruit een concentratie van de individuele componenten volgt. De opbouw van dergelijk model wordt besproken in paragraaf 3.2.5.

3.2.3 Implementatie van een basislijncorrectie

Bij de opname van een FTIR-spectrum blijkt uit de praktijk dat de basislijn verschuift naar boven toe. Door deze verschuiving wordt de kwantificatie bemoeilijkt. Het spectrum moet daarom gecorrigeerd worden op een reproduceerbare en herhaalbare manier. De correctie moet immers steeds op dezelfde wijze gebeuren zodanig dat de kwantificatie eenduidig is en het algoritme correct in staat kan gesteld worden om uit de data te leren. Figuur 3.4 geeft dit probleem weer op een IR-spectrum. De pijlen op Figuur 3.4 duiden de verschuivingen van de basislijn aan.



Figuur 3.4: Het probleem van een basislijnverschuiving op een niet-gecorrigeerd spectrum

De basislijncorrecties die in deze thesis gebruikt worden, zijn een AsLS- en een AirPLS-correctie. Dit zijn algoritmen die geïmplementeerd worden via het schrijven van een iteratief programma in Python. Hiervoor wordt gebruik gemaakt van gedefinieerde functies uit pakketten en zelf geschreven code. (96–99)

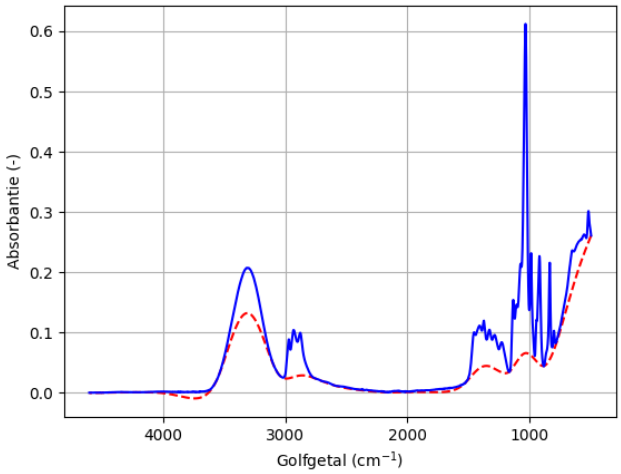
Bij deze algoritmes wordt ieder spectrum individueel (dus ieder staal afzonderlijk) gecorrigeerd. Beide basislijncorrecties maken gebruik van de Whittaker smoother. Deze smoother is een manier om het signaal gladder te maken zonder verlies aan relatieve gewichten. De correctie-algoritmen bevatten de wiskundige bewerkingen van AsLS en AirPLS. De Python-code die gebruikt wordt, is afkomstig van het ontwikkelaarsplatform GitHub. De code wordt aangepast om toe te passen op spectrale data. De toepassing van deze basislijncorrecties wordt in de volgende paragrafen in meer detail besproken. (96–99)

3.2.3.1 Toepassing van de AsLS-basislijncorrectie

AsLS staat zoals eerder vermeld voor Asymmetric Least Squares. Deze methode berust op paarsgewijze vergelijking tussen een gecorrigeerde spectrum en het origineel spectrum in combinatie met een smoother (Whittaker in dit geval). De evaluatie van deze vergelijking gebeurt op basis van de kleinste kwadraten (vandaar least squares in ‘AsLS’). De term ‘asymmetric’ vindt zijn oorsprong in de asymmetrieparameter die moet gedefinieerd worden bij de opstart van het algoritme. Deze parameter geeft aan welk deel van de meetpunten een nieuw gewicht moet krijgen. (96,97)

Ook een error-waarde wordt in rekening gebracht. Een gewichtsparameter wordt gekozen opdat de ruis niet de bovenhand neemt en opdat het signaal niet te veel wordt uitgevlakt. Ook regularisatieparameters worden gekozen opdat de fit correct gebeurt. Zodoende volgt de nieuwe basislijn de vorm van de curve. (96,97)

Figuur 3.5 toont een voorbeeld van een AsLS-basislijncorrectie toegepast op een infrarood spectrum.



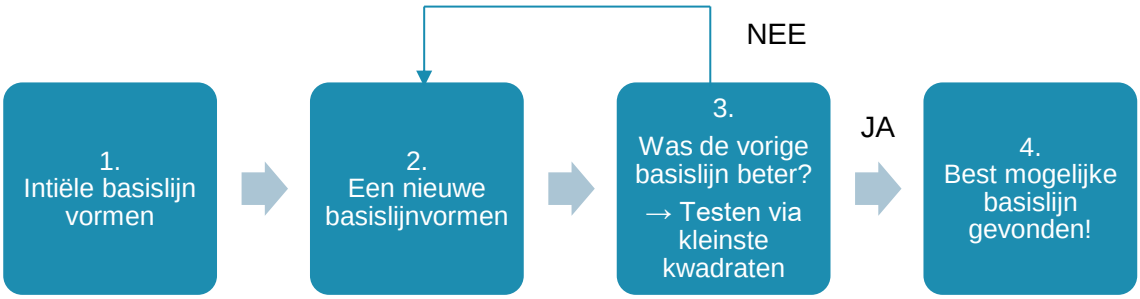
Figuur 3.5: AsLS-basislijncorrectie toegepast op een IR-spectrum

Figuur 3.5 geeft weer dat een AsLS-basislijncorrectie voltrokken wordt door middel van een polynoom. Dit heeft te maken met het feit dat een AsLS-algoritme een polynoomfitting aan de basis heeft: door de gecorrigeerde punten van de 'nieuwe' basislijn wordt een polynoom getekend. Dit is ook de reden waarom nog een dal te zien is rond tussen 3700 cm^{-1} en 3900 cm^{-1}. (96,97)

Een voorbeeld het resultaat van een AsLS-basislijncorrectie op een IR-spectrum, wordt besproken in paragraaf 3.2.3.3. De Python-code van deze basislijncorrectie kan gevonden worden in Bijlage A.

3.2.3.2 Toepassing van de AirPLS-basislijncorrectie

AirPLS berust op een mechanisme waarbij er adaptief (adaptively in 'AirPLS') en iteratief (iteratively in 'AirPLS') gezocht wordt naar de beste basislijn. Net zoals bij AsLS wordt dit getest door middel van de kleinste kwadraten methode (least squares in 'AirPLS'). Een gelijkaardige parameter zoals de gewichtsparameter bij AsLS, wordt meegegeven aan het algoritme. Deze parameter is een maat voor de ruwheid die nog is toegestaan in het signaal. Figuur 3.6 geeft het werkingsmechanisme van dit algoritme weer. (98,99)

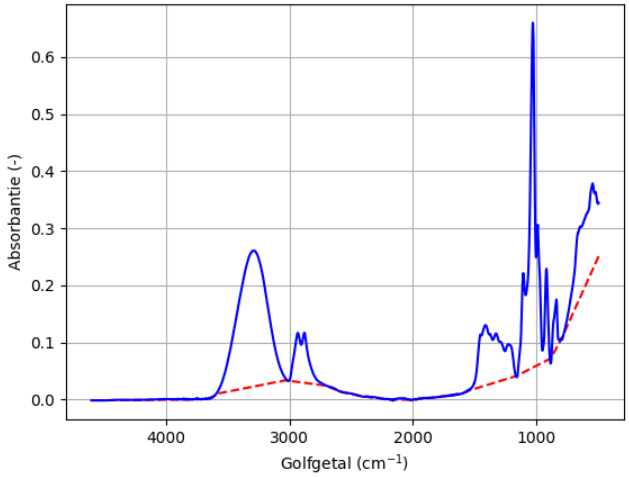


Figuur 3.6: Het adaptief en iteratief proces van AirPLS (99)

Figuur 3.6 toont dat er eerst een initiële basislijn gevormd in een eerste stap. Dit is een willekeurige inschatting. In een tweede stap wordt er een nieuwe inschatting gemaakt en worden deze basislijn met die uit stap 1 vergeleken. Afhankelijk van dit resultaat wordt er telkens een nieuwe basislijn gevormd en vergeleken met de vorige. De basislijnen worden vergeleken met elkaar door een kleinste kwadraten methode. Dit proces gaat door tot wanneer de vorige basislijn beter is dan (of gelijkwaardig is aan) een nieuw gegenereerde basislijn. Dit is het iteratieve karakter waar de 'A' in AirPLS naar verwijst. (98,99)

Een nieuwe basislijn wordt in stap 2 gevormd door iedere keer andere gewichten toe te kennen (vandaar reweighted in 'AirPLS'). Deze toekenning gebeurt op basis van de vorige waarden en worden adaptief (vandaar adaptively in 'AirPLS') bijgewerkt in de richting van een betere basislijn. 'Penalized' komt van het feit dat er een zogenoemde penalty-term wordt toegevoegd in het algoritme. Dit is typisch voor machine learning en deep learning waarbij deze term zorgt dat overfitting voorkomen wordt. (98,99)

Figuur 3.7 geeft een voorbeeld van een AirPLS-basislijncorrectie dat toegepast is op een IR-spectrum.



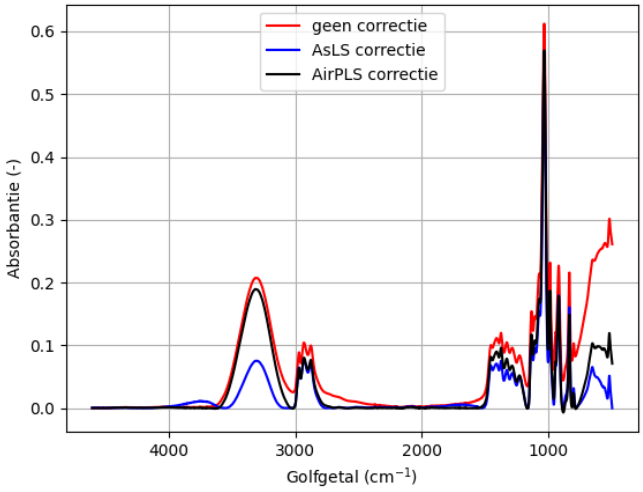
Figuur 3.7: AirPLS-basislijncorrectie toegepast op een IR-spectrum

Figuur 3.7 toont hoe een AirPLS-basislijncorrectie zorgt voor een lineaire afsnijding van de basis. Hier wordt het karakter van penalty-term gedemonstreerd. Door deze juist in te schatten, lijkt het immers alsof er een lineaire afsnijding is. Indien deze niet correct ingeschat zou worden, zou er een eerder polynomisch verloop waargenomen worden. Een polynomisch verloop wordt waargenomen in Figuur 3.6, bij de AsLS-basislijncorrectie, maar dit heeft niets te maken met een foutieve inschatting van de parameters. (98,99)

Een voorbeeld van het resultaat van een AirPLS-basislijncorrectie, wordt besproken in paragraaf 3.2.3.3. De desbetreffende Python-code wordt in Bijlage B getoond.

3.2.3.3 AsLS- en AirPLS-basislijncorrectie toegepast op een voorbeeld

Een voorbeeld van een gecorrigeerd met spectrum wordt gegeven in Figuur 3.8.

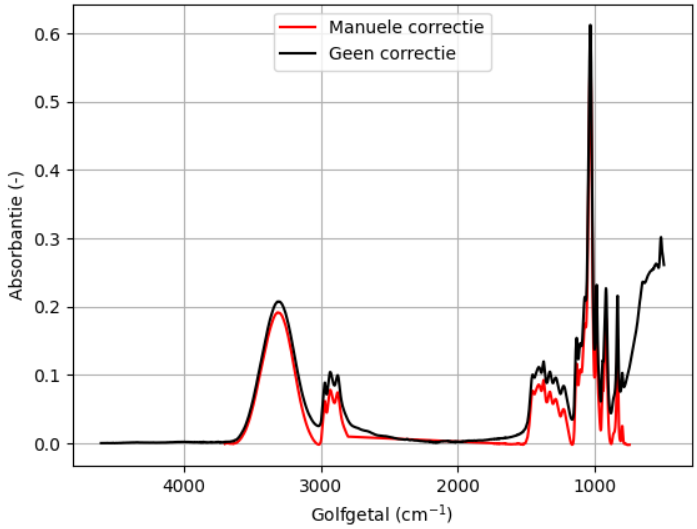


Figuur 3.8: Vergelijking van het origineel spectrum (rood) en de gecorrigeerde spectra met AsLS (blauw) en AirPLS (zwart)

Figuur 3.8 toont dat er onderlinge verschillen merkbaar zijn tussen de correctiemethoden. Bijvoorbeeld bij de hydroxylpiek (3288 cm^{-1}) en in de regio voor 1000 cm^{-1} geven de twee methoden aanleiding tot een ander resultaat. De manier van construeren ligt aan de basis van de onderlinge verschillen. De hydroxylpiek wordt hoger ingeschat bij een lineaire afsnijding door AirPLS (zie Figuur 3.7) dan bij een polynomische inschatting van de basislijn door AsLS (Figuur 3.5). In hoofdstuk 4 omtrent de resultaten en discussie wordt hier dieper op ingegaan.

3.2.3.4 Toepassing van een manuele basislijncorrectie

Ook wordt een manuele basislijncorrectie uitgevoerd. Deze wordt niet geprogrammeerd in Python maar uitgevoerd via de LabSolutions IR software gekoppeld aan de FTIR-spectrofotometer IRAffinity-1S (Shimadzu). Een voorbeeld van dergelijke correctie wordt gegeven in Figuur 3.9 en wordt vergeleken ten opzichte van het originele spectrum.



Figuur 3.9: Manuele basislijncorrectie (rood) versus origineel spectrum (zwart)

Figuur 3.9 toont dat het profiel van het origineel spectrum bewaard wordt door de manuele basislijncorrectie. Een vergelijking van een manuele basislijncorrectie met een AsLS- en een AirPLS-basislijncorrectie wordt gedaan in paragraaf 4.1.2.

3.2.4 Implementatie van pre-processingsmethoden

Een pre-processingstap wordt, zoals aangegeven in Figuur 3.1, uitgevoerd na een eventuele basislijncorrectie. Deze denkwijze wordt gevolgd aangezien de pre-processing werkt op basis van het spectrum in zijn uiteindelijke vorm. Twee mogelijk pre-processingsmethoden worden onderzocht, namelijk standard normal variate (SNV) en principale component analyse (PCA).

3.2.4.1 Standard normal variate

Standard normal variate ofwel SNV wordt in Python-code vertaald met dezelfde betekenis als Vergelijking 2.6 (paragraaf 2.3.3.2). In dit geval dient er echter geen rekening gehouden te worden met de basislijn, daar deze aangepast te wordt door een basislijncorrectie. Ook in het geval dat er geen basislijncorrectie gebeurt, dient er geen rekening gehouden te worden met de intensiteit van de basislijn, aangezien deze als constante factor wordt gehouden. De Python-code voor het SNV-algoritme kan worden teruggevonden in Bijlage C. Hierin wordt deze pre-processingsmethode voorgesteld als de functie ‘snv()’. (100)

De gedefinieerde functie ‘snv()’ heeft enkel een input nodig. De input is een IR-spectrum onder de vorm van een Excel-bestand. Een nieuwe rij wordt aangemaakt met dezelfde grootte als de inputdata. Vervolgens wordt Vergelijking 2.6 toegepast en wordt deze nieuwe waarde overgeschreven naar deze nieuwe, lege rij. De volledig gevulde rij en dus het gecorrigeerde spectrum, wordt als output teruggegeven.

3.2.4.2 Principale component analyse

Analoog aan het SNV-algoritme wordt een algoritme geconstrueerd voor de PCA pre-processing. Deze code wordt getoond Bijlage D en voorgesteld door de functie ‘PCA_Spectrale_Data()’. Ook hier is de enige vereiste input-parameter, de data die dient verwerkt te worden. De functie PCA wordt geïmporteerd uit het pakket ‘sklearn’. Er wordt gekozen voor twee PC’s wanneer binaire modellen worden gebruikt en voor drie PC’s wanneer een ternair model wordt gebruikt. (101)

Er wordt een transformatie doorgevoerd van het hele golfgebied naar twee of drie principale componenten en deze worden opgeslagen in een dataframe. Ook wordt de functie ‘Explained_variance_ratio_’ gebruikt. Deze functie geeft aan hoeveel procent van de variabiliteit is opgenomen in de eerste, de tweede en eventueel in de derde PC. Dit getal is best zo hoog mogelijk. (101)

3.2.5 Constructie van een deep learning neuraal netwerk

Om het neuraal netwerk op te bouwen met Python, wordt gebruik gemaakt van het boek *Deep learning with Python* en van de thesis *Interpretatie en modellering van multi instrumentele analytische data met Deep Learning* geschreven door Marjolein Saelens. In het boek en in de thesis, wordt beschreven hoe de opbouw precies dient te gebeuren en welke (hyper)parameters en pakketten er allemaal nodig zijn om het netwerk naar behoren te laten functioneren. (69,102)

De manier van werken voor de opbouw van een neuraal netwerk met behulp van deel learning, kan als volgt samengevat worden:

1. Laden van de benodigde pakketten (3.2.5.1)
2. Inlezen van de input en omzetten naar een dataframe (3.2.5.1)
3. Manipulatie van de input (3.2.5.2)
   - *3.I. Dataframe herindelen en/of herdefiniëren*
   - *3.II. Dataframe opsplitsen in een trainingset en testset, verschillend naargelang:*
     - *3.II.i. Binair mengsel van stabilisatoren*
     - *3.II.ii. Ternair mengsel van stabilisatoren*
4. Constructie van het neuraal netwerk (3.2.5.3)
   - *4.I. Opbouw van het model, verschillend naargelang:*
     - *4.I.i. Binair mengsel van stabilisatoren*
     - *4.I.ii. Ternair mengsel van stabilisatoren*
   - *4.II. Selectie van de hyperparameters*
   - *4.III. Trainen van het model*
5. Opslaan van het model en laden van het model (3.2.5.4)
6. Validatie van het model vanuit opgeslagen toestand (3.2.5.5)

3.2.5.1 Laden van de benodigde pakketten en inlezen van de data

De eerste stap in de opbouw van een neuraal netwerk, is het inladen van de nodige pakketten en het inlezen van de input. De Python-code die daarvoor nodig is, wordt weergegeven in Bijlage E. In deze bijlage worden ook alle andere codes weergegeven die nodig zijn voor de opbouw van een neuraal netwerk. Deze codes volgen de respectievelijke nummers die in het bovenstaande schema gebruikt worden.

De gebruikte pakketten zijn: 'Numpy', 'Keras', 'Pandas' en 'Matplot'. Uit deze pakketten kan via 'import' een specifieke functie geïmporteerd worden. Het is mogelijk om deze functie een kortere naam te geven via de 'as'-functie. Nadien kan deze opgeroepen worden via de kortere naam. (69)

Het inladen van het Excel-bestand gebeurt via het Pandas-pakket. Dit bestand wordt vervolgens omgezet naar een dataframe. Dit wordt gedaan om later gemakkelijker manipulaties op de data te kunnen uitvoeren, bijvoorbeeld het splitsen van het dataframe in een training- en testset (paragraaf 3.2.5.2). (69)

Indien er een basislijncorrectie en/of een pre-processing gebeurt, wordt dit gecorrigeerde spectrum ingelezen. De implementatie van een basislijncorrectie en pre-processing werd reeds besproken in respectievelijk paragraaf 3.2.3 en 3.2.4. (69)

3.2.5.2 Manipulatie van de data

Na het inlezen van de data, wordt deze gemanipuleerd en onderverdeeld in de juiste sets. Dit wordt ook weergegeven Bijlage E. De code wordt gebruikt om het dataframe in twee (bij binaire mengsels) of drie (bij ternaire mengsels) subsets in te delen. Deze subsets koppelen één concentratie van één stabilisator aan het spectrum van het mengsel. Bijvoorbeeld bij het MPG-sorbitol mengsels, worden twee subsets gemaakt: een dataset waarbij de concentratie van MPG gelinkt wordt aan het spectrum van het mengsel en een dataset die hetzelfde doet voor sorbitol. (69)

Deze subsets van data, worden vervolgens elk opnieuw ingedeeld in een training- en testset volgens (respectievelijk) een 80 % - 20 % regel. Om later de opbouw van het model vlot te laten verlopen, worden de dataframes omgezet in rijen en vertaald naar het karaktertype 'float32'. (69)

3.2.5.3 Constructie van het neuraal netwerk

I) Opbouw van het model

Eens de data gemanipuleerd is, kan het neuraal netwerk worden opgebouwd. In Bijlage E kan de desbetreffende Python-code worden teruggevonden. De code in Bijlage E wordt specifiek gebruikt om een neuraal netwerk met drie lagen op te bouwen. Er wordt een onderscheid gemaakt qua neuraal netwerk tussen de verwerking van binaire en ternaire mengsels van stabilisatoren. De woorden die in de volgende paragrafen in het vet zijn gedrukt, zijn parameters die moeten gedefinieerd worden bij de opstart van het algoritme. (69,102)

I.i) Opbouw van een binair model

De code voor een binair model, bevat onder andere de functie ‘Sequential()’. Dit is een functie die opgeroepen wordt zodat een model wordt gecreëerd dat meerdere lagen kan bevatten. Die lagen zijn sequentieel met elkaar verbonden. Een laag toevoegen gebeurt door het gebruik van de ‘model.add’-functie. Het woord ‘dense’ (dicht) in de Python-code in Bijlage E, staat voor het gebruik van een dichte laag. Verder is de **activatiefunctie** in dit neuraal netwerk de ReLU-functie aangezien geen negatieve waarden worden verwacht. (69)

De **input-shape** heeft betrekking op de grootte van de dataset die aan het neuraal netwerk gevoed wordt. Dit wordt gedaan onder de vorm ‘(*getal,)*’. In het geval van een PCA-matrix met twee PC’s (binair model) is de input-shape ‘(2,)’. Het feit dat er geen getal volgt na de komma, duidt erop dat niet op voorhand wordt vastgelegd hoeveel regels (en dus stalen) er in de dataset zitten en bijgevolg worden gevoed aan het netwerk. In het geval van geen of een andere dan PCA pre-processing, is de input-shape ‘(2127,)’, aangezien er 2127 golfgetallen in een spectrum zitten. (69)

Daarnaast worden ook het **aantal (verborgen) eenheden** meegegeven. De laatste laag bevat slechts één eenheid met name de output. In het netwerk is de output de ingeschatte concentratie. Zoals reeds uitgelegd in paragraaf 2.4.3.3 wordt er ook een **optimizer** gebruikt. In dit geval is dat een RMSPROP. Het **verlies** in het netwerk wordt steeds gemeten aan de hand van de mean absolute error ofwel de gemiddelde fout (MAE). Dit wordt berekend tussen de voorspelde en de werkelijke waarde. Hierop wordt dan beoordeeld of een model al dan niet goed scoort. (69)

I.ii) Opbouw van een ternair model

De opbouw van een neuraal netwerk voor een ternair mengsel is analoog maar niet geheel gelijk. Ook deze code wordt weergegeven in Bijlage E. De grootste verschillen tussen de code van een binair en een ternair neuraal netwerk, is dat het model voor een binair mengsel eerst gevormd wordt en dan pas ingevuld wordt terwijl dit voor het ternair mengsel net andersom is. Eerst wordt de input gedefinieerd en daarna worden de lagen gemaakt. De code in Bijlage E toont twee dichte lagen met een 32 eenheden en een ReLU-activatiefunctie. (69)

De koppeling gebeurt in deze niet met de ‘Sequential()’-functie maar met een ‘(x)’ achteraan de lijn code. Er worden nu drie concentraties ingeschat en er moet dus per component een output gegenereerd worden. Door het werken met een ‘(x)’, kan het aantal lagen gemakkelijk worden aangepast. (69)

Nadat de lagen gevormd zijn, wordt het model samengesteld met de functie ‘Model()’. De input en de output die verbonden zijn aan de voorgaande lagen door een ‘(x)’, worden als parameters meegegeven. Nadien dient de ‘Compile()’-functie voor de samenstelling te voltrekken door de definitie van de optimizer en de verliesfunctie. De input-shape is hier ‘(3,)’ in het geval van PCA en ‘(2127,)’ in de andere gevallen. (69)

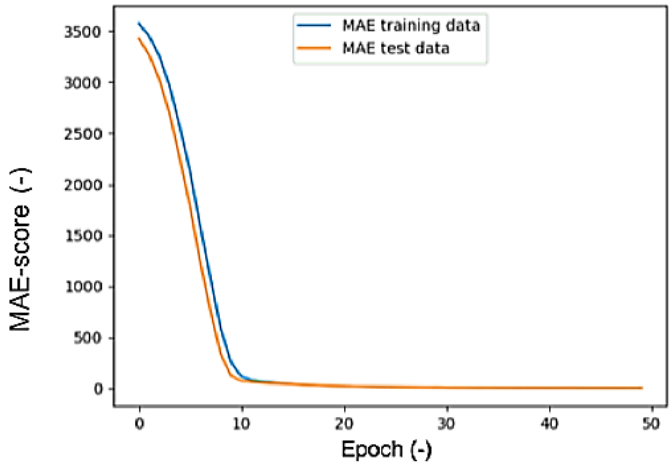
II+III) Selectie van de hyperparameters en trainen van het model

Om het netwerk te trainen en vervolgens te testen, dient meegegeven te worden wat de hyperparameters zijn. De desbetreffende Python-code kan teruggevonden worden in Bijlage E. Deze code geldt voor het binaire model. De code voor een model met meer componenten is analoog. (69)

Het **aantal epochs**, de **batch-grootte** (batch-size) en de **data** die moet gebruikt worden, dienen meegegeven te worden via de functie ‘model.fit()’. Zoals eerder vermeld, is een epoch het volledig doorlopen van een dataset en is de batch-grootte het aantal waarden dat simultaan door het algoritme verwerkt worden. (69)

In bijlage E wordt er nog een andere parameter vermeld, namelijk de parameter ‘verbose’. Deze geeft de mogelijkheid om tussentijdse MAE-resultaten weer te geven. Een waarde ‘1’ staat voor weergeven en een ‘0’ staat voor niet weergeven. (69)

Het verloop van de history-paramater geeft inzicht in mogelijke over- of underfitting. Daartoe worden de verliezen van trainings- en testdata vergeleken. Figuur 3.10 geeft een voorbeeld van dergelijk geconstrueerde grafiek voor een binair mengsel. (69)



Figuur 3.10: Weergave van under- of overfitting voor een binair mengsel

Figuur 3.10 toont dat er geen overfitting is na het doorlopen van 50 epochs daar de trainings- en testcurve samenvallen. Indien er wel overfitting zou zijn, zouden de curven uit elkaar gaan. Dit werd reeds uitgelegd in paragraaf 2.4.3.4. Figuur 3.10 toont dat de MAE-score snel daalt tussen 0 en 10 epochs en nadien stabiliseert. (69)

3.2.5.4 Opslaan en laden van een model

Het model dient opgeslagen te worden op de harde schijf van de computer wanneer nadien een validatie met nieuwe data dient te gebeuren. Het model wordt iteratief opgebouwd. Dit impliceert dat per keer dat het model opnieuw wordt opgebouwd, er andere resultaten worden gevonden. Aangezien er zo geen reproduceerbaarheid mogelijk is, wordt het model opgeslagen. De Python-code hiervoor kan worden teruggevonden in Bijlage E. (69)

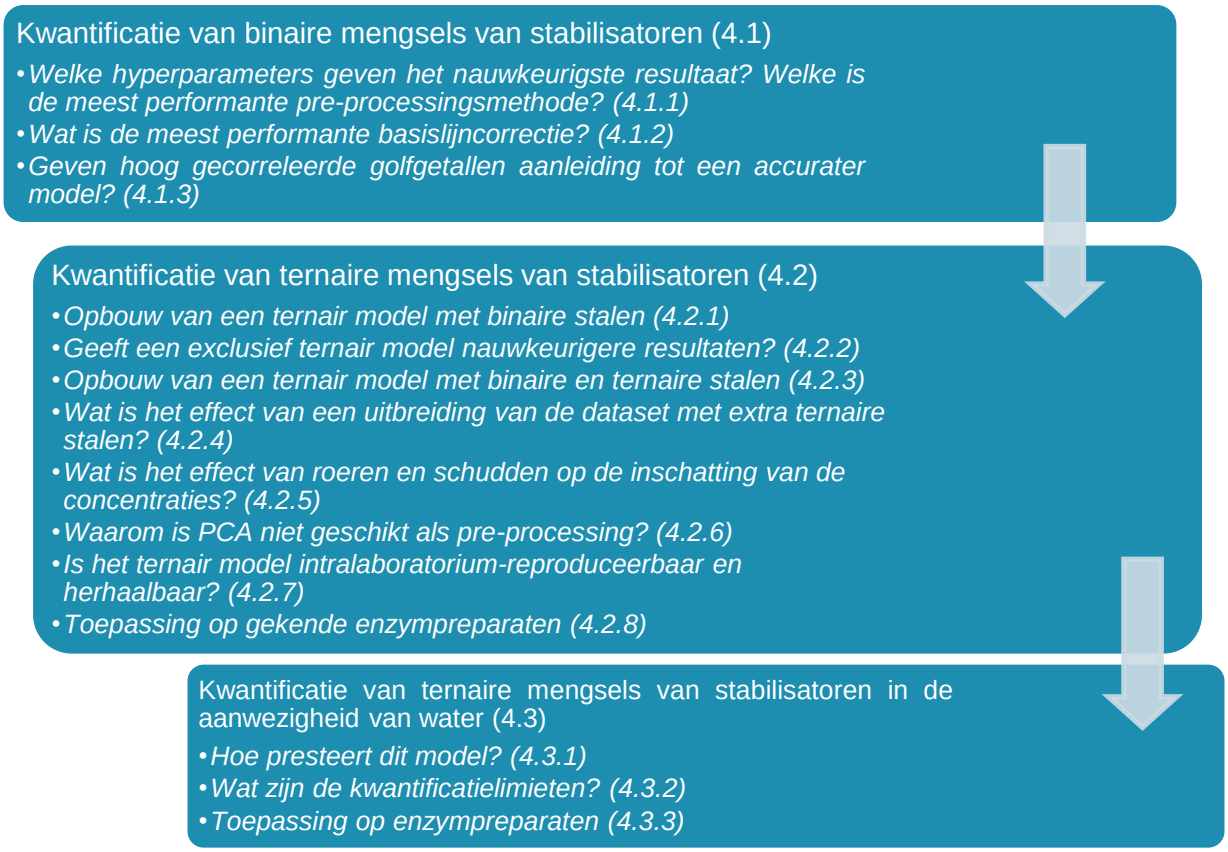
Nadat een model is opgeslagen, kan het gebruikt worden ter validatie van nieuwe data. Daartoe wordt het opgeroepen via de functie ‘Load()’. De Python-code die daarvoor nodig is, kan worden teruggevonden in Bijlage E. (69)

3.2.5.5 Validatie van een model vanuit opgeslagen toestand

Na het laden van een model, kan de validatie voltrokken worden. Als parameter wordt dan de validatiedata meegegeven. De voorspelde waarden kunnen vervolgens vergeleken worden met de werkelijke waarde en hierop kan de MAE of de AE berekend worden. De MAE- of AE-waarde van de validatie laat toe een beoordeling te maken omtrent de performantie van het model. Immers wordt de validatie gedaan met onbekende stalen voor het model. Des te accurater het model daarop scoort, des te performanter het model is in effectief gebruik bij analyses. Ook deze Python-codes kunnen worden teruggevonden in Bijlage E. (69)

4 RESULTATEN EN DISCUSSIE

Deze discussie bevat een opeenvolging van logische stappen qua experimenten ter kwantificatie van een ternair mengsel van stabilisatoren. Om het overzicht te behouden omtrent wat besproken zal worden, toont Figuur 4.1 schematisch wat er allemaal aan bod komt in deze paragraaf. Er wordt gestart met een reeks experimenten met binaire mengsels. Die kennis wordt meegenomen naar de experimenten met ternaire mengsels. Vervolgens wordt het meest complexe model opgebouwd. Dit is het model voor ternaire mengsels van stabilisatoren in de aanwezigheid van water. De complexiteit zit in de inschatting van vier componenten tegelijkertijd. Dit model zal overigens toegepast worden op enzympreparaten.



Figuur 4.1: Overzicht van de uitgevoerde experimenten

4.1 Kwantificatie van binaire mengsels van stabilisatoren

De binaire mengsels die geanalyseerd en verwerkt worden, zijn steeds combinaties van ofwel glycerol en MPG ofwel sorbitol en MPG. Het is de bedoeling dat de concentraties van deze mengsels zo nauwkeurig mogelijk worden ingeschat door het model. Naargelang de (hyper)parameters die in het neuraal netwerk worden gebruikt, zal een andere inschatting gebeuren. Het doel van deze set experimenten is om aan de ene kant te onderzoeken welke basislijncorrectie het meest performant is en aan de andere kant de verschillende pre-processingsmethoden met elkaar te vergelijken. Deze resultaten worden meegenomen naar

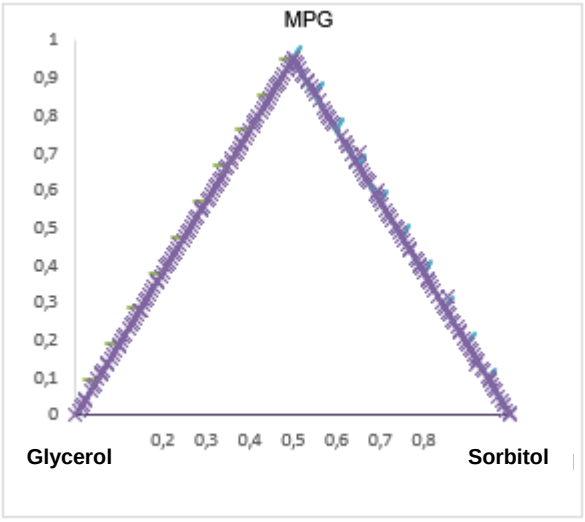
een volgende set experimenten waarbij ternaire mengsels van stabilisatoren worden onderzocht.

De trainings- en testfase worden volbracht met telkens 125 binaire stalen (100 stalen voor de training en 25 stalen voor het testen). De validatiefase zal worden voltrokken met 58 binaire stalen waarvan 28 van het mengsel MPG-glycerol en 30 van het MPG-sorbitol mengsel. Tabel 4.1 geeft dit weer.

Tabel 4.1: Overzicht van het aantal training-, test-, en validatiestalen

Trainingstalen	Teststalen	Validatiestalen
MPG-glycerol: 100 stalen	MPG-glycerol: 25 stalen	MPG-glycerol: 28 stalen
MPG-sorbitol: 100 stalen	MPG-sorbitol: 25 stalen	MPG-sorbitol: 30 stalen

De stalen in Tabel 4.1 zijn aangemaakt in welbepaalde concentraties. Figuur 4.2 geeft aan welke binaire concentraties er gehanteerd worden. Dit wordt voorgesteld in een ternair diagram. Er wordt voor dit type diagram gekozen omdat zo ook de evolutie van de concentraties bij de ternaire experimenten kunnen worden opgevolgd.



Figuur 4.2: Concentraties van de binaire stalen voorgesteld op een ternair diagram

De modellen worden vergeleken op basis van de absolute fout (AE) en de gemiddelde waarde van de absolute fout over het hele concentratiebereik (MAE). De standaardafwijkingen die gerapporteerd worden, hebben eveneens betrekking op de AE of de MAE.

4.1.1 Evaluatie van de binaire modellen en variatie van de hyperparameters

4.1.1.1 Evaluatie van de binaire modellen

In elk neuraal netwerk is het mogelijk om de hyperparameters aan te passen zodanig dat er een optimalisatie van het model voltrokken wordt. Deze optimalisatie is een manueel en iteratief proces en werd door trial-and-error aangepast. Op die manier werd gekeken naar welke modellen aanleiding gaven tot de nauwkeurigste trainings- en testresultaten. Daarna wordt een validatie met onbekende stalen uitgevoerd en wordt de MAE vergeleken met andere modellen.

De waarden in Tabel 4.2 zijn de hyperparameters die gebruikt werden om de verschillende modellen op te bouwen. Dezelfde variaties in de hyperparameters werden gebruikt voor de verschillende basislijncorrecties en pre-processingsmethoden. Om correct te vergelijken, werd telkens gerefereerd worden naar dezelfde hyperparameters.

Tabel 4.2: Overzicht van de gebruikte hyperparameters

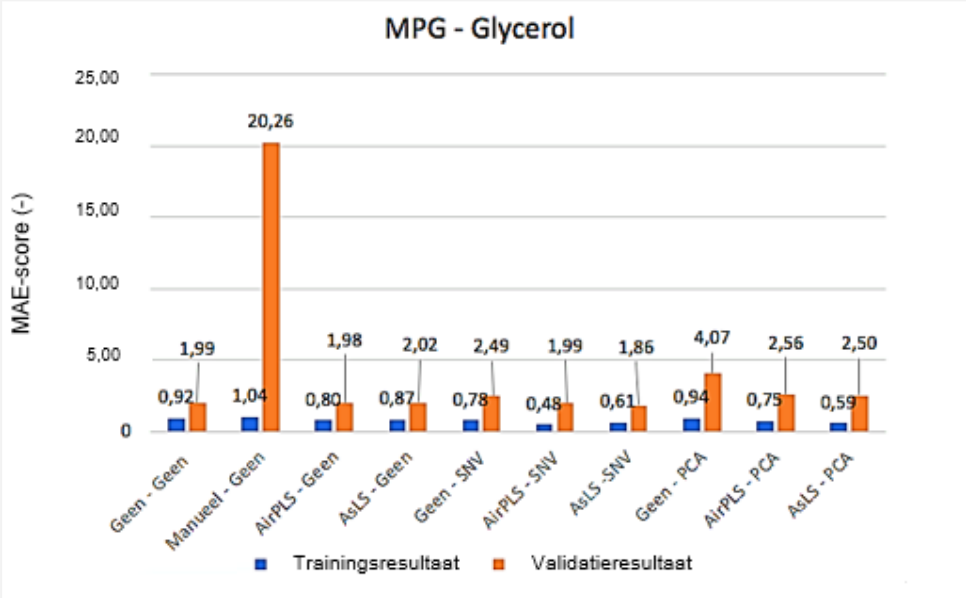
Hyperparameter	Waarde (-)
Epochs	69 - 120 - 333 - 500
Aantal lagen	3 - 4
Eenheden in de laag	32 - 64
Batch-grootte	1 - 2 - 5

In totaal werden 116 modellen getraind. Deze werden opgebouwd door een combinatie van de hyperparameters in Tabel 4.2, de verschillende mogelijkheden van de basislijncorrecties en pre-processingsmethoden.

Wanneer PCA als pre-processingsmethode werd gehanteerd, werden er steeds twee principale componenten gebruikt die meer dan 97 % van de variabiliteit van de data bevatten. Dit is zo voor elk binair experiment waarin PCA als pre-processing gebruikt wordt.

Voor de validatie werd verder gewerkt met de vier meest accurate modellen per combinatie van pre-processing en basislijncorrectie. Zodoende werden 80 modellen uitgekozen en verder geëvalueerd in de validatiefase. De modellen werden geselecteerd op basis van de MAE-waarde die zo laag mogelijk diende te zijn. Van de 80 gevalideerde modellen wordt telkens het model (per combinatie van een basislijncorrectie en pre-processingsmethode) uitgekozen die de laagste MAE-score had. Zodoende worden er 20 modellen overgehouden (10 modellen per binair mengsel).

Figuur 4.3 geeft grafisch de MAE-scores weer van de tien nauwkeurigste modellen bij de kwantificatie van het binair mengsel MPG-glycerol.



Figuur 4.3: Grafische weergave van de laagste MAE-scores uit de trainings- en validatiefase voor het MPG-glycerol mengsel

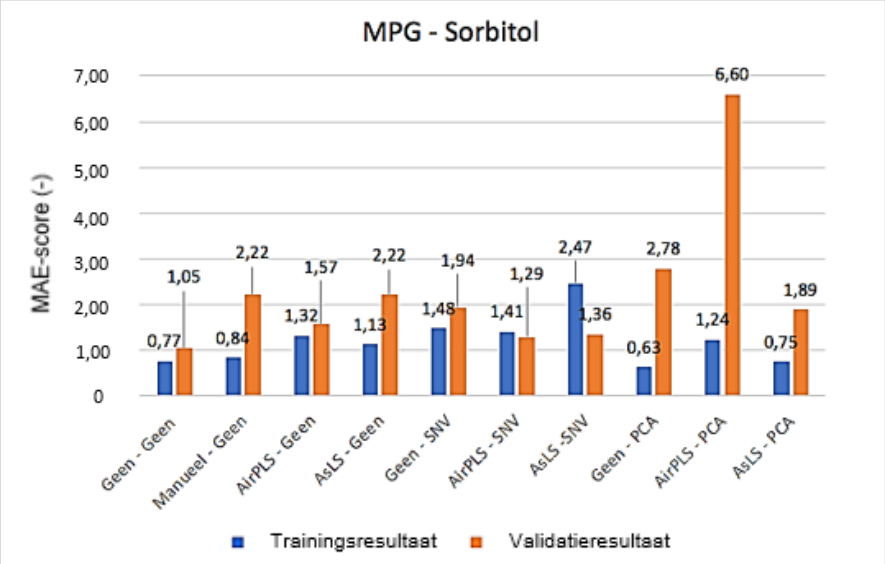
Figuur 4.3 geeft weer dat de trainingscores bij alle modellen gelijkaardig is. Het 'AirPLS-SNV'-model heeft de laagste trainingsscore, namelijk 0,48 ± 0,54. Een 'AirPLS-SNV'-model staat voor een model dat opgebouwd is via een AirPLS-basislijncorrectie in combinatie met SNV pre-processing. Het model die het minst nauwkeurig de concentraties kan inschatten in de trainingsfase, is het 'Manueel-Geen'-model met een MAE-score van 1,04 ± 0,66.

Voor de validatiescores geldt eveneens dat het merendeel van de modellen gelijkaardig scoort. Het model met de laagste MAE-waarde, is het 'AsLS-SNV'-model met een MAE-score van 1,86 ± 1,86. Het 'Geen-PCA'-model heeft een lichtelijk hogere MAE-score dan de overige modellen. Figuur 4.3 toont dat de hoogste MAE-validatiescore gevonden wordt bij het 'Manueel-Geen'-model, namelijk 20,26 ± 12,81. De inschattingen die dit model maakt, zijn met andere woorden moeilijk in overeenstemming te brengen met de werkelijke concentraties.

Ook toont Figuur 4.3 dat bijvoorbeeld het 'AsLS-PCA'-model een lagere MAE-trainingscore heeft dan het 'AirPLS-geen'-model. Toch blijkt de validatiescore van dit laatste model kleiner te zijn. De reden van deze discrepantie is te vinden in de manier waarop het model werkt. Het model schat in de trainingsfase de gewichten van de inputvectoren in op een iteratieve manier. Deze gewichten worden getest door een testset waarbij de MAE-score wordt afgebeeld in Figuur 4.3 (Trainingsresultaat). Nadien worden er onbekende validatiestalen aan het model gevoed. Deze validatiestalen zijn niet rechtstreeks gelinkt aan de gewichten van de trainingsstalen waardoor er geen verband is tussen de uitkomst van de trainingsfase en de validatiefase. Dit maakt, zoals Figuur 4.3 toont, dat het moeilijk is om in de trainingsfase in te schatten of een model goed zal scoren bij de validatie.

Ook valt op in Figuur 4.3 dat alle modellen accurater scoren in de trainingsfase dan in de validatiefase. Dit komt omdat in de validatiefase onbekende stalen worden gebruikt om het model te valideren. Ook hier liggen de inschattingen van de gewichten in de trainingsfase aan de basis van inschattingsfouten in de validatie.

Figuur 4.4 toont de resultaten van de binaire experimenten met het MPG-sorbitol mengsel betreffende de training en de validatie.



Figuur 4.4: Grafische weergave van de laagste MAE-scores uit de trainings- en validatiefase voor het MPG-sorbitol mengsel

Om beter op de individuele verschillen in te gaan, heeft Figuur 4.4 een andere schaling in de y-as dan Figuur 4.3. Hierbij wordt een maximale MAE-score van 7 aangeduid. De verschillen tussen training en validatie zijn ongeveer even groot bij de twee binaire mengsels maar door de andere schaling worden deze uitvergroot.

Figuur 4.4 toont dat voor het MPG-sorbitol het laagste MAE-trainingsresultaat (0,63 ± 0,46) wordt verkregen bij het 'Geen-PCA'-model. De andere modellen scoren gelijkaardig in de training. Het hoogste trainingsresultaat wordt gevonden bij het 'AsLS-SNV'-model met een MAE-score van 2,47 ± 1,68.

De validatiescores die worden weergegeven in Figuur 4.4, tonen aan dat de laagste MAE-waarde wordt teruggevonden bij het 'Geen-Geen'-model met een MAE-waarde van 1,05 ± 1,45. Het minst nauwkeurige model is het 'AirPLS-PCA'-model met een MAE-score van 6,60 ± 3,69.

In dit experiment valt op dat het 'Manueel-Geen'-model vergelijkbaar scoort met de andere modellen. Daarnaast wordt in tegenstelling tot het binair mengsel MPG-glycerol (Figuur 4.3) gevonden dat er bij het MPG-sorbitol mengsel (Figuur 4.4) twee modellen zijn die beter scoren in de validatiefase dan in de trainingsfase. Dit wordt gevonden bij het 'AsLS-SNV'-model en het 'AirPLS-SNV'-model. De score van de validatie is overigens kleiner dan bij het MPG-glycerol mengsel. Net zoals bij het MPG-glycerol mengsel, geeft PCA pre-processing in vergelijking met de andere pre-processingsmethoden aanleiding tot hogere validatiescores.

Ter vervollediging wordt Tabel 4.3 weergegeven. Deze Tabel 4.3 geeft een overzicht van de resultaten van de trainingsfase, alsook van de hyperparameters waarbij dit resultaat voorkomt.

Tabel 4.3: Resultaten van de trainingsfase met aanduiding van het model met de laagste MAE-score (groen) en de hoogste MAE-score (oranje)

Basislijncorrectie	Pre-processing	MPG - glycerol	
		Hyperparameters (epochs - batch-grootte - hidden units)	Trainingsresultaat (MAE-score ± standaardafwijking) (-)
Zonder correctie	Geen	69-1-64	0,92 ± 0,78
Manuele correctie		333-1-64	1,04 ± 0,66
AirPLS-basislijncorrectie		333-1-32	0,80 ± 0,59
AsLS-basislijncorrectie		333-1-32	0,87 ± 0,84
Zonder correctie	SNV + standaardisatie	69-1-64	0,78 ± 0,84
AirPLS-basislijncorrectie		333-1-64	0,48 ± 0,54
AsLS-basislijncorrectie		333-1-32	0,61 ± 0,68
Zonder correctie	PCA	333-1-32	0,94 ± 0,63
AirPLS-basislijncorrectie		69-1-64	0,75 ± 0,48
AsLS-basislijncorrectie		333-1-64	0,59 ± 0,40
Basislijncorrectie	Pre-processing	MPG - sorbitol	
		Hyperparameters (epochs - batch-grootte - hidden units)	Trainingsresultaat (MAE-score ± standaardafwijking) (-)
Zonder correctie	Geen	333-1-32	0,77 ± 0,52
Manuele correctie		333-1-32	0,84 ± 0,63
AirPLS-basislijncorrectie		333-1-32	1,32 ± 1,48
AsLS-basislijncorrectie		69-1-64	1,13 ± 0,91
Zonder correctie	SNV + standaardisatie	333-1-64	1,48 ± 1,39
AirPLS-basislijncorrectie		333-1-32	1,41 ± 1,12
AsLS-basislijncorrectie		333-1-32	2,47 ± 1,68
Zonder correctie	PCA	333-1-32	0,63 ± 0,46
AirPLS-basislijncorrectie		333-1-64	1,24 ± 1,10
AsLS-basislijncorrectie		333-1-32	0,75 ± 0,88

Uit Tabel 4.3 kunnen dezelfde conclusies worden getrokken als uit Figuur 4.3 en Figuur 4.4. Ook volgt uit Tabel 4.3 dat het model met de laagste MAE-score teruggevonden wordt bij hetzelfde aantal epochs (333) en dezelfde batch-grootte (1) voor zowel het MPG-glycerol als het MPG-sorbitol mengsel.

Tabel 4.4 wordt eveneens ter vervollediging meegegeven en geeft een overzicht van de modellen met de laagste validatiescores.

Tabel 4.4: Resultaten van de validatiefase met aanduiding van het model met de laagste MAE-score (groen) en de hoogste MAE-score (oranje)

Basislijncorrectie	Pre-processing	MPG - glycerol	
		Hyperparameters (epochs - batch-grootte - hidden units)	Validatieresultaat (MAE-score ± standaardafwijking) (-)
Zonder correctie	Geen	69-1-64	1,99 ± 1,76
Manuele correctie		333-1-64	20,26 ± 12,81
AirPLS-basislijncorrectie		333-1-64	1,98 ± 1,78
AsLS-basislijncorrectie		333-1-64	2,02 ± 2,02
Zonder correctie	SNV + standaardisatie	69-1-64	2,49 ± 1,76
AirPLS-basislijncorrectie		69-1-64	1,99 ± 1,60
AsLS-basislijncorrectie		69-1-64	1,86 ± 1,86
Zonder correctie	PCA	69-5-64	4,07 ± 2,76
AirPLS-basislijncorrectie		333-1-64	2,56 ± 1,41
AsLS-basislijncorrectie		69-5-64	2,50 ± 1,63
Basislijncorrectie	**Pre-processing**	**MPG - sorbitol**	
		Hyperparameters (epochs - batch-grootte - hidden units)	**Validatieresultaat (MAE-score ± standaardafwijking) (-)**
Zonder correctie	Geen	69-1-64	1,05 ± 1,45
Manuele correctie		69-1-64	2,22 ± 2,04
AirPLS-basislijncorrectie		69-5-64	1,57 ± 1,11
AsLS-basislijncorrectie		69-1-64	2,22 ± 1,61
Zonder correctie	SNV + standaardisatie	333-1-32	1,94 ± 1,81
AirPLS-basislijncorrectie		333-1-32	1,29 ± 1,04
AsLS-basislijncorrectie		333-1-32	1,36 ± 0,85
Zonder correctie	PCA	69-5-64	2,78 ± 1,97
AirPLS-basislijncorrectie		333-1-64	6,60 ± 3,69
AsLS-basislijncorrectie		69-5-64	1,89 ± 2,07

Dezelfde conclusies volgen uit Tabel 4.4 zoals uit Figuur 4.3 en Figuur 4.4.

Er wordt op basis van Tabel 4.3 en Tabel 4.4 opgemerkt dat in de trainingsfase de MAE-waarden in het algemeen kleiner zijn in het geval van het mengsel MPG-glycerol dan van het MPG-sorbitol mengsel. De oorzaak hiervan kan niet gelinkt worden aan het IR-spectrum door het ontbreken van een specifieke piek voor glycerol, althans voor het menselijk oog. Het neuraal netwerk kan echter wel een dergelijke piek in het spectrum van een MPG-glycerol mengsel gevonden hebben waardoor de kwantificatie nauwkeuriger verloopt.

Op basis van al deze bevindingen uit de binaire experimenten, wordt besloten dat de MAE-validatiescores te weinig verschillen om een ideale combinatie af te leiden. Er kan dus niet gezegd worden dat een bepaalde basislijncorrectie performanter is dan een andere. Hetzelfde geldt voor de pre-processingsmethoden.

De minder performante combinaties kunnen er echter wel uitgehaald worden. De manuele basislijncorrectie bleek aanleiding te geven tot minder nauwkeurige modellen. Daarom wordt deze manier van corrigeren ook uitgesloten in volgende experimenten.

Ook zorgt het gebruik van PCA als pre-processingsmethode voor hogere MAE-scores. Uit de literatuur bleek echter dat dit wel een interessante methode zou zijn inzake deep learning. Dit wordt later afgetoetst met het model dat zal worden opgebouwd voor ternaire mengsels.

4.1.1.2 Variatie van de hyperparameters

De variatie van de hyperparameters werd individueel gedaan per combinatie van een basislijncorrectie en een pre-processing. Uit Tabel 4.3 en Tabel 4.4 volgt dat algemeen gezien de MAE-scores kleiner worden bij een toenemend aantal epochs. Een epoch-waarde van 500 is in de meeste gevallen echter te groot. Individueel zal er voor elke combinatie van een basislijncorrectie en een pre-processingsmethode een optimaal aantal epochs zijn.

De batch-grootte werd ook gevarieerd. In Tabel 4.3 en Tabel 4.4 staat louter de waarde '1' als grootte van de batch aangezien een batch-grootte van 2 of 5 steeds minder nauwkeurige resultaten gaven. Tabel 4.5 toont wat de gevolgen zijn van een variatie van de batch-grootte op vlak van MAE-trainingscore voor het 'AirPLS-Geen'-model. Hierbij zijn het aantal epochs ingesteld op 69, het aantal lagen op 3 en het aantal hidden units op 64.

Tabel 4.5: Variatie van de batch-grootte voor het 'AirPLS-Geen'-model

Batch-grootte (-)	MAE-score (-)
1	1,19
2	1,39
5	2,30

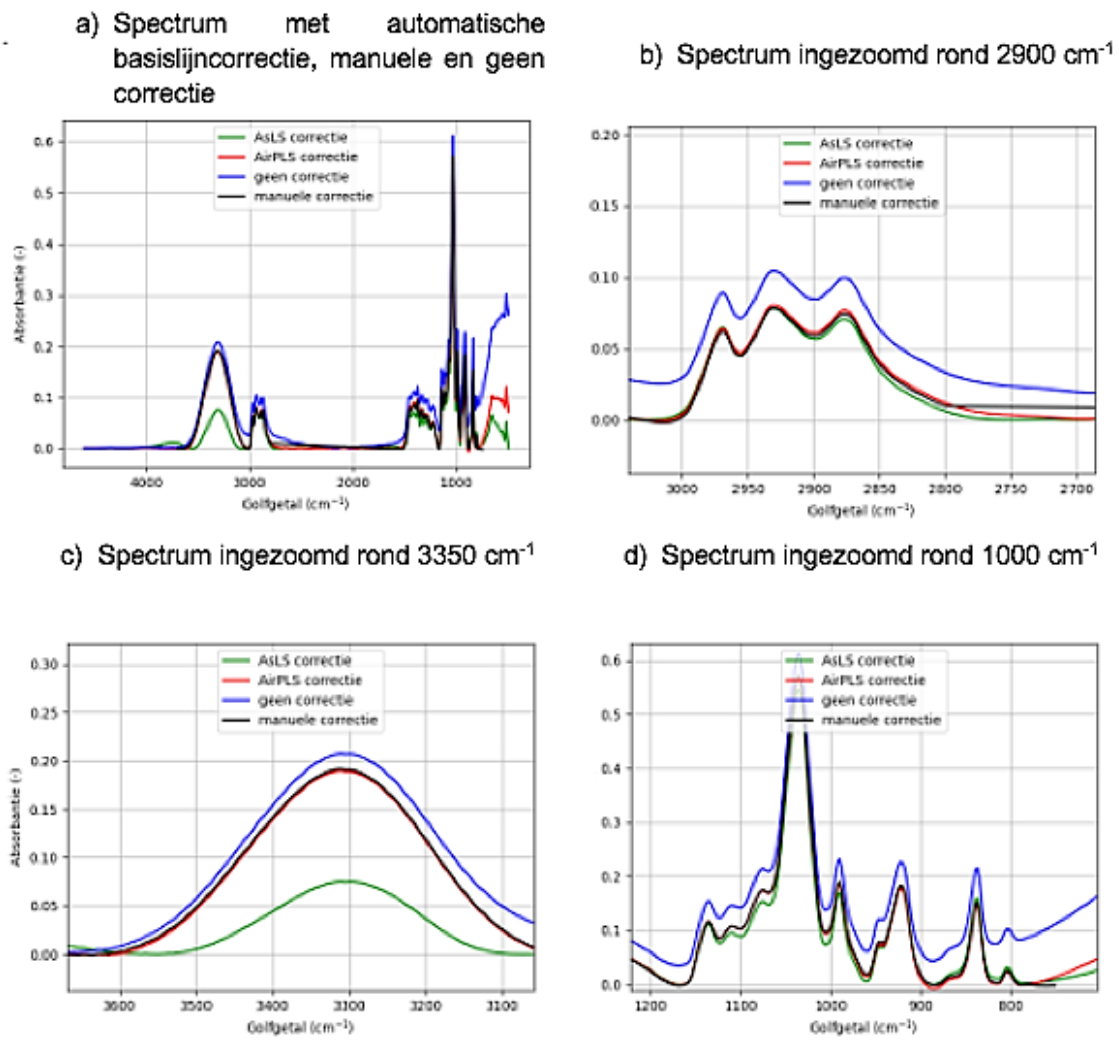
Tabel 4.5. toont aan dat naarmate de batch-grootte toeneemt, de MAE-score eveneens toeneemt. Hier wordt het voorbeeld aangehaald van het 'AirPLS-Geen'-model, maar uit experimenten bleek dat dit eveneens doorgetrokken kan worden voor de andere basislijncorrecties en pre-processingsmethoden.

Moeilijker is het om een algemene trend vast te stellen voor het aantal eenheden en het aantal epochs. Zoals Tabel 4.3 toont, is het veelal een hoge epoch-waarde die aanleiding geeft tot een laag resultaat maar dat is niet altijd het geval. Bij de validatie wordt opgemerkt dat meestal een kleiner aantal epochs aanleiding geeft tot een beter resultaat (Tabel 4.4). Hetzelfde geldt voor het aantal hidden units.

De onderlinge verschillen tussen de modellen zijn te klein om gefundeerde conclusies te trekken omtrent het aantal verborgen eenheden en het aantal epochs. Dit wordt bemoeilijkt door de individuele optimalisatie die mogelijk is per model. Een complexer mengsel met name het ternair mengsel van stabilisatoren, zou aanleiding kunnen geven tot een duidelijker resultaat. Er kon wel worden vastgesteld dat een batch-grootte gelijk aan 1 aanleiding geeft tot de nauwkeurigste inschattingen.

4.1.2 Vergelijking van de verschillende basislijncorrecties

Zoals in paragraaf 4.1.1 aangehaald, is het moeilijk om een besluit te trekken omtrent welk model en daarbij horend welke basislijncorrectie het meest geschikt is. Er kon wel besloten worden dat een manuele basislijncorrectie geen aanleiding geeft tot accurate inschattingen van de concentraties van glycerol, sorbitol en MPG. De reden hiervoor is een manuele correctie een verandering van de data teweegbrengt, waardoor de data niet langer in overeenstemming te brengen is met het oorspronkelijke profiel van het spectrum. De manuele inschatting verandert het profiel dusdanig waardoor het model geen accurate inschattingen kan leveren. Figuur 4.5 geeft een vergelijking weer van de automatische, manuele en niet-gecorrigeerde spectra.



Figuur 4.5: Vergelijking van de basislijncorrecties; a) AsLS-, AirPLS-, geen en manuele basislijncorrectie, b) ingezoomd rond 2900 cm^{-1}, c) ingezoomd rond 3350 cm^{-1} en d) ingezoomd rond 1000 cm^{-1}

Figuur 4.5 toont telkens in de x-as het golfgetal (cm^{-1}) en in de y-as de absorbantie (-). De rode curve is steeds de AirPLS-basislijn, de groene curve is de AsLS-basislijn, de blauwe curve is het originele spectrum en de zwarte curve stelt het manueel gecorrigeerde spectrum voor. Het spectrum in Figuur 4.5 is louter afkomstig van één staal.

Figuur 4.5.a toont dat op het eerste gezicht de correcties weinig verschillen van de manuele correctie. Daarom worden in deel b, c en d van Figuur 4.5 ingezoomd op enkele belangrijke regio's:

- Figuur 4.5.b zoomt in op het gebied rond 2900 cm^{-1} waar informatie te vinden is omtrent het alkaangedeelte. Hier valt op dat de drie correcties elkaar volgen en vrijwel evenveel verschillen ten opzichte van het originele spectrum.
- Deel c van Figuur 4.5 geeft de meest prominente hydroxylpiek (3350 cm^{-1}) uitvergroot weer. Hierop is te zien dat de AsLS-correctie de andere twee correcties niet volgt. De curven zijn duidelijk van elkaar te onderscheiden.
- Figuur 4.5.d geeft de regio rond 1000 cm^{-1} weer. Hier bevindt zich informatie omtrent primair, secundair en tertiair gebonden hydroxylgroepen. De drie correcties volgen elkaar opnieuw constant in deze regio.

Het vermoeden zou rijzen dat de AsLS-correctie aanleiding zou geven tot minder nauwkeurige resultaten dan de manuele of de AirPLS-correctie aangezien de hydroxylpiek lager wordt ingeschat. Echter is dit niet het geval wanneer de resultaten van Tabel 4.4 worden geanalyseerd. Dit kan als volgt verklaard worden: Elke AsLS-correctie wordt op dezelfde wijze doorgevoerd op een individueel spectrum waardoor elke hydroxylpiek op dezelfde wijze verkleind wordt. Daardoor leert het model hier evengoed uit desondanks het verschil met de andere correctiemethoden.

Hoewel er niet meteen kan worden vastgesteld dat de manuele correctie sterk afwijkt van de automatische correcties, scoort deze manier van corrigeren duidelijk minder (Tabel 4.4). De manuele basislijn werd steeds op dezelfde punten toegepast waardoor geen individuele correctie per spectrum werd voltrokken. Hierdoor is het model minder nauwkeurig. Dit werd reeds afgeleid uit de resultaten in paragraaf 4.1.1.1.

Een manuele basislijncorrectie is dus geen aangewezen manier van werken om een model te trainen daar de basislijn per concentratie iets anders zal liggen. Een individuele optimalisatie per staal zou kunnen helpen. Dit is echter tijdrovend en dus niet de ideale manier van corrigeren.

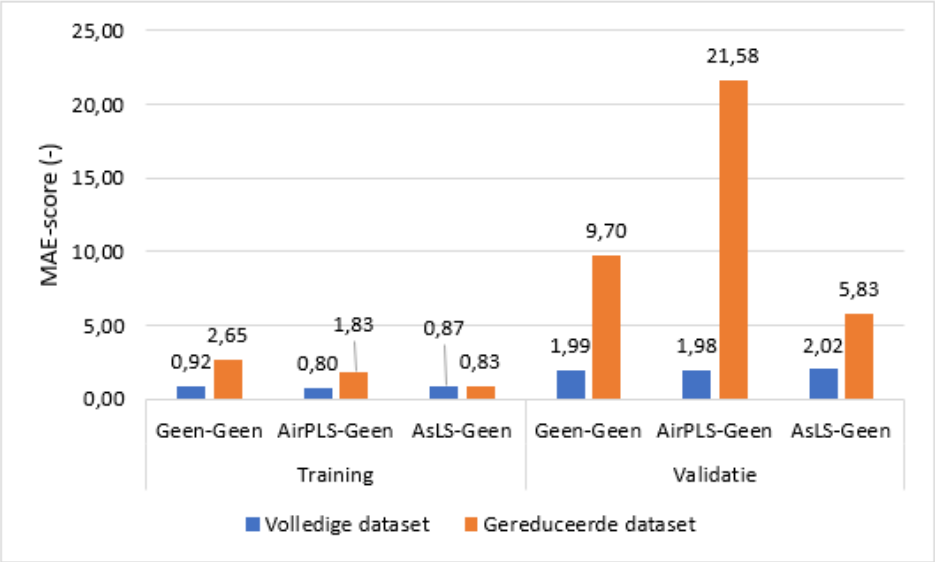
4.1.3 Optimalisatie met hoog gecorreleerde golfgetallen

Er wordt in dit experiment enkel gekeken naar golfgetallengebieden die een hoge correlatie vertonen met de concentraties. Daarbij wordt dus aan datareductie gedaan want enkel de golfgetallen met een correlatie hoger dan een vooraf bepaalde waarde worden nog meegenomen in de verwerking. In deze context wordt er gesproken van een optimalisatie als een aanpassing van het netwerk ervoor zorgt dat de AE- en MAE-waarde lager zijn dan voor deze aanpassing. De vergelijking werd gedaan voor de modellen zonder pre-processing zodanig dat deze geen invloed heeft op de correlaties en omdat daarenboven het effect van de datareductie het grootst is in deze situatie.

Er werd getracht ongeveer 60 % van de golfgetallen te behouden. Daartoe werd gewerkt met golfgetallen die een correlatie hadden die groter was dan 0,4. Deze correlatie werd berekend door de hele dataset aan stalen te nemen en te kijken welke golfgetallen sterk varieerden per concentratie en welke niet. De golfgetallen die het grootste verband vertoonden met de concentratie kregen de hoogste correlatie-waarden en omgekeerd. Op die manier werd gedifferentieerd tussen hoog gecorreleerde golfgetallen en laag gecorreleerde golfgetallen.

Er wordt verwacht dat een vermindering van het aantal golfgetallen ervoor zou zorgen dat het netwerk sneller zou leren en even nauwkeurige inschattingen zou maken zoals in het voorgaande experiment. Door de datareductie zijn immers minder relevante datapunten uit de dataset verwijderd waardoor de grote van de dataset is afgenomen. Dit experiment wordt louter gedaan voor het MPG-glycerol mengsel.

Figuur 4.6 geeft visueel weer wat het effect is van een datareductie op basis van een correlatie-selectie op de trainingsscore en de validatiescore. Het linkergedeelte van Figuur 4.6 geeft de resultaten van de trainingsfase weer en het rechtergedeelte toont de validatieresultaten.



Figuur 4.6: Effect van datareductie op basis van de correlatie op de MAE-scores van training en validatie

Figuur 4.6 toont dat het werken met hoog gecorreleerde waarden globaal gezien geen aanleiding geeft tot een kleinere MAE-waarde in de trainingsfase. Slechts in één situatie is dit wel het geval, namelijk bij een AsLS-basislijncorrectie. Het verschil in resultaat is echter klein (MAE-verschil van 0,04).

Uit Figuur 4.6 volgt ook dat de modellen zonder de reductie, een MAE-validatiescore hebben rond de waarde 2. Na de datareductie wordt de MAE-score beduidend groter. Enigszins kan besloten worden dat de AsLS-basislijncorrectie nog steeds aanleiding geeft tot het laagste resultaat. Dit resultaat is wel significant hoger dan wanneer geen datareductie wordt toegepast. De waarde van 21,58 ± 2,56 bij het 'AirPLS-Geen'-model valt op in Figuur 4.6. Deze waarde is meer dan tien keer groter dan wanneer er geen datareductie wordt doorgevoerd.

De reden waarom de datareductie geen aanleiding geeft tot nauwkeurigere resultaten, kan verklaard worden door het verlies aan informatie die de datareductie met zich meebrengt. De golfgetallen die laag gecorreleerd zijn, bevatten op hun beurt nog informatie waaruit het model kennis put. Door de datareductie is er ook verlies aan kennis waardoor de inschattingen bijgevolg minder nauwkeurig zijn.

Ter vervollediging worden Tabel 4.6 en Tabel 4.7 meegegeven waarin respectievelijk de resultaten van de datareductie op de trainingsfase en de validatiefase worden weerggegeven. Hierbij worden ook de hyperparameters en de standaardafwijking vermeld.

Tabel 4.6: Resultaten van de datareductie op basis van correlatie in de trainingsfase

Basislijncorrectie	Pre-processing	Hyperparameters (epochs - batch-grootte - hidden units)	Trainingsresultaat (MAE-score ± standaardafwijking) (-)
Zonder correctie	Geen	69-1-64	0,92 ± 0,78
AirPLS-basislijncorrectie		333-1-32	0,80 ± 0,59
AsLS-basislijncorrectie		333-1-32	0,87 ± 0,84
Zonder correctie	Geen - Hoge correlatie	69-1-64	2,65 ± 1,32
AirPLS-basislijncorrectie		333-1-32	1,83 ± 0,92
AsLS-basislijncorrectie		333-1-32	0,83 ± 0,96

Tabel 4.7: Resultaten van de datareductie op basis van correlatie in de validatiefase

Basislijncorrectie	Pre-processing	Hyperparameters (epochs - batch-grootte - hidden units)	Validatieresultaat (MAE-score ± standaardafwijking) (-)
Zonder correctie	Geen	69-1-64	1,99 ± 1,76
AirPLS-basislijncorrectie		333-1-64	1,98 ± 1,78
AsLS-basislijncorrectie		333-1-64	2,02 ± 2,02
Zonder correctie	Geen - Hoge correlatie	69-1-64	9,70 ± 6,19
AirPLS-basislijncorrectie		333-1-64	21,58 ± 2,56
AsLS-basislijncorrectie		333-1-64	5,83 ± 2,90

Uit Tabel 4.6 en Tabel 4.7 volgen dezelfde conclusies die getrokken werden op basis van Figuur 4.6. Daarnaast kan globaal gesteld worden dat de spreiding groter is wanneer gewerkt wordt met hooggecorreleerde golfgetallen. Dit wijst op een ongelijke inschatting van de stalen.

Er kan geconcludeerd worden dat een verkleining van de dataset op basis van correlatie, geen aanleiding geeft tot nauwkeurigere inschattingen. Ook laag gecorreleerde golfgetallen dragen informatie die bijdraagt aan de accuraatheid van de modellen.

4.1.4 Tussentijdse conclusie op basis van de binaire experimenten

Uit de binaire experimentenreeks kon geen ideale combinatie van een basislijncorrectie en pre-processing worden vastgesteld. Het werd wel duidelijk dat een manuele basislijncorrectie geen aanleiding gaf tot nauwkeurige resultaten in vergelijking met de andere correctiemethoden. Daarom wordt deze manier van basislijncorrectie ook niet meer meegenomen in de verdere experimenten.

De hyperparametervariaties hebben niet geleid tot een ideale waarde van het aantal epochs en het aantal verborgen eenheden. Wel gaf een batch-grootte van 1 in elk model aanleiding tot het nauwkeurigste resultaat.

Daarnaast bleek het werken met hoog gecorreleerde golfgetallen niet bij te dragen aan de nauwkeurigheid van de inschattingen. Het verkleinen van het aantal golfgetallen heeft aantgetoond dat ook lager gecorreleerde golfgetallen belangrijke informatie bevatten. Het verlies aan deze informatie leidt tot een minder nauwkeurig model.

4.2 Kwantificatie van een ternair mengsel van stabilisatoren

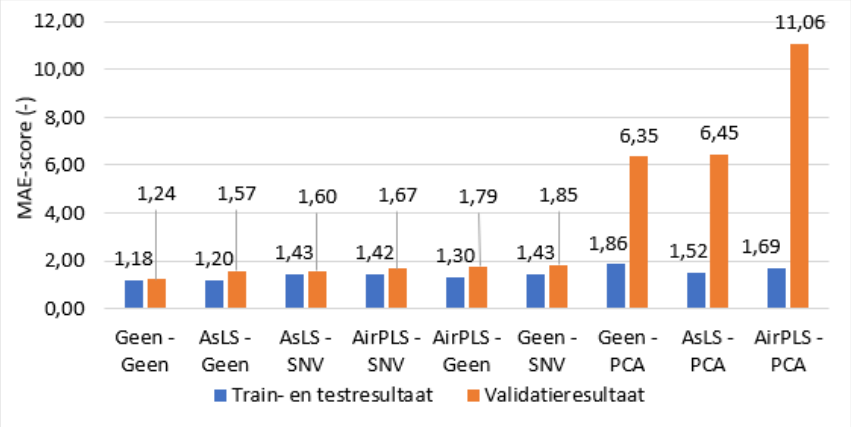
Er werd op basis van vorige experimenten besloten om in deze reeks experimenten als basislijncorrectie: geen, AsLS en AirPLS te gebruiken. Als pre-processing werden: geen, SNV en PCA getest. In deze paragraaf worden modellen besproken die betrekking hebben op ternaire mengsels van stabilisatoren.

4.2.1 Analyse van ternaire en binaire stalen met een binair opgebouwd ternair model

4.2.1.1 Analyse van het ternair model aan de hand van binaire stalen

Vanuit een eerste aanleg werd gepoogd een ternair model op te stellen door middel van de 250 binaire stalen waarbij de derde componenten als 0 % werd beschouwd. De concentraties zijn dus dezelfde als deze voorgesteld in Figuur 4.2.

De binaire mengsels werden dus gebruikt als training- en testdata. Eens deze modellen opgebouwd waren, werden deze gevalideerd met binaire validatiestalen. Wanneer PCA als pre-processing werd gebruikt, werden steeds drie principale componenten gebruikt die samen meer dan 98 % van de variabiliteit van de data bevatten. Dit wordt zo gedaan voor elk ternair model. De resultaten van de meest performante validatiemodellen worden gegeven in Figuur 4.7 en in Tabel 4.8.



Figuur 4.7: Resultaten van test- en validatiefase van binaire stalen

De resultaten die getoond worden in Figuur 4.7, zijn geordend van links naar rechts met toenemende validatiescore. Strikt genomen is het meest linkse model (‘Geen-Geen’) het meest performant en het meest rechtse model, het minst performant (‘AirPLS-PCA’).

Deze resultaten zijn vergelijkbaar met deze bekomen bij het binaire model (paragraaf 4.1.1). De MAE-trainingscores liggen in dezelfde grootteorde. De MAE-validatiescores hebben voor de meeste ternaire modellen absoluut gezien een kleinere waarde dan voor de binaire modellen. Alle modellen, buiten deze waarbij PCA gebruikt werd, hebben een MAE-validatiescore kleiner dan 2. Daarnaast blijkt PCA pre-processing hier eveneens minder geschikt dan SNV of geen pre-processing.

Er kan dus gezegd worden dat de binaire mengsels met een vergelijkbare nauwkeurigheid kunnen bepaald worden door een binair als door een ternair model.

Ook bij de opbouw van een ternair model met binaire stalen, kan op het eerste gezicht niet gezegd worden of een pre-processingsmethode en een basislijncorrectie bijdragen aan de nauwkeurigheid van de kwantificatie en of er een ideale combinatie van een basislijncorrectie en een pre-processing bestaat. Wel kan opnieuw gezegd worden dat PCA pre-processing aanleiding geeft tot minder nauwkeurige modellen.

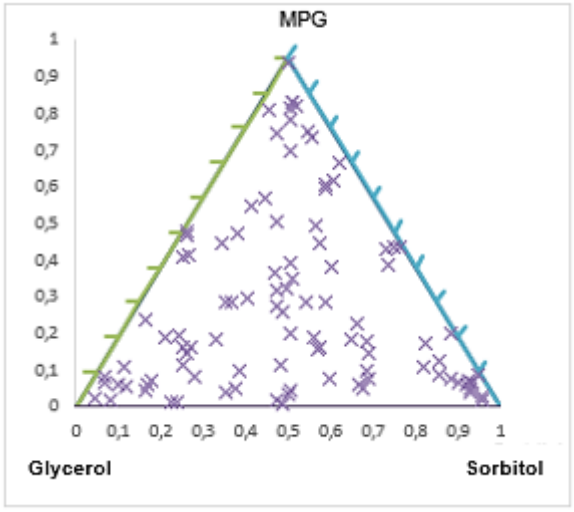
Tabel 4.8 wordt ter vervollediging gegeven en laat toe dezelfde conclusies te trekken als deze getrokken op basis van Figuur 4.7.

Tabel 4.8: Resultaten van test- en validatiefase van binaire stalen

Basislijncorrectie - pre-processing	Hyperparameters (epochs - aantal lagen - hidden units)	Train- en testresultaat (MAE-score ± standaardafwijking) (-)	Validatieresultaat (MAE-score ± standaardafwijking) (-)
Geen - Geen	500-4-64	1,18 ± 0,32	1,24 ± 0,39
AsLS -Geen	350-3-64	1,20 ± 0,33	1,57 ± 0,29
AsLS -SNV	100-3-64	1,43 ± 0,40	1,60 ± 0,43
AirPLS - SNV	350-3-32	1,42 ± 0,40	1,67 ± 0,55
AirPLS -Geen	350-3-32	1,30 ± 0,36	1,79 ± 0,48
Geen -SNV	250-3-32	1,43 ± 0,36	1,85 ± 0,44
Geen - PCA	350-3-32	1,86 ± 0,59	6,35 ± 2,44
AsLS - PCA	2000-3-32	1,52 ± 0,32	6,45 ± 2,87
AirPLS - PCA	400-4-64	1,69 ± 0,79	11,06 ± 4,38

4.2.1.2 Analyse van het ternair model aan de hand van ternaire stalen

De ternaire modellen die opgebouwd zijn in het experiment besproken in paragraaf 4.2.1.1, zijn reeds gevalideerd met binaire stalen. In dit experiment worden diezelfde modellen gevalideerd met ternaire stalen. Daarom werden ternaire mengsels van glycerol, sorbitol en MPG gevoed aan de modellen die voorgesteld zijn in Figuur 4.7 en op basis van die stalen gevalideerd. De concentraties van deze stalen, kunnen worden teruggevonden in het ternaire diagram voorgesteld in Figuur 4.8.

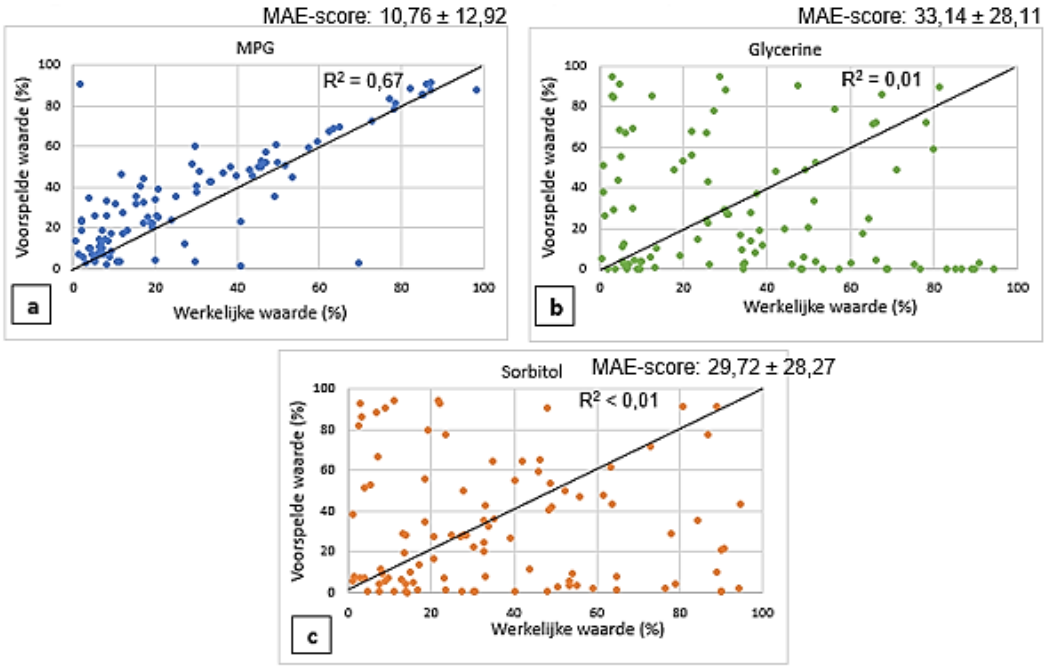


Figuur 4.8: Concentraties van de ternaire stalen, gebruikt voor het exclusief ternair model (in een ternair diagram)

Bij dit experiment werden MAE-validatiescores van ongeveer 25 bekomen.

Als voorbeeld wordt het ‘Geen-Geen’-model weergegeven in Figuur 4.9. De drie deelfiguren zijn afkomstig van de drie samenstellende stabilisatoren van het mengsel. Hierin worden de voorspelde waarden en de effectieve waarden met elkaar vergeleken. Dit model wordt als

voorbeeld aangehaald aangezien deze het meest performant bleek uit voorgaand experiment. Toch haalt dit model een gemiddelde MAE-score van 24,51 ± 18,86 bij de validatie met ternaire stalen.



Figuur 4.9: Resultaten van de validatie van ternaire stalen op het 'Geen-Geen'-model opgebouwd uit binaire stalen; a) MPG, b) Glycerine en c) Sorbitol

Figuur 4.9.a tot en met Figuur 4.9.c hebben allemaal een diagonale lijn in de grafiek. In een ideale situatie zouden de punten op deze curve liggen en is de voorspelde waarde (op de y-as) ook de effectieve waarde (op de x-as). Deze diagonaal stelt met andere woorden een ideaal model voor.

Figuur 4.9.a toont dat er vaak een te hoge inschatting is van de hoeveelheid MPG maar dat deze inschattingen enigszins nabij de diagonaal liggen. De MAE-score voor MPG-bepaling bedraagt 10,76 ± 12,92. Ook wordt voor MPG de grootste R^2-waarde (0,67) gevonden van alle componenten.

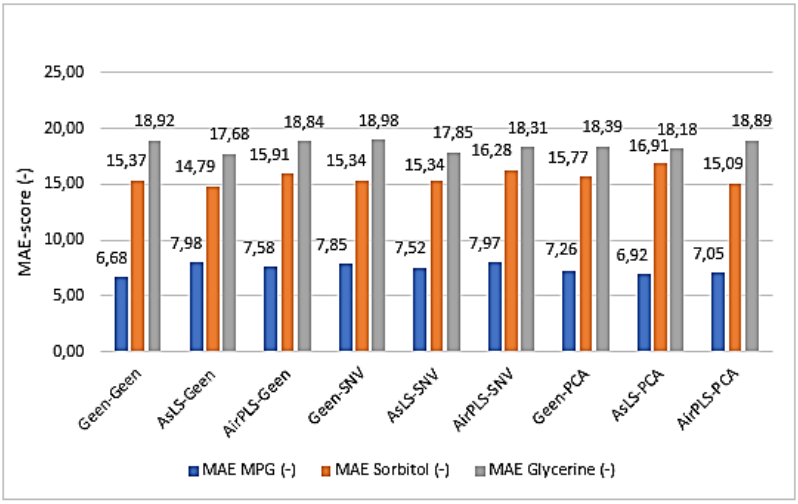
Voor sorbitol en glycerine (Figuur 4.9.b en Figuur 4.9.c) is de inschatting niet in overeenstemming te brengen met de effectieve waarde en is er een grote spreiding. Respectievelijk wordt een MAE-waarde van 29,72 ± 28,27 en 33,14 ± 28,11 gevonden voor sorbitol en glycerine. Zoals Figuur 4.9.b en Figuur 4.9.c tonen, liggen de punten verspreid over het gehele concentratiebereik. Dit verklaart ook de grote standaardafwijkingen.

Dit experiment toont aan dat ternaire stalen in de trainingsdataset noodzakelijk zijn voor de inschatting van ternaire stalen door een ternair model.

4.2.2 Exclusief ternair model

In dit experiment werd een ternair model opgebouwd met een ternaire dataset. Deze dataset werd gebruikt voor de trainings- en testfase en omvat 100 stalen. De concentraties van deze stalen, kunnen worden teruggevonden in het ternaire diagram voorgesteld in Figuur 4.8.

De laagste trainings- en testresultaten worden weergegeven in Figuur 4.10 en in Tabel 4.9 waarin deze gerangschikt worden per pre-processingsmethode.



Figuur 4.10: Visuele weergave van de resultaten van de trainings- en testfase van ternaire stalen in een exclusief ternair model

Tabel 4.9: Resultaten van de trainings- en testfase van ternaire stalen in een exclusief ternair model

Basislijncorrectie - pre-processing	Hyperparameters (epochs - hidden units - aantal lagen)	MAE MPG (-)	MAE sorbitol (-)	MAE glycerol (-)
Geen - Geen	250-32-4	6,68 ± 6,83	15,37 ± 13,17	18,92 ± 14,50
AsLS - Geen	350-64-4	7,98 ± 7,23	14,79 ±12,20	17,68 ± 14,82
AirPLS - Geen	250-32-3	7,58 ± 8,13	15,91 ± 12,04	18,84 ± 14,41
Geen - SNV	100-64-3	7,85 ± 7,85	15,34 ± 13,01	18,98 ± 13,49
AsLS - SNV	250-128-3	7,52 ± 7,04	15,34 ± 11,88	17,85 ± 14,04
AirPLS - SNV	125-64-3	7,97 ± 7,23	16,28 ± 11,84	18,31 ± 14,75
Geen - PCA	150-64-3	7,26 ± 6,50	15,77 ± 12,42	18,39 ± 13,75
AsLS - PCA	100-32-3	6,92 ± 8,06	16,91 ± 12,00	18,18 ± 13,61
AirPLS - PCA	150-128-3	7,05 ± 6,80	15,09 ± 13,09	18,89 ± 13,69

Uit Figuur 4.10 en Tabel 4.9 volgt dat de inschatting van MPG opnieuw beter gaat dan die van sorbitol en glycerine. Ook hier geldt net zoals in paragraaf 4.2.1, dat een model zonder pre-processing en zonder basislijncorrectie de laagste MAE-resultaten geeft voor MPG.

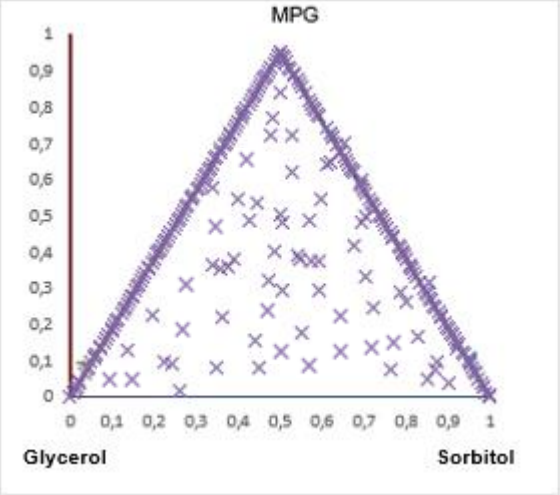
Figuur 4.10 en Tabel 4.9 geven weer dat de resultaten verkregen met PCA pre-processing in dit experiment, wel vergelijkbaar zijn met de andere modellen en soms aanleiding geven tot een lagere MAE-waarde dan een ander model. Deze conclusie dient echter genuanceerd te worden aangezien het hier enkel gaat over de trainings- en testfase.

De inschattingen van de verschillende stabilisatoren zijn in elk model onvoldoende nauwkeurig, waarbij vooral de inschattingen van sorbitol en glycerol opmerkelijk hoger zijn dan de inschatting van MPG. De resultaten zijn reeds accurater dan deze verkregen in het experiment met een binair opgebouwd ternair model (paragraaf 4.2.1) maar nog steeds ontoerijkend. De reden hiervoor is dat de trainingset te klein is (slechts 100 stalen). Ook de extreme grenzen, verkregen door binaire stalen in de trainingset, ontbreken in dit experiment.

In een volgende set experimenten wordt getracht deze MAE-waarden te laten dalen. Daartoe wordt een eerste experiment opgezet met zowel binaire als ternaire stalen ter opbouw van een netwerk.

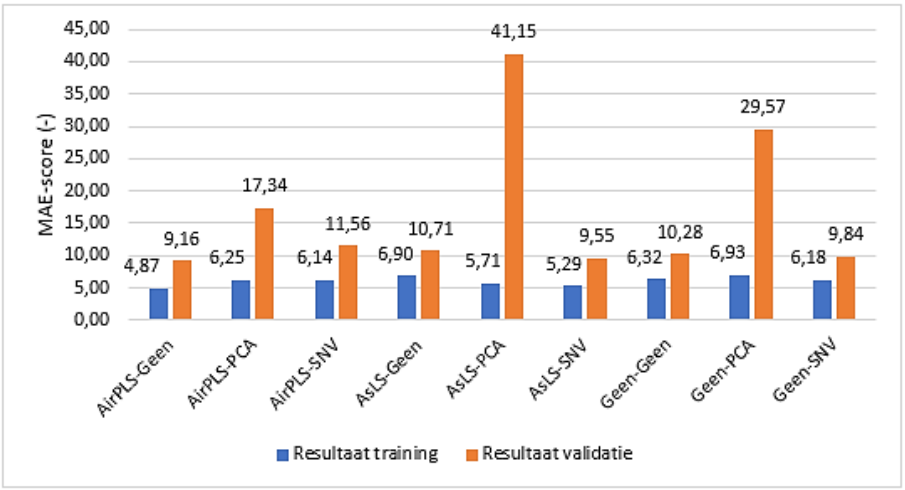
4.2.3 Ternair model opgebouwd met binaire en ternaire stalen

Dit ternair model wordt opgebouwd met 259 trainingstalen, 65 teststalen en 56 validatiestalen. Deze drie sets bevatten zowel binaire als ternaire stalen. Het idee achter dit model is dat naarmate er meer stalen voorhanden zijn, de inschattingen nauwkeuriger worden door een intensievere training. De concentraties van de gebruikte stalen, kunnen worden teruggevonden in het ternaire diagram in Figuur 4.11.



Figuur 4.11: Concentraties van de binaire en ternaire stalen, gebruikt voor het ternair model (in een ternair diagram)

De meest accurate validatiemodellen worden gegeven in Figuur 4.12 waarbij deze gegroepeerd zijn per basislijncorrectie. Hierbij wordt ook telkens het trainingsresultaat meegegeven.



Figuur 4.12: Gemiddelde resultaten van de validatiemodellen met de laagste MAE-score

Figuur 4.12 toont dat het 'AirPLS-Geen'-model aanleiding geeft tot het laagste validatieresultaat en ook het trainingsresultaat is het laagst. Daarnaast toont Figuur 4.12 dat ook het 'AsLS-SNV'-model en 'Geen-SNV'-model lagere MAE-waarden hebben dan de andere modellen.

Er valt ook op in Figuur 4.12 dat de PCA-modellen bij elke basislijncorrectie aanleiding geven tot de hoogste validatiescores. Dit werd ook al aangehaald in de vorige experimenten. De verschillen tussen de PCA-modellen en de andere zijn groot en daarom kan gezegd worden

dat deze pre-processingsmethode niet bijdraagt aan de nauwkeurigheid van de kwantificatie. Tabel 4.10 verduidelijkt deze bevindingen.

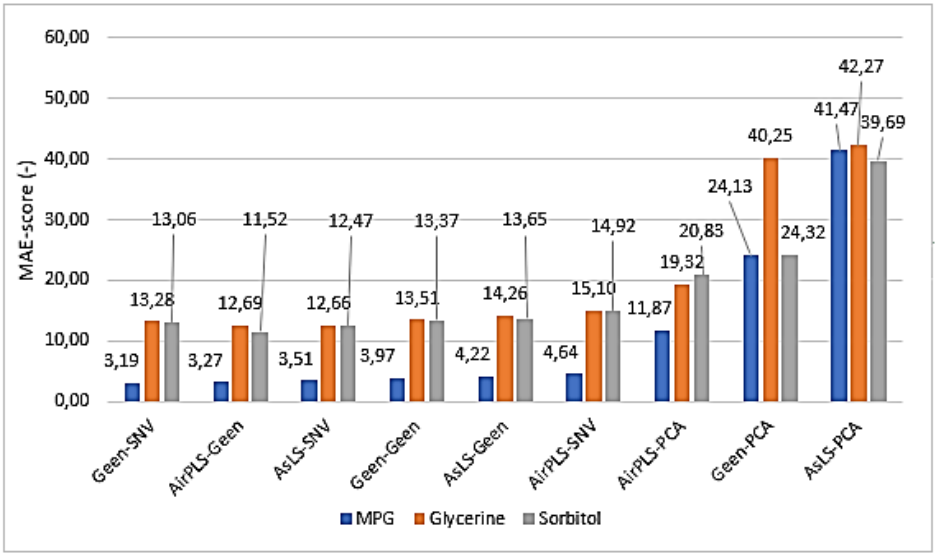
Tabel 4.10: Gemiddelde resultaten van de nauwkeurigste validatiemodellen

Basislijncorretie - pre-processing	MAE-score training ± standaardafwijking (-)	MAE-score validatie ± standaardafwijking (-)
AirPLS-Geen	4,87 ± 2,21	9,16 ± 4,19
AirPLS-PCA	6,25 ± 2,49	17,34 ± 3,91
AirPLS-SNV	6,14 ± 2,27	11,56 ± 4,89
Geen-Geen	6,32 ± 2,52	10,28 ± 4,47
Geen-PCA	6,93 ± 2,27	29,57 ± 7,56
Geen-SNV	6,18 ± 2,58	9,84 ± 4,70
AsLS-Geen	6,90 ± 2,51	10,71 ± 4,59
AsLS-PCA	5,71 ± 2,48	41,15 ± 1,08
AsLS-SNV	5,29 ± 2,35	9,55 ± 4,27

Tabel 4.10 geeft weer dat de spreidingen voor de training allemaal rond de waarde 2,50 schommelen. Bij de validatie heeft het ‘Geen-PCA’-model de grootste spreiding namelijk 7,56 en het ‘AsLS-PCA’-model de kleinste spreiding met een waarde van 1,08.

Opnieuw zijn de verschillen in MAE-scores (buiten deze van de PCA-modellen) klein en kan daaruit niet meteen een sluitende conclusie worden getrokken omtrent een ideale combinatie van een basislijncorrectie en een pre-processing. Daarom wordt gekeken naar de inschattingen voor elke component individueel. De voorgaande experimenten toonden reeds aan dat MPG het nauwkeurigst kan worden ingeschat ten opzichte van de andere stabilisatoren.

Figuur 4.13 zoomt in op de validatieresultaten van de inschattingen voor de binaire en ternaire mengsels. De modellen zijn geordend op basis van de inschattingen voor MPG.



Figuur 4.13: Validatieresultaten van de inschattingen van de afzonderlijke stabilisatoren van de nauwkeurigste ternaire modellen

Figuur 4.13 geeft weer de inschatting van MPG met het ‘Geen-SNV’-model het nauwkeurigst is. Het blijkt dat deze inschatting ongeveer 13 keer dichter bij de werkelijke waarde zit in vergelijking met het ‘AsLS-PCA’-model met betrekking tot MPG. Dit model heeft de grootste validatiescore.

Ook hier valt op dat de inschattingen met behulp van de PCA-modellen minder goed zijn dan die van de andere modellen. Alle andere modellen die geen gebruik maken van PCA pre-processing, scoren ongeveer hetzelfde op vlak van validatie.

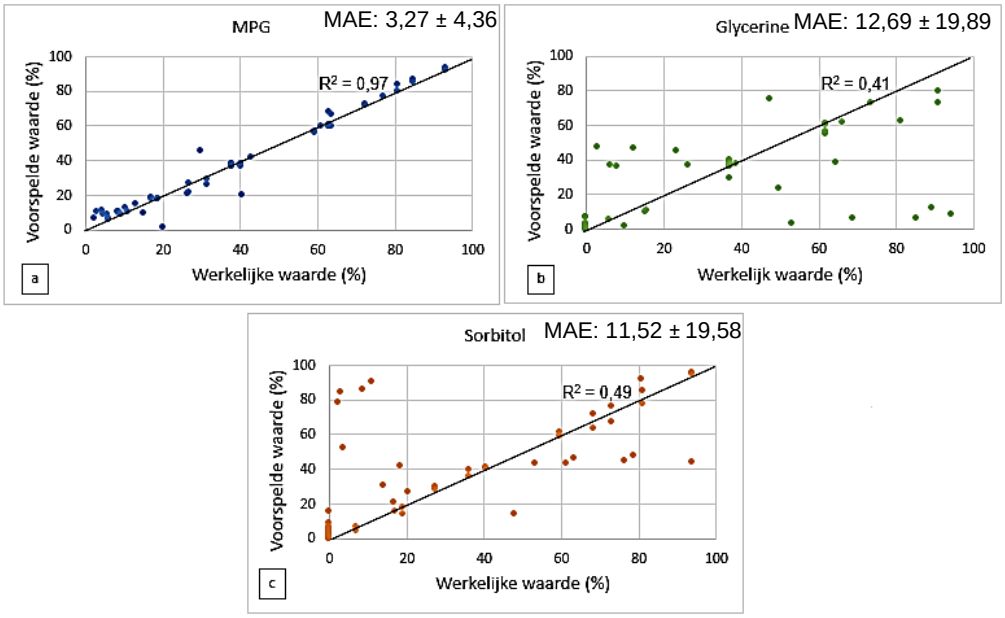
De inschatting van MPG is ook hier het meest accuraat in vergelijking met de andere componenten. Sorbitol en glycerine worden telkens in dezelfde mate van nauwkeurigheid ingeschat. Deze inschattingen zijn echter nog niet nauwkeurig genoeg om deze praktisch te gaan implementeren.

De waarde van de MAE-scores van de verschillende solventen met standaardafwijking worden ter volledigheid gegeven in Tabel 4.11. Algemeen wordt vastgesteld dat de spreidingen relatief groot zijn tegenover de gemiddelde MAE-score. Een ideaal model schat de drie stabilisatoren ongeveer gelijk in en heeft bijgevolg een kleine spreiding. Er is dus nog ruimte voor verbetering.

Tabel 4.11: Trainings- en validatieresultaten van de afzonderlijke stabilisatoren van de ternaire modellen, met aanduiding van het model met de laagste (groen) en hoogste (oranje) MAE-waarde

Basislijncorretie - pre-processing	Trainingsfase (MAE-score ± standaardafwijking) (-)			Validatiefase (MAE-score ± standaardafwijking) (-)		
	MPG	Glycerine	Sorbitol	MPG	Glycerine	Sorbitol
Geen-SNV	2,53 ± 2,36	8,14 ± 14,41	7,86 ± 14,90	3,19 ± 4,06	13,28 ± 21,17	13,06 ± 21,30
AirPLS-Geen	1,78 ± 1,82	6,81 ± 13,14	6,01 ± 12,86	3,27 ± 4,36	12,69 ± 19,89	11,52 ± 19,58
AsLS-SNV	1,97 ± 1,73	7,10 ± 13,45	6,79 ± 13,60	3,51 ± 4,58	12,66 ± 21,06	12,47 ± 21,43
Geen-Geen	2,78 ± 2,48	8,43 ± 14,17	7,75 ± 14,58	3,97 ± 4,10	13,51 ± 20,54	13,37 ± 21,41
AsLS-Geen	3,36 ± 2,61	8,87 ± 15,66	8,48 ± 15,37	4,22 ± 3,85	14,26 ± 22,71	13,65 ± 23,29
AirPLS-SNV	2,96 ± 2,25	8,08 ± 13,40	7,38 ± 12,49	4,64 ± 4,00	15,10 ± 19,20	14,92 ± 19,46
AirPLS-PCA	2,74 ± 2,53	8,20 ± 14,58	7,82 ± 14,48	11,87 ± 9,17	19,32 ± 19,63	20,83 ± 21,41
Geen-PCA	3,74 ± 2,64	8,30 ± 14,32	8,76 ± 14,60	24,13 ± 17,58	40,25 ± 23,20	24,32 ± 16,11
AsLS-PCA	2,22 ± 2,25	7,72 ± 14,24	7,19 ± 14,41	41,47 ± 28,95	42,27 ± 23,09	39,69 ± 26,01

Een overzicht van de inschattingen van het model ten opzichte van de werkelijke waarde voor het ‘AirPLS-Geen’-model worden gegeven in Figuur 4.14. Deel a van deze figuur vertegenwoordigt de inschatting van MPG, deel b en c staan respectievelijk voor de inschatting van glycerine en sorbitol. De diagonale lijn staat voor het ideale model.



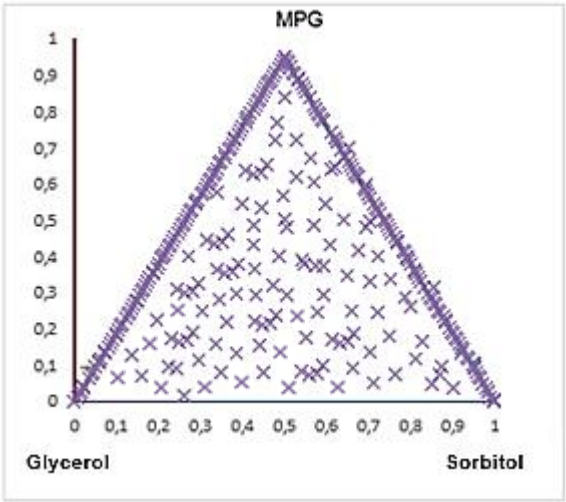
Figuur 4.14: Visuele weergave van de voorspelde waarde ten opzichte van de werkelijke waarde van het ‘AirPLS-Geen-model; a) MPG, b) Glycerine en c) Sorbitol

Zoals uit Figuur 4.13 besloten werd, is de inschatting van de MPG-concentratie (Figuur 4.14.a) het nauwkeurigst van de drie inschattingen. De inschattingen van glycerine en sorbitol respectievelijk voorgesteld door Figuur 4.14.b en Figuur 4.14.c, zijn reeds beter dan deze door het volledige ternaire model (in Figuur 4.9) doch opmerkelijk meer verspreid dan de inschatting van MPG.

In een volgend experiment zullen er 100 extra ternaire stalen aan de dataset worden toegevoegd opdat de inschatting en de spreiding zouden verbeteren.

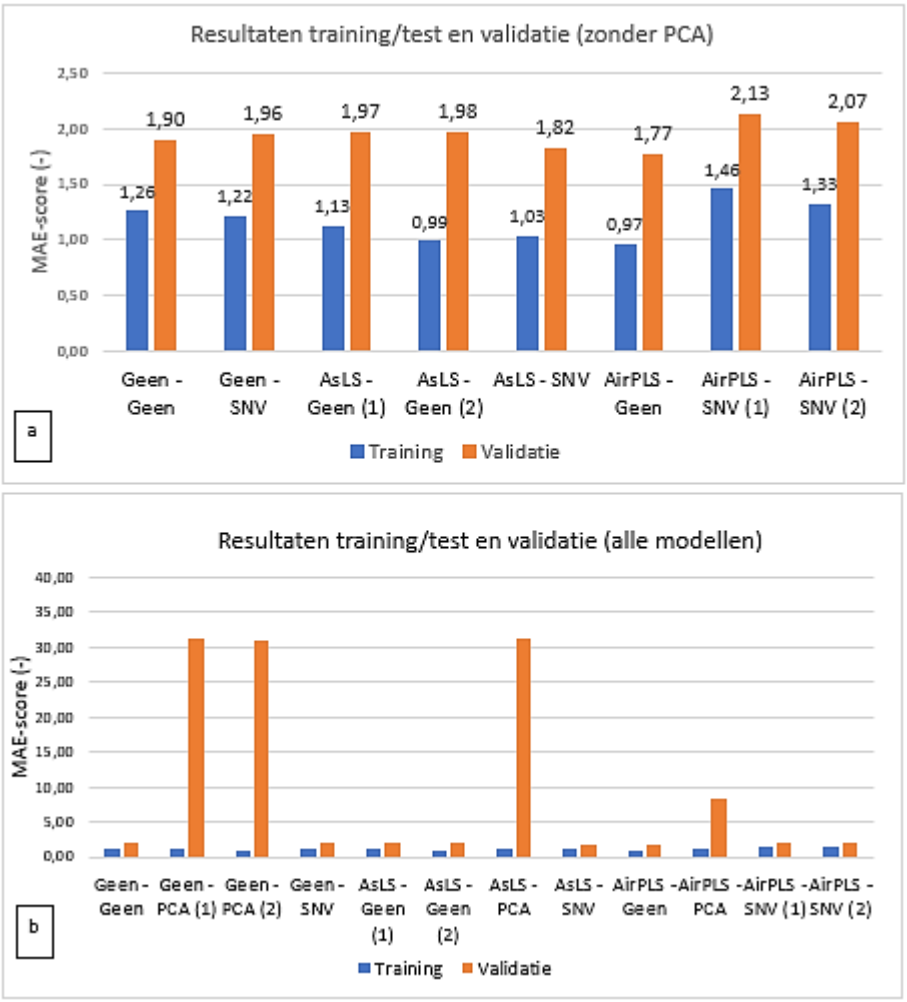
4.2.4 Uitbreiding van het aantal ternaire stalen

In deze fase van het experiment worden 100 nieuwe ternaire stalen met gekende concentraties aangemaakt en geanalyseerd. De dataset heeft nu een omvang van: 399 training- en teststalen (waarvan respectievelijk 319 voor het trainen en 80 voor het testen) alsook 102 stalen ter validatie. De binaire stalen en ternaire stalen zijn in deze set ongeveer gelijk verdeeld alsook de zuivere grondstoffen behoren tot de dataset. De concentraties van deze stalen, worden gegeven in het ternair diagram in Figuur 4.15.



Figuur 4.15: Ternair diagram van de binaire en ternaire stalen na toevoeging van 100 extra ternaire stalen

Het effect van de toevoeging van de extra ternaire stalen op het model wordt gegeven in Figuur 4.16. De modellen in Figuur 4.16 zijn gegroepeerd per basislijncorrectie. De waarden op Figuur 4.16.a worden herhaald in Tabel 4.12 met de desbetreffende standaardafwijking.



Figuur 4.16: Resultaten van de trainings-testfase en validatiefase van de modellen na toevoeging van 100 extra ternaire stalen; a) Zonder PCA pre-processing b) Alle modellen

Figuur 4.16.a toont dat de gemiddelde inschattingen van de modellen nauwkeuriger zijn dan zonder de extra ternaire stalen (getoond in Figuur 4.12). Het is duidelijk dat het verhogen van de trainingset met 100 extra ternaire stalen, aanleiding geeft tot accuratere inschattingen. Het toevoegen van deze extra 100 stalen, had geen significante invloed op de tijd die nodig was om het model op te bouwen.

Op Figuur 4.16.a en in Tabel 4.12 is te zien dat de trainingsresultaten tussen 0,97 ± 0,25 en 1,46 ± 0,42 liggen en de validatieresultaten liggen tussen 1,77 ± 0,43 en 2,13 ± 0,32. Uit Figuur 4.16.a en Tabel 4.12 blijkt dat het 'AirPLS-Geen'-model (MAE-validatiescore: 1,77 ± 0,43) het model is die de nauwkeurigste inschattingen levert en dat het 'AirPLS-SNV'-model (MAE-validatiescore: 2,13 ± 0,32) het minste presteert. Hierbij worden reeds de PCA-modellen buiten beschouwing gelaten. De overige modellen hebben ongeveer een MAE-validatiescore rond de waarde 1,95. In het algemeen kan gezegd worden dat de modellen onderling niet veel verschillen qua validatiescore.

Dit resultaat illustreert overigens dat niet per se een basislijncorrectie op zich aanleiding geeft tot het meest performante model, maar wel degelijk de combinatie van basislijncorrectie en pre-processingsmethode.

In Figuur 4.16 is soms de notatie '(1)' of '(2)' te zien. Dit is afkomstig van het feit dat er van eenzelfde combinatie van basislijncorrectie en pre-processing verschillende modellen getest

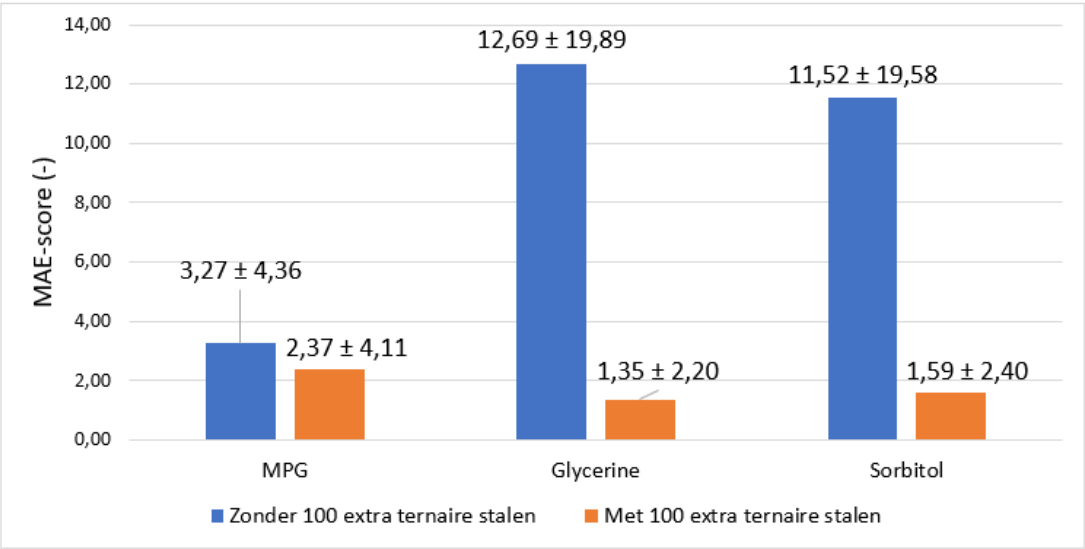
en gevalideerd zijn. Deze modellen hadden betrekking op andere hyperparameters, waardoor er een onderscheid diende gemaakt te worden via '(1)' en '(2)'.

Figuur 4.16.b toont dat in overeenstemming met voorgaande experimenten, PCA pre-processing aanleiding geeft tot een hogere validatiescore. Dit hoewel de MAE-waarde van de training wel in dezelfde lijn ligt van de andere modellen. Ook hier blijkt, na toevoeging van 100 extra ternaire stalen, dat PCA niet geschikt is als pre-processing. Op de reden waarom PCA niet geschikt blijkt te zijn, wordt in paragraaf 4.2.6 verder ingegaan.

Tabel 4.12: Gemiddelde trainings- en validatieresultaten van de ternaire modellen met extra ternaire stalen

Basislijncorrectie - pre-processing	Trainings- en testfase	Validatiefase
	Gemiddelde MAE-score ± standaardafwijking (-)	Gemiddelde MAE-score ± standaardafwijking (-)
Geen - Geen	1,26 ± 0,37	1,90 ± 0,40
Geen - PCA (1)	1,19 ± 0,27	31,11 ± 11,02
Geen - PCA (2)	1,01 ± 0,31	30,96 ± 10,71
Geen - SNV	1,22 ± 0,32	1,96 ± 0,39
AsLS - Geen (1)	1,13 ± 0,18	1,97 ± 0,34
AsLS - Geen (2)	0,99 ± 0,14	1,98 ± 0,29
AsLS - PCA	1,10 ± 0,29	31,16 ± 14,37
AsLS - SNV	1,03 ± 0,38	1,82 ± 0,32
AirPLS - Geen	0,97 ± 0,25	1,77 ± 0,43
AirPLS - PCA	1,16 ± 0,26	8,22 ± 1,38
AirPLS - SNV (1)	1,46 ± 0,42	2,13 ± 0,32
AirPLS - SNV (2)	1,33 ± 0,41	2,07 ± 0,42

De resultaten in Figuur 4.16 en in Tabel 4.12, tonen aan dat 100 extra ternaire stalen een significant verschil uitmaken. Om dit te verduidelijken wordt het meest performante model uit dit experiment, namelijk het 'AirPLS-Geen'-model vergeleken met het 'AirPLS-Geen'-model uit vorig experiment (dus zonder de extra 100 stalen). Dit wordt gedemonstreerd in Figuur 4.17.

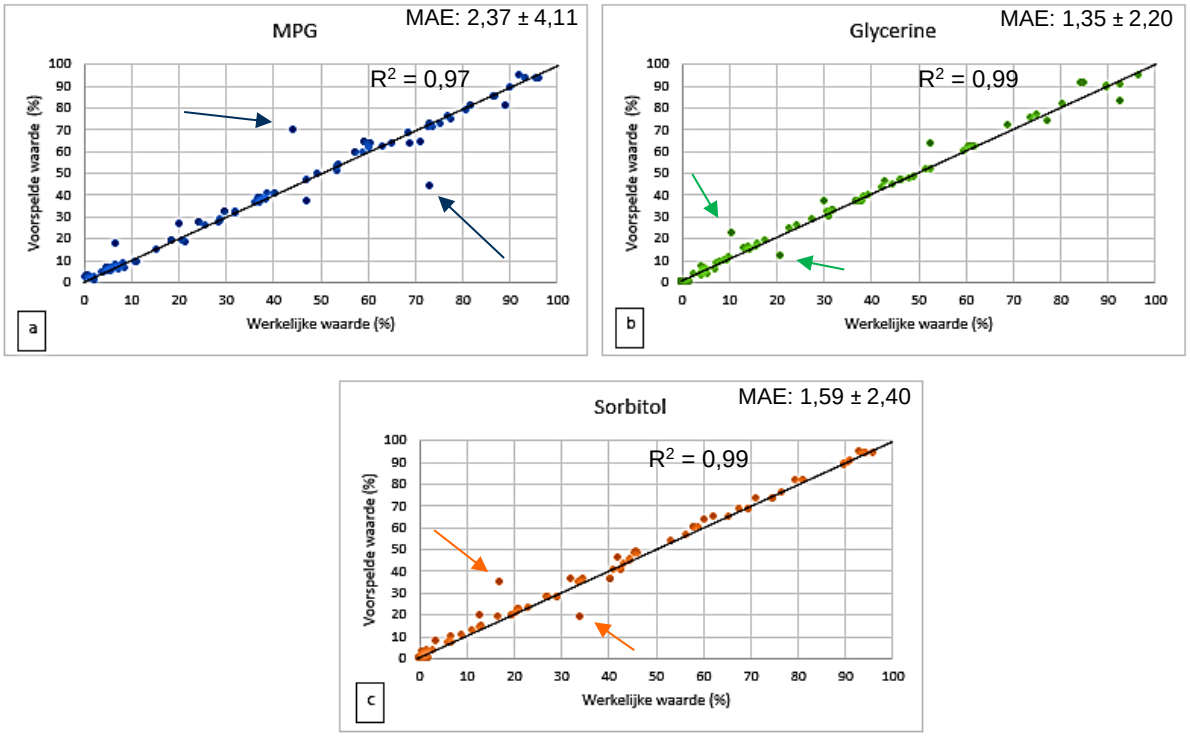


Figuur 4.17: Effect van de toevoeging van 100 extra ternaire stalen voor de inschatting van sorbitol, MPG en glycerine

Figuur 4.17 geeft weer dat MPG beter ingeschat wordt met ongeveer 1 MAE-waarde verschil na het toevoegen van 100 extra ternaire stalen. De grotere verschillen bevinden zich bij de inschatting van glycerine en sorbitol. Daar zakken de MAE-waarden met ongeveer 10 eenheden en worden deze nu beter ingeschat dan MPG. Dit was omgekeerd in het geval zonder de extra 100 ternaire stalen.

Het effect van het uitbreiden van de dataset wordt zo merkbaar: Door het vergroten van de dataset, wordt het algoritme intensiever getraind en kan deze daardoor accuratere inschattingen maken.

Figuur 4.18 geeft een overzicht van de inschattingen van glycerine, sorbitol en MPG voor het 'AirPLS-Geen'-model uit het laatste experiment, waarbij de werkelijke en ingeschatte waarde tegenover elkaar worden uitgezet.



Figuur 4.18: Visuele weergave van de voorspelde waarde ten opzichte van de werkelijke waarde van het 'AirPLS-Geen'-model; a) MPG, b) Glycerine en c) Sorbitol

Figuur 4.18 toont, in tegenstelling tot Figuur 4.9 en Figuur 4.14, dat de inschattingen van glycerine, sorbitol en MPG die van het ideale model benaderen (diagonale lijn in Figuur 4.18). MPG werd in voorgaande experimenten vrij accuraat ingeschat maar nog nooit zo accuraat als in dit experiment. De meetpunten vertonen voor de MPG-inschatting een lineair karakter met een R^2-waarde van 0,97.

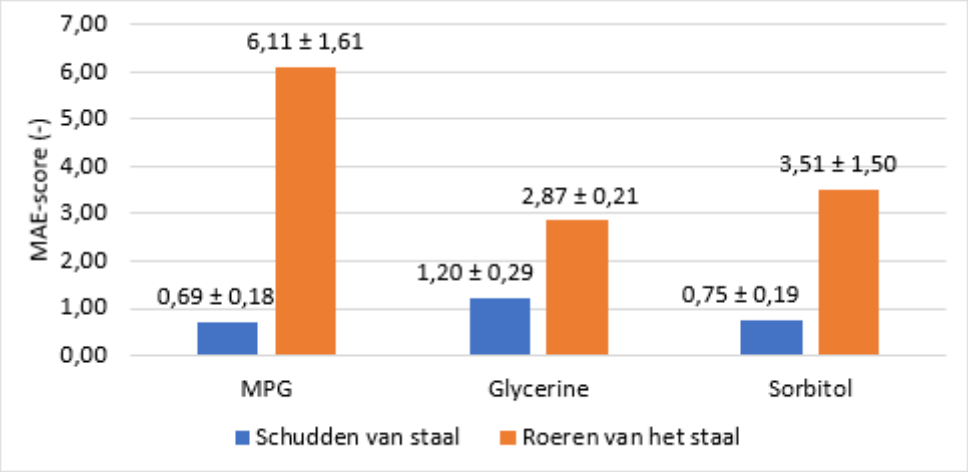
Het grootste verschil zit in de inschatting van de componenten sorbitol en glycerine. Waar er nog geen onderling verband kenbaar was in Figuur 4.9 en Figuur 4.14, is dit wel het geval in Figuur 4.18. De datapunten liggen in Figuur 4.18.b tot Figuur 4.18.c niet langer verspreid maar gelokaliseerd rond de diagonale lijn. Er wordt voor beide componenten een lineair verband vastgesteld tussen de voorspelde waarde en de werkelijke waarde, waarbij voor zowel sorbitol als glycerine een R^2-waarde van 0,99 werd gevonden. Dit wijst op het feit dat dit model het ideale karakter benadert. De toevoeging van extra ternaire stalen leidt dus tot een nauwkeuriger model.

Op Figuur 4.18.a zijn er twee datapunten te zien die meer afwijken van de diagonale lijn dan de andere datapunten. Deze worden aangeduid met een pijl. Dit wordt opgemerkt tussen 45 % en 50 % MPG (werkelijke waarde). Ook in Figuur 4.18.c worden de afwijkingen vastgesteld van dezelfde stalen en worden ook aangeduid met een pijl. Voor de inschatting van glycerine is dit minder te merken maar ook hier worden diezelfde stalen gemarkeerd met een pijl. Dit is niet verwonderlijk daar de inschatting van glycerine het nauwkeurigst verliep (Figuur 4.17).

Deze afwijkingen zijn ofwel te wijten aan de sterk gelijkende spectra van de stalen waardoor de inschatting moeilijker verloopt, ofwel door foutieve metingen. In het geval dat de sterk gelijkende spectra aan de basis liggen van de inschattingsfout, kan dit worden weggewerkt door het model te trainen met extra stalen.

4.2.5 Effect van roeren of schudden op de bepaling van de concentraties

Bij de opname van de spectra werd gemerkt dat wanneer de stalen hard geschud werden een heldere vloeistof bekomen werd. Wanneer daarentegen grondig geroerd wordt met de stalen, bleef een ondoorzichtige vloeistof over. Het belang van het goed mengen werd duidelijk wanneer een vergelijking gemaakt werd met eenzelfde staal indien dit geroerd werd tegenover geschud werd. Figuur 4.19 geeft dit effect weer op het 'AirPLS-Geen'-model uit het vorige experiment.



Figuur 4.19: Effect van roeren en schudden op de inschatting van de stabilisatoren voor het 'AirPLS-Geen'-model

Figuur 4.19 geeft weer dat de grootste impact van roeren van het staal zichtbaar is bij MPG. Hier wordt een verschil waargenomen van meer dan 5 MAE-eenheden. Bij glycerine en sorbitol is het verschil kleiner met ongeveer 2 MAE-eenheden. Toch kan gezegd worden dat krachtig schudden ook hier de laagste resultaten geeft.

Dit kan verklaard worden door de grote verschillen in viscositeit van de stabilisatoren. Glycerine is het meest viskeus, gevolgd door respectievelijk sorbitol en MPG. Om een homogene menging te verkrijgen, moet het staal krachtig geschud worden. Indien dit niet gedaan wordt, bestaat het risico dat er een niet-representatief IR-spectrum wordt opgenomen en bijgevolg een foute inschatting gemaakt wordt door het model. Dit brengt met zich mee dat voor een hoge reproduceerbaarheid en herhaalbaarheid, het staal steeds krachtig geschud zal moeten worden om zo het algoritme in staat te stellen om correct te leren uit de spectra. (51,52,54)

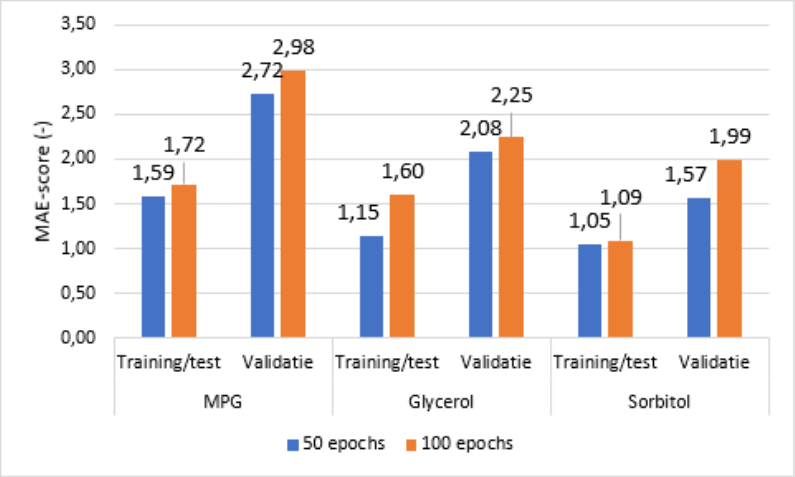
4.2.6 Waarom PCA niet geschikt is als pre-processingsmethode

Uit de voorgaande experimenten bleek telkens dat PCA als pre-processingstechniek aanleiding geeft tot hogere validatiescores dan de andere modellen. Dit hoewel de trainingsfase steeds veelbelovende resultaten gaf. Een voorbeeld daarvan kan gevonden worden in Figuur 4.16.b.

In een eerste aanleg werd het aantal epochs en het aantal verborgen eenheden verkleind om te kijken of overfitting aan de basis lag. Dit bleek niet het geval te zijn. Ook een variatie van het aantal PC's bleek geen oplossing te kunnen bieden. Daarom werd meer ingezoomd op de manier van werken bij PCA.

De PC's die gevormd worden, zijn afhankelijk van de dataset waaruit ze gemaakt worden. Die afhankelijkheid vertaalt zich in een lineaire combinatie. Wanneer de trainings- en testset en de validatieset apart omgezet worden in PC's, mist de onderlinge coherentie tussen de twee sets aan PC's. Dit komt omdat andere lineaire combinaties gemaakt worden in de twee aparte datasets. (103,104)

Om dit aan te tonen, wordt de gehele dataset (training/test en validatie) samen omgezet in PC's en pas nadien gesplitst. Het resultaat van dit experiment is te zien in Figuur 4.20. De gemiddelde MAE-waarden met de daarbij horende standaardafwijkingen worden gegeven in Tabel 4.13. (103,104)



Figuur 4.20: Effect van PCA pre-processing voor splitsing in training- en testset en validatieset voor het 'AirPLS-Geen'-model

Tabel 4.13: MAE-scores van de training- en validatiefase voor de individuele stabilisatoren bij een PCA pre-processing uit eenzelfde set van data

AirPLS-Geen	MPG (MAE-score ± standaardafwijking) (-)		Glycerol (MAE-score ± standaardafwijking) (-)		Sorbitol (MAE-score ± standaardafwijking) (-)	
Epochs (-)	Training	Validatie	Training	Validatie	Training	Validatie
50	1,59 ± 1,65	2,72 ± 2,15	1,15 ± 1,18	2,08 ± 1,99	1,05 ± 1,00	1,57 ± 1,09
100	1,72 ± 1,51	2,98 ± 2,09	1,60 ± 1,29	2,25 ± 1,95	1,09 ± 1,04	1,99 ± 1,17

Figuur 4.20 en Tabel 4.13 geven weer dat de validatie, zowel voor 50 epochs als 100 epochs, vergelijkbare resultaten geeft zoals voor de andere modellen in paragraaf 2.4.4. Hierbij wordt een MAE-waarde kleiner dan 2 waargenomen bij de training en een MAE-score tussen 2 en 3 eenheden bij de validatie.

Ook MPG wordt minder goed ingeschat dan de andere twee componenten en dit is eveneens in overeenstemming met de andere modellen in paragraaf 2.4.4. De spreiding op de inschattingen zijn gelijkaardig aan de spreidingen in het experiment van paragraaf 2.4.4. Dit toont aan dat het PCA-model de inschattingen voor de verschillende componenten gelijkaardig voltrekt. Het probleem van de connectie van de PC's is daarmee aangetoond.

Hieruit kan besloten worden dat het niet mogelijk is om PCA te gebruiken als pre-processing voor dit deep learning model. Het is immers de bedoeling dat het training/testen en de validatie in aparte fasen verlopen waarbij ook de pre-processing apart voltrokken wordt. De stalen voor de validatie worden namelijk niet tezamen met de trainingstalen en teststalen omgezet in PC's. Immers is het anders onmogelijk om onbekende stalen achteraf te analyseren met een statisch geworden deep learning model. In andere toepassingen waarbij er geen statisch gemaakt deep learning model is, kan PCA pre-processing wel geschikt zijn.

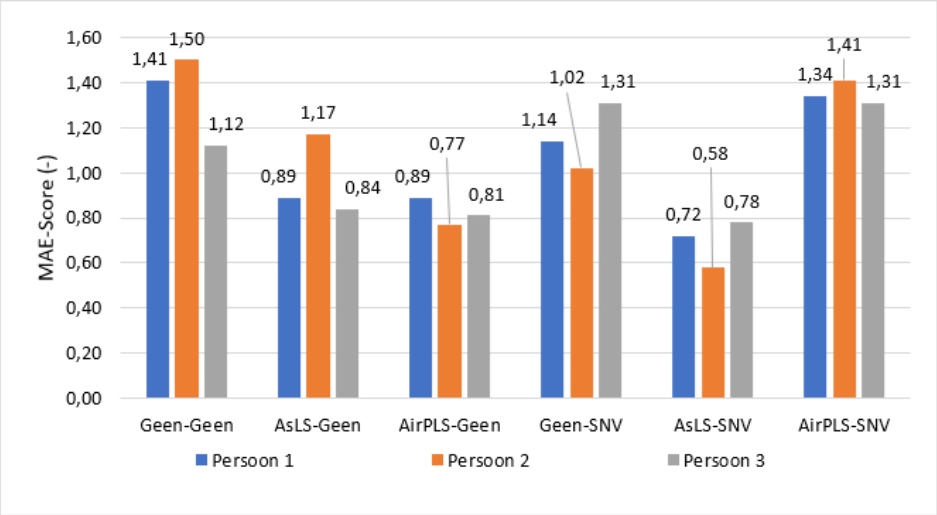
4.2.7 Intralaboratorium-reproduceerbaarheid en herhaalbaarheid van het ternair model

Het ideale model is een model dat voor elke analist hetzelfde resultaat geeft en ook onafhankelijk is van de dag waarop het staal geanalyseerd wordt. In deze paragraaf wordt het experiment besproken waarbij ternaire validatiestalen door drie verschillende analisten worden geanalyseerd. De verschillende basislijncorrecties en pre-processingsmethoden worden daarbij in acht genomen. PCA pre-processing wordt hierbij niet in rekening gebracht.

Ook wordt de herhaalbaarheid onderzocht. Dit deelexperiment zal aantonen of het model dezelfde resultaten geeft op een ander tijdstip waarbij het experiment wordt uitgevoerd door dezelfde analist en onder dezelfde omstandigheden.

4.2.7.1 Intralaboratorium-reproduceerbaarheidsexperimenten

Er werden 18 ternaire validatiestalen geselecteerd en geanalyseerd door drie verschillende personen op dezelfde dag en op hetzelfde toestel. De resultaten van dit experiment worden weergegeven in Figuur 4.21 en Tabel 4.14. In Figuur 4.21 worden de gemiddelde MAE-waarden weergegeven per model en per persoon. In deze bespreking zal intralaboratorium-reproduceerbaarheid steeds afgekort worden als IL-reproduceerbaarheid.



Figuur 4.21: Resultaten van het intralaboratorium-reproduceerbaarheidsexperiment

Uit Figuur 4.21 volgt dat het meest IL-reproduceerbare model het 'AirPLS-SNV'-model is. Dit model geeft voor de drie verschillende analisten de meest vergelijkbare resultaten. De MAE-waarden van dit model zijn echter bij de grootste van alle modellen. Dus hoewel dit model IL-reproduceerbare resultaten geeft, is dit model niet het meest performant.

Het 'AirPLS-Geen'-model daarentegen, is het op een na meest IL-reproduceerbare model en geeft gunstige resultaten qua MAE-waarde. Ook in de vorige reeks experimenten bleek het 'AirPLS-Geen'-model het meest accuraat te zijn. Een AirPLS-basislijncorrectie geeft in het algemeen aanleiding tot modellen waarvan de IL-reproduceerbaarheid aanvaardbaar is.

Overigens valt op dat de modellen in het algemeen zonder basislijncorrectie hogere MAE-scores hebben qua IL-reproduceerbaarheid dan diegene met een basislijncorrectie. Hiermee wordt het nut van een basislijncorrectie aangetoond.

Ter vervollediging wordt Tabel 4.14 gegeven waaruit dezelfde conclusies kunnen worden getrokken zoals op basis van Figuur 4.21.

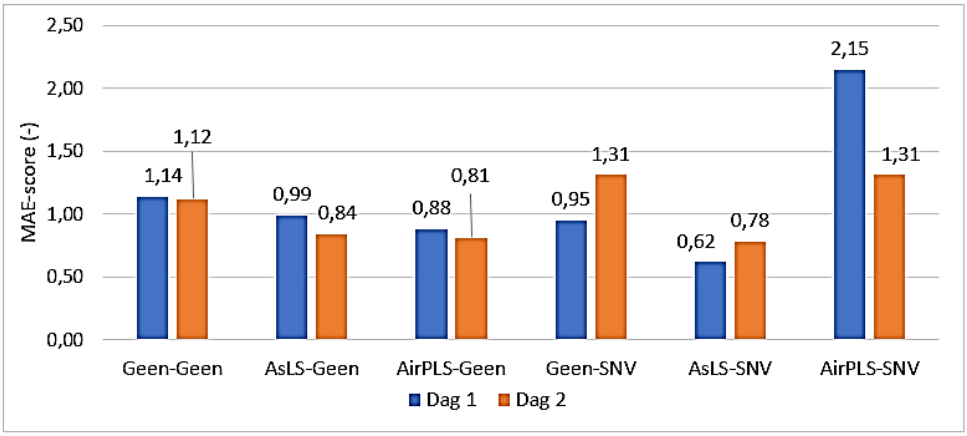
Tabel 4.14: Resultaten van het intralaboratorium-reproduceerbaarheidsexperiment

Model	Hyperparameters (epochs - hidden units)	Persoon 1	Persoon 2	Persoon 3
		Gemiddelde MAE ± standaardafwijking (-)	Gemiddelde MAE ± standaardafwijking (-)	Gemiddelde MAE ± standaardafwijking (-)
Geen-Geen	250-64	1,41 ± 0,83	1,50 ± 0,97	1,12 ± 0,71
AsLS-Geen	150-64	0,89 ± 0,41	1,17 ± 0,78	0,84 ± 0,31
AirPLS-Geen	250-64	0,89 ± 0,26	0,77 ± 0,21	0,81 ± 0,24
Geen-SNV	350-64	1,14 ± 0,27	1,02 ± 0,21	1,31 ± 0,33
AsLS-SNV	250-64	0,72 ± 0,20	0,58 ± 0,09	0,78 ± 0,38
AirPLS-SNV	250-64	1,34 ± 0,34	1,41 ± 0,31	1,31 ± 0,30

Tabel 4.14 toont ook aan dat de standaardafwijkingen steeds kleiner zijn dan 1. Dit duidt op een gelijkmatige inschatting van de verschillende stabilisatoren voor elke analist.

4.2.7.2 Herhaalbaarheidsexperimenten

Dezelfde validatiestalen die gebruikt werden in paragraaf 4.2.7.1 werden tweemaal getest op een andere dag door dezelfde persoon (22 dagen verschil tussen het testen). Figuur 4.22 en Tabel 4.15 geven de resultaten van dit experiment weer.



Figuur 4.22: Resultaten van het herhaalbaarheidsexperiment

Figuur 4.22 toont dat het 'Geen-Geen'-model aanleiding geeft tot de meest herhaalbare resultaten met slechts 0,02 MAE-eenheden verschil tussen de twee dagen. Daarop volgt het 'AirPLS-Geen'-model met 0,07 MAE-eenheden verschil. Het minst herhaalbare model is het 'AirPLS-SNV'-model met 0,84 MAE-eenheden verschil. Dit laatste model scoorde ook minder

goed qua IL-reproduceerbaarheid. Over de andere modellen kan gezegd worden dat de herhaalbaarheid aanvaardbaar is.

Daarnaast is hier het verschil tussen al dan niet de toepassing van een basislijncorrectie minder duidelijk dan bij het IL-reproduceerbaarheidsexperiment.

Wanneer naar het 'Geen-Geen'-model wordt gekeken, heeft dit model een hogere MAE-score dan andere modellen. Hoewel dit model dus gunstig is naar herhaalbaarheid toe, is het minder gunstig qua MAE-score. Daarom kan hier worden besloten dat het 'AirPLS-Geen'-model het meest aangewezen model is. Dit omdat het 'AirPLS-Geen'-model tot de meest herhaalbare modellen behoort en omdat de MAE-scores bij de laagste zijn van alle modellen voorgesteld in Figuur 4.22.

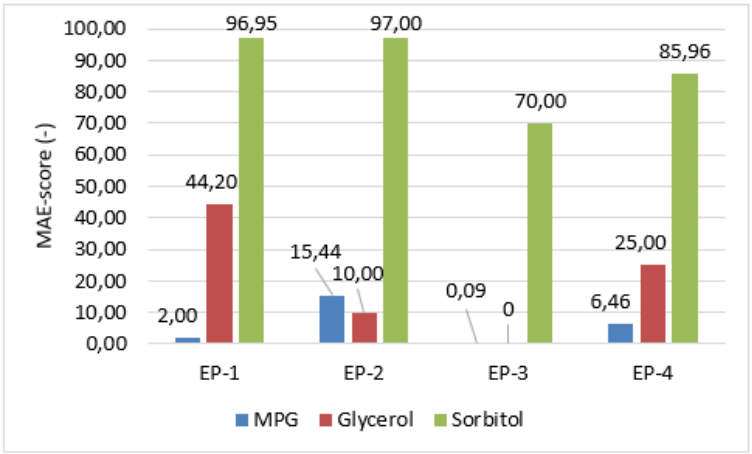
Tabel 4.15 wordt ter vervollediging ook meegegeven. Net zoals bij de IL-reproduceerbaarheid, wordt een standaardafwijking gevonden die kleiner is dan 1. Dit duidt ook hier op een gelijkmatige inschatting van de samenstellende componenten.

Tabel 4.15: Resultaten van het herhaalbaarheidsexperiment

Model	Hyperparameters (epochs - hidden units)	Dag 1	Dag 2
		Gemiddelde MAE ± standaardafwijking (-)	Gemiddelde MAE ± standaardafwijking (-)
Geen-Geen	250-64	1,14 ± 0,71	1,12 ± 0,71
AsLS-Geen	150-64	0,99 ± 0,32	0,84 ± 0,31
AirPLS-Geen	250-64	0,88 ± 0,26	0,81 ± 0,24
Geen-SNV	350-64	0,95 ± 0,08	1,31 ± 0,33
AsLS-SNV	250-64	0,62 ± 0,31	0,78 ± 0,38
AirPLS-SNV	250-64	2,15 ± 1,05	1,31 ± 0,30

4.2.8 Toepassing van het ternair model op gekende enzympreparaten

In dit experiment wordt het meest accurate ternair model, namelijk het 'AirPLS-Geen'-model, gebruikt om de samenstelling van gekende enzympreparaten (EP's) in te schatten. Er worden vier verschillende EP's getest. Omwille van confidentialiteitsredenen, zal niet expliciet gezegd worden om welke enzympreparaten het gaat. Er zal daarom steeds gesproken worden over EP 1 tot EP 4. Figuur 4.23 geeft dit resultaat weer.



Figuur 4.23: Resultaten van het toepassen van het 'AirPLS-Geen'-model op gekende enzympreparaten

Figuur 4.23 geeft weer dat MPG het nauwkeurigst van alle stabilisatoren kan worden ingeschat. Vooral sorbitol wordt foutief ingeschat met MAE-waarden die tot 97,00 kunnen

oplopen. Ook voor glycerol zijn de inschattingsfouten te groot om dit model in praktijk te gebruiken. Deze inschattingen van het model zijn niet in overeenstemming te brengen met de werkelijke samenstelling van de EP's.

Er kan dus worden besloten dat het 'AirPLS-Geen'-model, getraind met binaire en ternaire mengsels van stabilisatoren, niet in staat is om enzympreparaten nauwkeurig in te schatten.

Het oplossen van deze inschattingsfouten kan gebeuren door water aan de training- en teststalen toe te voegen, totdat de waterconcentratie ongeveer 50 % bedraagt. Betreffende EP's wordt namelijk vaak gezien dat er ongeveer 50 % water in de matrix zit. Het ontbreken van water in de trainingsfase, maakt dat het model niet in staat is de concentraties nauwkeurig in te schatten.

In paragraaf 4.3 wordt er water aan de matrix van de trainings- en teststalen toegevoegd en wordt er onderzocht of er dan wel een model kan worden opgebouwd dat de concentraties van de EP's nauwkeurig kan inschatten.

4.2.9 Tussentijdse conclusie op basis van de ternaire experimenten

Het analyseren van ternaire mengsels van stabilisatoren, lukt het nauwkeurigst wanneer er binaire en ternaire stalen in de training- en testdata zitten. Er werd aangetoond dat extra ternaire stalen toevoegen, bijdraagt aan de nauwkeurigheid van de inschattingen. Het 'AirPLS-Geen'-model werd het nauwkeurigst bevonden (MAE-validatiescore van 1,77 ± 0,43).

Dit 'AirPLS-Geen'-model is het meest IL-reproduceerbare en het meest herhaalbare model waarbij ook de MAE-waarde in acht werd genomen. Elk getest model werd IL-reproduceerbaar en herhaalbaar bevonden.

Het ternaire 'AirPLS-Geen'-model werd eveneens gebruikt om gekende EP's te analyseren. Dit toonde aan dat het ontbreken van water in de matrix zorgt voor onnauwkeurige inschattingen.

Daarnaast kon worden aangetoond dat PCA niet kan gebruikt worden als pre-processingsmethode onder de gehanteerde vorm van deep learning in deze thesis. De oorzaak ligt in de manier waarop de PC's worden gecreëerd.

4.3 Kwantificatie van een ternair mengsel van stabilisatoren in aanwezigheid van water

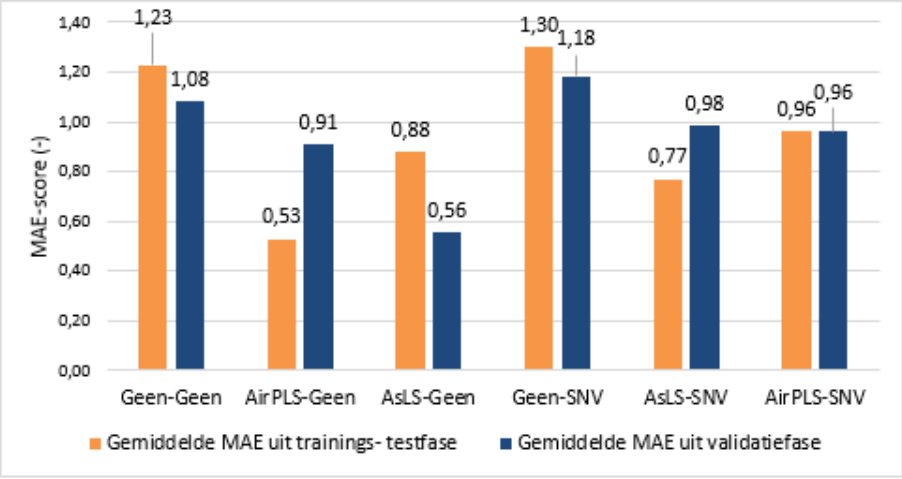
4.3.1 Evaluatie van het ternair model in aanwezigheid van water

Uit de vorige experimenten bleek dat het 'AirPLS-Geen'-model aanleiding gaf tot de nauwkeurigste validatieresultaten. Dit model werd ook het meest IL-reproduceerbaar en meest herhaalbaarheid bevonden.

In deze paragraaf wordt aan de binaire en ternaire stalen uit Figuur 4.15, RO-water toegevoegd tot de concentratie ervan ongeveer 50 % bedraagt. Dit wordt gedaan omdat deze stalen op die manier meer gelijken op de eigenlijke enzympreparaten. In deze context wordt er nog steeds gesproken van een binair of ternair mengsel van stabilisatoren. Nog een reden waarom deze stalen worden aangemaakt, is omdat geen van de voorgaande modellen er in slaagden om de enzympreparaten nauwkeurig in te schatten (paragraaf 4.2.8). Er worden nieuwe modellen aangemaakt die wel getraind zullen zijn om de waterconcentratie in te schatten.

In het deel omtrent Materialen en Methode (paragraaf 3) werd aangehaald dat de standaardoplossing van sorbitol 30 % water bevatte. Dit werd nagegaan via een Karl Fischer titratie met het toestel 915 KF Ti-Touch (Metrohm). In deze experimentenreeks wordt hiermee rekening gehouden en wordt de hoeveelheid zuivere sorbitol en water van elkaar gescheiden. Ook de andere standaarden werden getest op de aanwezigheid van water met een Karl Fischer titratie (915 KF Ti-Touch (Metrohm)) en bleken een waterconcentratie te hebben die kleiner was dan 0,50 %. Hier wordt bijgevolg geen rekening mee gehouden.

De training, het testen en de validatie werden voltrokken voor de verschillende basislijncorrecties in combinatie met geen of SNV pre-processing. De trainingsfase bevat 186 stalen, de testfase wordt voltrokken met 47 stalen en de validatiefase wordt gedaan met behulp van 38 stalen. De resultaten hiervan worden weergegeven in Figuur 4.24 en Tabel 4.16.



Figuur 4.24: Resultaten van de trainings- en testfase en validatiefase van het ternaire mengsel in aanwezigheid van water

Tabel 4.16: Resultaten van de trainings-, test- en validatiefase van het ternaire mengsel in aanwezigheid van water

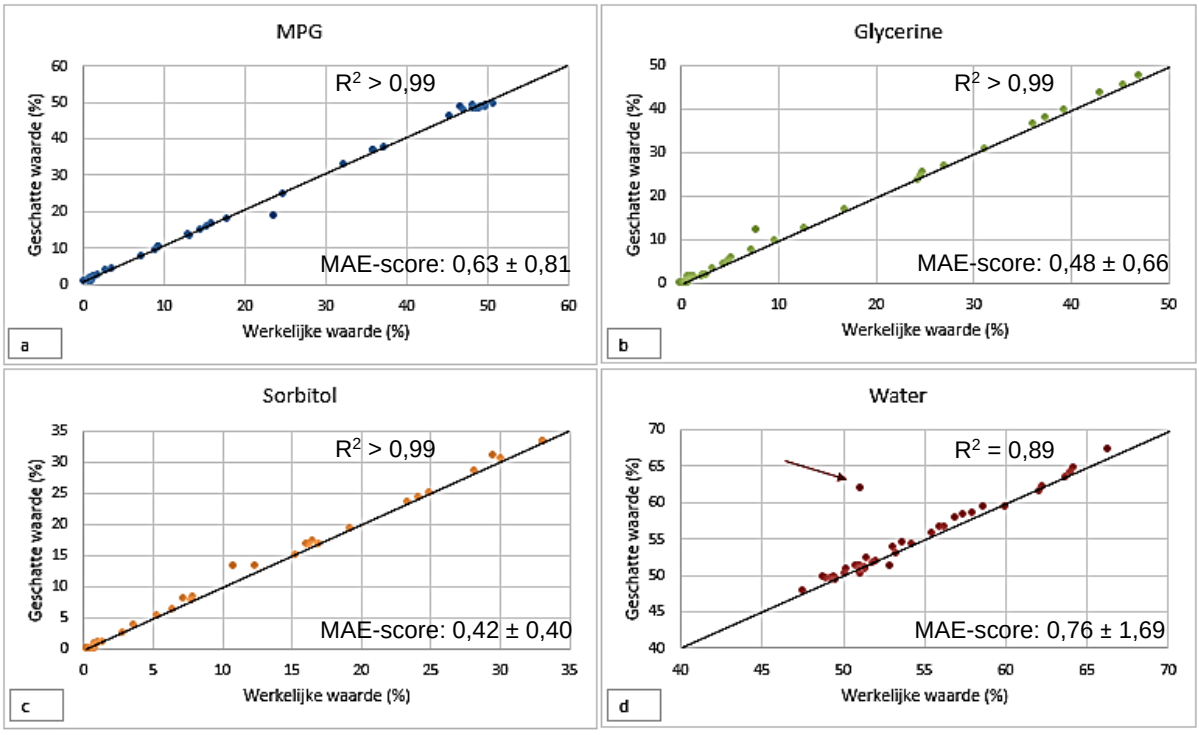
Model	Hyperparameters (epochs - hidden units)	Trainings- testfase	Validatiefase
		Gemiddelde MAE ± standaardafwijking (-)	Gemiddelde MAE ± standaardafwijking (-)
Geen-Geen	350-64	1,23 ± 0,51	1,08 ± 0,20
AirPLS-Geen	150-64	0,53 ± 0,11	0,91 ± 0,42
AsLS-Geen	250-64	0,88 ± 0,41	0,56 ± 0,15
Geen-SNV	350-64	1,30 ± 0,45	1,18 ± 0,38
AsLS-SNV	250-64	0,77 ± 0,15	0,98 ± 0,24
AirPLS-SNV	350-64	0,96 ± 0,28	0,96 ± 0,24

Figuur 4.24 en Tabel 4.16 tonen dat het laagste validatieresultaat verkregen wordt bij het 'AsLS-Geen'-model met een waarde van 0,56 ± 0,15. De kleinste MAE-waarde voor de trainings- en testfase wordt gevonden bij het 'AirPLS-Geen'-model met een waarde van 0,53 ± 0,11. Het grootste training- en testresultaat (1,30 ± 0,45) alsook de grootste validatiescore (1,18 ± 0,38), wordt gevonden bij het 'Geen-SNV'-model.

Hier wordt eveneens duidelijk dat een basislijncorrectie aanleiding geeft tot nauwkeurigere resultaten. Bijvoorbeeld het 'Geen-Geen'-model heeft een gemiddelde MAE-score van 1,08 ± 0,20 wat groter is dan de MAE-waarden van het 'AirPLS-Geen'-model en het 'AsLS-Geen'-model. Hetzelfde geldt voor de SNV pre-processing.

Op basis van Figuur 4.24 en Tabel 4.16 is geen pre-processing gunstiger dan een SNV pre-processing. Dit aangezien elk model lagere MAE-scores heeft wanneer geen pre-processing voltrokken wordt dan het overeenkomstig SNV-model.

De individuele bepalingen van de componenten worden voor het 'AsLS-Geen'-model getoond in Figuur 4.25.



Figuur 4.25: Grafische weergave van de ingeschatte waarde ten opzichte van de werkelijke waarde voor a) MPG, b) Glycerine, c) Sorbitol en d) Water voor het 'AsLS-Geen'-model

Er wordt opgemerkt dat Figuur 4.25 in elke deelfiguur een andere schaling kent van de assen om het lineaire karakter beter te visualiseren. De R^2-waarden zijn telkens te vinden op de desbetreffende deelfiguren. Daarnaast stelt de zwarte diagonaal het ideale model voor.

In Figuur 4.25.a wordt de inschatting van MPG getoond. Deze inschattingen benaderen het ideale model (voorgesteld door de diagonaal) in grote mate. De MAE-waarde voor deze inschatting bedraagt 0,63 ± 0,81.

De inschatting voor glycerine en sorbitol, respectievelijk Figuur 4.25.b en Figuur 4.25.c, hebben een MAE-score van respectievelijk 0,48 ± 0,66 en 0,42 ± 0,40. Ook deze benaderen het ideale model en hebben MAE-waarden die telkens lager is dan deze van MPG.

De MAE-score van water is 0,76 ± 1,69. Deze inschattingsfout is slechts iets groter dan die bij de andere componenten. Ook voor water wordt het ideale model benaderd.

Er is slechts één staal dat minder accuraat wordt ingeschat met een MAE van 10,84. Dit wordt gedemonstreerd in Figuur 4.25.d met een pijl. Bij dit staal slaagt het model er ook minder goed in om sorbitol, glycerol en MPG in te schatten. De inschattingsfout op deze componenten is wel betrekkelijk lager dan de fout op de inschatting van het water. Waarschijnlijk is er een fout gebeurd bij de aanmaak van dit staal aangezien dit staal als enige uit de reeks niet nauwkeurig wordt ingeschat, terwijl andere vergelijkbare concentraties wel accuraat worden ingeschat.

In dit experiment worden de laagste MAE-scores gehaald in vergelijking met alle voorgaande experimenten. Er kan dus gezegd worden dat de gehanteerde modellen, de meest performante zijn die tot nog toe werden opgebouwd.

4.3.2 Bepaling van de kwantificatielimiet

Uit vorig experiment werd geconcludeerd dat het ‘AsLS-Geen’-model het nauwkeurigste was inzake de inschatting van vier componenten. Aangezien dit model zal worden gebruikt om onbekenden te analyseren, wordt de kwantificatielimiet (LOQ) berekend. Dit wordt gedaan voor de vier componenten afzonderlijk. Hiervoor worden tussen vijf en tien blanco-metingen gedaan per componenten. Dit wil zeggen dat in desbetreffende mengsels de overige drie componenten wél aanwezig zijn.

Na de blanco-meting wordt vervolgens de LOQ berekend via Vergelijking 4.1. De standaarddeviatie op de blanco-metingen wordt vermenigvuldigd met een factor. Deze factor heeft een waarde tussen 0 en 10. Er wordt aangeraden deze factor gelijk te stellen aan 10 (het maximum). Op die manier wordt de LOQ waarde niet onderschat. (105)

$$LOQ = s_{blanco} * 10 \quad \textbf{(4.1)}$$

Met LOQ de kwantificatielimiet (%) en

s_{blanco} de standaarddeviatie van de blanco-stalen (%)

Dit wordt als voorbeeld gedaan voor sorbitol met behulp van negen blanco-metingen. De standaarddeviatie van deze blanco-stalen bedraagt 0,177 %. Dit geeft met behulp van Vergelijking 4.1 volgende waarde voor de LOQ:

$$LOQ_{Sorbitol} = 0{,}177 * 10 = 1{,}77\ \%$$

De LOQ voor sorbitol bedraagt 1,77 %. Voor de overige LOQ-waarden wordt verwezen naar Tabel 4.17. Deze LOQ-waarden werden eveneens berekend via Vergelijking 4.1 en bestonden telkens uit negen blanco-metingen.

Tabel 4.17: LOQ-bepaling van sorbitol, glycerine, MPG en water

Component	Standaarddeviatie (%)	LOQ (%)
Sorbitol	0,177	1,77
Glycerine	0,223	2,23
MPG	0,250	2,50
Water	2,98(4)	29,84

De LOQ-waarden van sorbitol (1,77 %), glycerine (2,23 %) en MPG (2,50 %) liggen in dezelfde lijn. De ondergrens voor de kwantificatie door het model is dus voor deze drie componenten ongeveer gelijk.

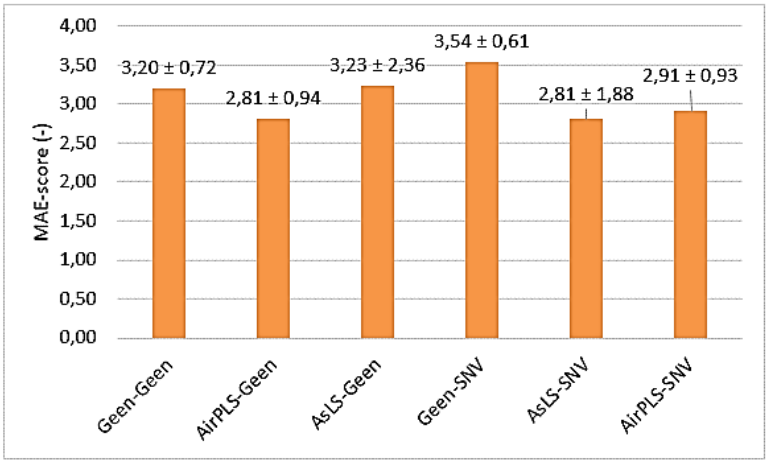
Voor water is de LOQ (29,84 %) duidelijk hoger dan de andere waarden in Tabel 4.17. Dit is resultaat is te nuanceren. Dit hogere resultaat kan verklaard worden door de training die het model heeft gekregen. In de trainingsfase bedroeg de minimale waterconcentratie 47,89 %. Het is dus niet verwonderlijk dat de LOQ dan hoger ligt dan bijvoorbeeld die van sorbitol waar de minimaal aanwezige concentratie 0,23 % was.

Ook wordt opgemerkt dat het model er in acht van de negen stalen in geslaagd is om een concentratie van 0 % aan te geven inzake MPG-bepaling. Slechts in één geval werd er 0,75 % MPG ingeschat terwijl er eigenlijk geen MPG in het mengsel zat.

4.3.3 Toepassing van het model op enzympreparaten

4.3.3.1 Enzympreparaten met gekende samenstelling

In deze paragraaf wordt getest hoe de modellen scoren op commercieel verkrijgbaar enzympreparaten (EP's). Er worden vier verschillende EP's getest en omwille van confidentialiteitsredenen, zal ook hier niet expliciet gezegd worden om welke enzympreparaten het gaat. Er zal daarom steeds gesproken worden over EP 1 tot EP 4. Dit zijn dezelfde EP's zoals in paragraaf 4.2.8. De verschillende EP's werden geanalyseerd en verwerkt door de modellen. De resultaten hiervan worden globaal, als een gemiddelde, weergegeven in Figuur 4.26.



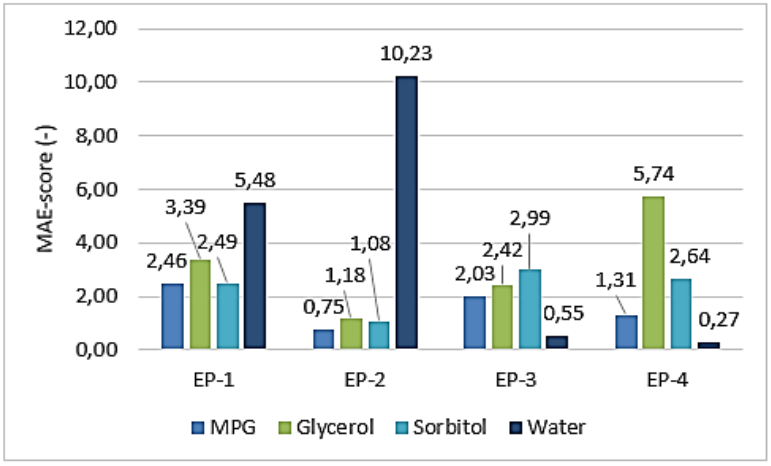
Figuur 4.26: Resultaten van de validatie van enzympreparaten

Figuur 4.26 toont dat 'AirPLS-Geen'-model en het 'AsLS-SNV'-model beide een gemiddelde MAE-score hebben van 2,81. Dit is de laagste MAE-score van alle modellen. De andere

modellen scoren vergelijkbaar. De spreiding van het 'AirPLS-Geen'-model is echter kleiner dan die van het 'AsLS-SNV'-model waardoor kan gezegd worden dat het 'AirPLS-Geen'-model de voorkeur krijgt. Het 'Geen-SNV'-model heeft de grootste validatiescore (3,54 ± 0,61).

Een reden waarom de inschatting van het 'AsLS-SNV'-model nu accuraat is en in de vorige experimenten minder, ligt aan de trainingsfase. Zoals reeds een aantal keren vermeld, worden in de trainingsfase de gewichten ingeschat van de verschillende noden. Deze inschatting blijkt voor deze toepassing, namelijk enzympreparaten, beter te zijn dan voor louter ternaire stalen met water.

Omdat de inschatting per EP anders zijn, wordt voor het 'AirPLS-Geen'-model de inschattingen per componenten getoond in Figuur 4.27.



Figuur 4.27: Inschattingen van de individuele componenten door het 'AirPLS-Geen'-model

Figuur 4.27 geeft weer dat de inschattingen van de verschillende EP's niet gelijkaardig zijn. EP 1 heeft ongeveer eenzelfde inschattingsfout voor zowel MPG als sorbitol. Vooral water wordt in tegenstelling tot de andere componenten minder nauwkeurig ingeschat.

EP 2 heeft vooral een minder goede inschatting voor water, terwijl de andere componenten vergelijkbaar worden ingeschat met een MAE-waarde van ongeveer 1. Water wordt niet steeds als minst nauwkeurig ingeschat: bij EP 3 en EP 4 wordt deze component het meest accuraat bepaald.

De foutieve inschattingen zijn voornamelijk te wijten aan de andere componenten die zich nog in de matrix bevinden waarop het model niet getraind is. De totale inschatting van het model bedraagt 100 % terwijl er minder dan 100 % vertegenwoordigd wordt door MPG, glycerol, sorbitol en water. Aangezien het model de overige componenten niet kan inschatten, wordt een foutieve resultaat verkregen. De stabilisatoren alsook water vertegenwoordigen een groot aandeel van de matrix, waardoor de fouten nog relatief klein zijn.

Om het model te optimaliseren in een volgend onderzoek, zouden er extra componenten kunnen worden toegevoegd aan de training- en testset. Daardoor zal het relatieve aandeel van de stabilisatoren en water meer gelijken op de werkelijke samenstelling van de EP's.

4.3.3.2 Enzympreparaten met een onbekende samenstelling

Er werden enkele andere enzympreparaten getest waarvan de samenstelling onbekend is. Het model geeft dan ingeschatte waarden waarvan in de regel niet kan worden nagegaan of ze correct zijn. Zodoende werden er tien bijkomende EP's (EP 5 tot en met EP 14) getest met behulp van het 'AsLS-Geen'- en het 'AirPLS-Geen'-model uit paragraaf 4.3.3.1.

Het ‘AirPLS-Geen’-model gaf resultaten waarvan de totale ingeschatte concentratie rond 130 % lag. Dit kan dus beschouwd worden als foutief. Het ‘AsLS-Geen’-model gaf een totale concentratie die om en bij de 100 % bedroeg en daarom wordt hiermee verder gewerkt. De resultaten hiervan worden weergegeven in Tabel 4.18. Ook werd het watergehalte van enkele EP’s bepaald via een Karl Fischer meting (915 KF Ti-Touch (Metrohm)).

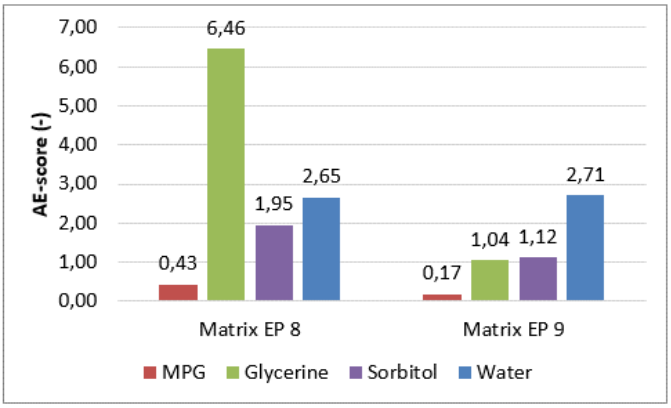
Tabel 4.18: Resultaten van de analyse op onbekende EP's door middel van het ‘AsLS-Geen’-model

EP	MPG (%)	Glycerine (%)	Sorbitol (%)	Water (%)	Water via Karl Fischer (%)	AE op waterbepaling (-)
5	50,44	1,41	2,35	49,01	-	-
6	50,87	1,97	2,80	48,85	43,64	5,21
7	0	29,72	24,21	54,25	50,02	4,23
8	12,17	19,07	6,00	70,03	59,06	10,97
9	6,99	47,54	5,99	45,40	31,55	13,85
10	42,31	4,71	6,21	48,58	-	-
11	42,70	6,20	4,90	48,15	-	-
12	22,06	5,43	20,44	53,97	46,07	7,90
13	0	28,57	0	79,44	59,53	19,91
14	18,52	18,69	13,71	50,66	-	-

Zoals Tabel 4.18 weergeeft, werd van sommige EP’s het watergehalte via een Karl Fischer meting nagegaan om op die manier toch een evaluatie te koppelen aan de inschattingen die het ‘AsLS-Geen’-model gemaakt heeft. EP 6 en EP 7 hebben een inschattingsfout van ongeveer 5 AE-eenheden met betrekking tot de waterconcentratie. EP 8, EP 9 en EP 12 hebben een inschattingsverschil met Karl Fischer van ongeveer 10 AE-eenheden.

Wat vooral opvalt, zijn de inschattingen van 70,03 % (EP 8) en 79,44 % (EP 13). Dit is opmerkelijk aangezien het model nooit getraind is om dergelijke hoge inschattingen te maken. Het model is getraind met stalen van maximaal 65 % water. Een hogere inschatting maken, lijkt dus in eerste instantie onmogelijk. De Karl Fischer bepaling geeft bij diezelfde stalen een inschatting van het water van 59,06 % en 59,53 % bij respectievelijk EP 8 en EP 13.

Er werden twee stalen aangemaakt die qua concentratie gelijken op de inschatting van de matrix van EP 8 en EP 9 door het model. Deze werden vervolgens gevalideerd. Er werd gekozen voor EP 8 door de hoge inschatting van de waterconcentratie en voor EP 9 omdat hier ook een hoge AE-waarde voor water werd vastgesteld. Dit terwijl de concentratie wel zou moeten accuraat ingeschat worden aangezien het model hier wel op getraind is. Indien het model in staat is om de nagebootste matrices correct te in te schatten, dan zijn de validaties van de EP’s mogelijk correct. De validatie van deze stalen wordt weergegeven in Figuur 4.28 en voltrokken door het ‘AsLS-Geen’-model.



Figuur 4.28: Validatie van stalen die de matrix van EP 8 en EP 9 nabootsen met het ‘AsLS-Geen’-model

Figuur 4.28 toont dat de inschattingen van MPG nauwkeurig gebeuren met een AE die kleiner is dan 0,43. Ook voor hoeveelheid sorbitol en de hoeveelheid glycerine van de nagebootste matrix van EP 9, wordt een AE-score gevonden die kleiner is dan 1,12.

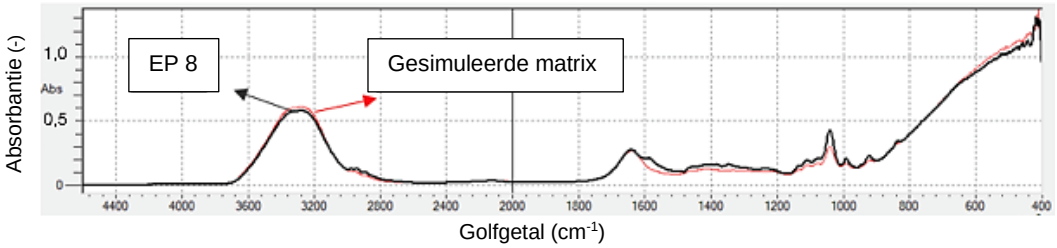
De concentratie aan glycerine van de nabootsing van de matrix van EP 8, kent een relatief hoge AE-score van 6,46.

Voornamelijk de inschattingen van de hoeveelheid water vallen op. Het water wordt, zowel voor de nabootsing van EP 8 en EP 9, bijna even nauwkeurig ingeschat met een AE-score van respectievelijk 2,65 en 2,71. Hieruit blijkt dat het 'AsLS-Geen'-model effectief een inschatting van meer dan 70 % water kan maken en dit overigens even nauwkeurig kan doen als een inschatting van ongeveer 45 % water. Dit is opmerkelijk aangezien het model hier nooit getraind is om 70 % water te herkennen in een IR-spectrum. Het feit dat het model ook accuraat 45 % water kan inschatten, maakt dat de inschattingsfout bij EP 9 veroorzaakt wordt door een bijkomende component in de matrix.

Het model zal op basis van het IR-spectrum van EP 8 en EP 13 wel degelijk hebben vastgesteld dat er respectievelijk 70,03 % en 79,44 % aan 'water' inzat. Deze concentraties aan water zijn te hoog van uit het perspectief van een stabiel EP. Hieruit volgt dus dat er minstens nog een component aanwezig is, die gelijkaardige pieken vertoond in het IR-spectrum zoals water. Door deze overlap in pieken ziet het model deze component(en) aan als water en is de inschatting bijgevolg te hoog. De oplossing voor dit probleem kan zijn om proberen te achterhalen om welke component het gaat en om het model hier vervolgens op te trainen.

Daarnaast is een mogelijke hypothese dat het model herkent dat er geen 70,03 % (EP 8) of 79,44 % (EP 13) in het EP zit maar door het willen genereren van een totale concentratie van 100 %, de concentraties aan MPG, glycerine, sorbitol en water heeft verhoogd.

Deze hypothese kan ontkracht worden een test via de IR-spectra. EP 8 en EP 13 hebben een vergelijkbaar spectrum, althans voor het menselijk oog, in vergelijking met de gesimuleerde matrices. Als voorbeeld hiervan wordt Figuur 4.29 gegeven die de vergelijking weergeeft voor EP 8 met de overeenkomstig gesimuleerde matrix.



Figuur 4.29: Vergelijking van het IR-spectrum van EP 8 (zwart) en de gesimuleerde matrix van EP 8

Figuur 4.29 toont dat de twee meest prominente pieken van het IR-spectrum van water, namelijk deze rond 3350 cm^{-1} en de piek rond 1650 cm^{-1}, grote gelijkenissen vertonen tussen het werkelijke EP en de gesimuleerde matrix. Er wordt vermoed dat, wanneer het gebied tussen 1000 cm^{-1} en 1200 cm^{-1} bekeken wordt, er minstens nog een component aanwezig is die een hydroxylgroep heeft (primair, secundair en/of tertiair) en daarnaast nog pieken heeft vergelijkbaar met die van water. Dit wordt vastgesteld door de hogere ligging van het IR-spectrum van EP 8 in de regio tussen 1000 cm^{-1} en 1200 cm^{-1}.

Welke component hier precies aanwezig is, wordt in deze thesis niet verder onderzocht.

4.4 Aanbevelingen voor verder onderzoek

Uit paragraaf 4.3.3 bleek dat de commercieel verkrijgbare EP's nog meer componenten dan enkel de stabilisatoren glycerol, sorbitol, MPG en water bevatten. Aangezien het model een totale concentratie van 100 % wil genereren, treden er fouten op naar inschatting toe van de individuele componenten. Deze fouten kunnen vermeden worden wanneer het model getraind wordt om ook deze andere componenten te herkennen.

In een hierop volgend onderzoek zouden het aantal componenten kunnen worden uitgebreid. Het model zou hierop vervolgens kunnen getraind worden om een nauwkeurigere inschatting te krijgen van de concentraties. Er kan dan verder gebouwd worden op het 'AsLS-Geen'-model en/of het 'AirPLS-Geen'-model aangezien deze het meest performant bleken met vooral de klemtoon op het 'AsLS-Geen'-model.

Daarnaast zou in een hierop verder bouwend onderzoek, de hoeveelheid water in de stalen kunnen worden verlaagd. In plaats van een concentratie van om en bij de 50 % water, zou deze concentratie verlaagd kunnen worden om de LOQ-waarde van het water te verlagen. Het model wordt op die manier nog robuuster.

5 BESLUIT

Door de jaren heen is de hoeveelheid aan surfactants in de wasmiddelen verminderd door de komst en toepassing van enzympreparaten (EP's). Deze zijn een ecologisch hoogwaardig alternatief en zorgen voor een doelgerichte verwijdering van het vuil. Om enzymen in een stabiele toestand te houden in de wasmiddelen, zijn stabilisatoren nodig. In deze thesis wordt de klemtoon gelegd op glycerol, sorbitol en monopropyleenglycol (MPG) als stabilisatoren. De doelstelling is om ternaire mengsels van deze stabilisatoren te kunnen kwantificeren met behulp van verzwakte totale reflectie – Fourrier transformatie infraroodspectroscopie (ATR-FTIR) en deep learning.

Uit de literatuur blijkt de kwantificatie met ATR-FTIR echter niet voor de hand liggend te zijn. Een toepassing van de wet van Lambert-Beer is hier niet mogelijk daar er vooronderstellingen zijn die voor ATR-FTIR niet opgaan. Toch lijkt deze techniek veelbelovend daar deze niet-destructief, snel en een groen alternatief is voor de meer conventionele methoden.

Om de kwantificatie in deze thesis te voltrekken, wordt gebruik gemaakt van deep learning. Daartoe wordt een fully connected neuraal netwerk opgebouwd dat de infraroodspectra zal ontleden. Deze spectra worden gegeven als input aan het model en als output wordt een concentratie verwacht die zo nauwkeurig mogelijk is. In de opbouw van dit neuraal netwerk worden trainings- en testfase alsook de validatiefase van elkaar onderscheiden. De validatiefase gebeurt op basis van onbekende stalen en beoordeelt op die manier de performantie van het model. Daarvoor wordt gebruik gemaakt van de gemiddelde absolute fout (MAE) of de absolute fout (AE).

In de praktijk blijkt echter dat deze spectra onderhevig zijn aan basislijnverschuivingen. Daarom wordt een basislijncorrectie-algoritme geschreven onder de vorm van adaptive iteratively reweighted penalized least squares (AirPLS) en asymmetric least squares (AsLS). Om te kijken of dit een gunstig effect heeft, wordt ook steeds de referentie gemaakt naar een manuele basislijncorrectie en het origineel spectrum. Door de grote hoeveelheid aan data, lijkt het ook aangewezen om aan pre-processing te doen. Dit domein wordt in acht genomen door de technieken van principale component analyse (PCA) en standard normal variate (SNV) te onderzoeken. Ook wordt het nut van pre-processing onderzocht door te refereren naar geen pre-processing.

In het experimenteel deel worden binaire en ternaire mengsels van stabilisatoren gebruikt. De experimenten kunnen worden onderverdeeld in drie grote delen waarin de doeltreffendheid van de opgebouwde modellen wordt nagegaan. Eerst worden binaire mengsels van stabilisatoren gebruikt om een relatief eenvoudig model op te bouwen. Deze kennis wordt vervolgens meegenomen naar een tweede deel waarbij een complexer ternair model wordt opgebouwd. Na de evaluaties van deze modellen, worden de mengsels van stabilisatoren aangelengd met water om EP's te kunnen analyseren.

Uit de experimenten met de binaire mengsels van stabilisatoren kon niet worden besloten welke combinatie van een basislijncorrectie en een pre-processing de ideaal was. Algemeen kon gezegd worden dat een manuele basislijncorrectie minder geschikt was alsook dat PCA pre-processing minder aanleiding gaf tot nauwkeurige resultaten. De manuele basislijncorrectie werd daarom niet meer in acht genomen. Er werd in deze experimentenreeks

ook gekeken naar wat het effect was van het werken met hoog gecorreleerde golfgetallen. Het bleek dat door het weglaten van spectrale informatie de performantie van de modellen gevoelig afnam.

In een volgende benadering werden ternaire modellen opgebouwd. Daar het ternair model reeds complexer was, konden binaire validatiemengsels in eenzelfde mate ingeschat worden zoals met de binaire modellen. Er werd daarnaast vastgesteld dat de aanwezigheid van ternaire stalen in de datasets noodzakelijk was om ternaire mengsels te kunnen analyseren. Door een verdere verhoging van het aantal ternaire stalen werden de modellen nauwkeuriger.

Het 'AirPLS-Geen'-model (AirPLS-basislijncorrectie en geen pre-processing) bleek het meest performant met een MAE-waarde van 1,77 ± 0,43. PCA pre-processing gaf ook hier aanleiding tot relatief hoge resultaten. De oorzaak hiervan ligt aan de coherentie van de principale componenten van de trainings- en testdata en de validatiedata. Daarom wordt PCA als pre-processingstechniek uitgesloten.

In deze set experimenten werd ook de intralaboratorium-reproduceerbaarheid en herhaalbaarheid nagegaan. Het bleek dat ook hier het 'AirPLS-Geen'-model het meest performante model was waarbij zowel de MAE als de onderlinge verhouding van persoon-persoon en dag-dag in rekening genomen werden. Uit deze reeks experimenten bleek dat geen pre-processing en een basislijncorrectie het meest gunstig waren.

De stap naar een ternair model van stabilisatoren in de aanwezigheid van water werd vervolgens gezet. Het 'AsLS-Geen'-model haalde het laagste validatieresultaat met een gemiddelde inschattingsfout van 0,56 ± 0,15. Dit resultaat is veelbelovend naar de analyse toe van commercieel verkrijgbare EP's. Dit model kent een kwantificatielimiet van 2,23 % voor glycerol, 1,17 % voor sorbitol, 2,50 % voor MPG en 29,84 % voor water.

Om deze modellen te testen, werden gekende EP's onderworpen aan een analyse. Daaruit bleek dat het 'AirPLS-Geen'-model het meest accuraat de concentraties kon bepalen met een MAE van 2,81 ± 0,84. Ook onbekende EP's werden onderzocht waaruit het 'AsLS-Geen'-model het meest performant bleek. Voor de validatie werd het watergehalte bepaald via een Karl Fischer titratie en vergeleken met de waarde die het model aangaf. Twee EP's werden gekenmerkt door een betrekkelijk grotere MAE-waarde waarbij de inschatting van water meer dan 70 % bedroeg. Dit is opmerkelijk want het model is hier niet op getraind. Daarom werd zelf een matrix nagebootst met ongeveer 70 % water. Het 'AsLS-Geen'-model slaagde erin om dit watergehalte in te schatten met een AE-waarde van 2,65. Hieruit blijkt dat er nog minstens een component in het enzympreparaat zit met vermoedelijk vergelijkbare pieken zoals die van water.

Er kan dus besloten worden dat het mogelijk is om ternaire stabilisatorenmengsels nauwkeurig te kwantificeren met behulp van FTIR en deep learning. Daarenboven is deze methode IL-reproduceerbaar en herhaalbaar. Er werd aangetoond dat een basislijncorrectie en geen pre-processing de ideale combinatie is. Uit de analyse van onbekende EP's blijkt het 'AsLS-Geen'-model veelbelovend. In een verder onderzoek zouden het aantal componenten in de trainingsdata kunnen uitgebreid worden. Ook zou de kwantificatielimiet van water kunnen verlaagd worden door de concentratie van water te verlagen in de trainingset.

Referentielijst

1. Christeyns NV. Christeyns. [Internet]. 2021. [Geraadpleegd op 2021 Mar 2]. Via: https://www.christeyns.com/nl/professionele-textielverzorging.
2. Olsen HS, Falholt P. The role of enzymes in modern detergency. J Surfactants Deterg. 1998; 1(4): 555–67.
3. Infinita Biotech. What Are The Benefits Of Enzymes In Detergent? [Internet]. 2020. [Geraadpleegd op 2021 Mar 28]. Via: https://infinitabiotech.com/blog/benefits-of-enzymes-in-detergents/.
4. Kahn Academy. Enzymes and the active site. [Internet]. 2021. [Geraadpleegd op 2021 Feb 10]. Via: https://www.khanacademy.org/science/ap-biology/cellular-energetics/enzyme-structure-and-catalysis/a/enzymes-and-the-active-site.
5. Kravchenko GB. Enzyme Classifications . Kinetics Mechanisms of action Specificity and Regulation. [PowerPoint presentation]. 2012.
6. Campbell PN. Enzymes in Industry: Chapter 3: General Production Methods. Vol. 6, Biochemical Education. 2004. 37–82 p.
7. Niyonzima FN. Detergent-compatible bacterial cellulases. J Basic Microbiol. 2019; 59(2): 134–47.
8. Kumari U, Singh R, Ray T, Rana S, Saha P, Malhotra K, et al. Validation of leaf enzymes in the detergent and textile industries: launching of a new platform technology. Plant Biotechnol J. 2019; 17(6): 1167–82.
9. Shivanand P, Jayaraman G. Isolation and characterization of a metal ion-dependent alkaline protease from a halotolerant Bacillus aquimaris VITP4. Indian J Biochem Biophys. 2011; 48(2): 95–100.
10. Beck K. Different Types of Enzymes. [Internet]. Sciencing. 2018. [Geraadpleegd op 2021 Feb 10]. Via: https://sciencing.com/feedback-inhibition-important-regulating-enzyme-activity-9661.html.
11. Singh R, Kumar M, Mittal A, Mehta PK. Microbial enzymes: industrial progress in 21st century. 3 Biotech. 2016; 6(2): 1–15.
12. Vittaladevaram V. Fermentative Production of Microbial Enzymes and their Applications: Present status and future prospects. J Appl Biol Biotechnol. 2017; 5(04): 90–4.
13. Singh S, Singh J. Effect of polyols on the conformational stability and biological activity of a model protein lysozyme. AAPS PharmSciTech. 2003; 4(3): 1–9.
14. Castellanos IJ, Crespo R, Griebenow K. Poly(ethylene glycol) as stabilizer and emulsifying agent: A novel stabilization approach preventing aggregation and inactivation of proteins upon encapsulation in bioerodible polyester microspheres. J Control Release. 2003; 88(1): 135–45.
15. Mokhtar NF, Raja Noor Zaliha RNZR, Muhd Noor ND, Mohd Shariff F, Ali MSM. The immobilization of lipases on porous support by adsorption and hydrophobic interaction method. Catalysts. 2020; 10(7): 1–17.
16. Sipos B, Szilágyi M, Sebestyén Z, Perazzini R, Dienes D, Jakab E, et al. Mechanism of the positive effect of poly(ethylene glycol) addition in enzymatic hydrolysis of steam pretreated lignocelluloses. Comptes Rendus - Biol. 2011; 334(11): 812–23.
17. Iyer P V., Ananthanarayan L. Enzyme stability and stabilization-Aqueous and non-aqueous environment. Process Biochem. 2008; 43(10): 1019–32.

18. Plaxton WC. Avoiding Proteolysis during the Extraction and Purification of Active Plant Enzymes. Plant Cell Physiol. 2019; 60(4): 715–24.

19. Milosavić NB, Prodanović RM, Velićković D, Dimitrijević A. Macroporous poly(GMA-co-EGDMA) for enzyme stabilization. Vol. 1504, Methods in Molecular Biology. 2017. 139–147 p.

20. Braham SA, Siar EH, Arana-Peña S, Bavandi H, Carballares D, Morellon-Sterling R, et al. Positive effect of glycerol on the stability of immobilized enzymes: Is it a universal fact? Process Biochem. 2021; 102: 108–21.

21. Mienda BS, Yahya A, Galadima IA, Shamsir MS. An overview of microbial proteases for industrial applications. Res J Pharm Biol Chem Sci. 2014; 5(1): 388–96.

22. Bioninja. 7.6 Enzymes. [Internet]. 2016. [Geraadpleegd op 2021 Mar 27]. Via: http://www.old-ib.bioninja.com.au/higher-level/topic-7-nucleic-acids-and/enzymes.html.

23. Sigma-Aldrich. PMSF: Phenylmethylsulfonyl fluoride. [Internet]. 2021. [Geraadpleegd op 2021 Mar 30]. Via: https://www.sigmaaldrich.com/catalog/product/roche/pmsfro?lang=fr&region=BE.

24. Kumar SS, Sabu A. Quality Control and Downstream Processing of Therapeutic Enzymes. Vol. 1148, Advances in Experimental Medicine and Biology. 2019. 345–381 p.

25. Poku RA, Amissah F, Duverna R, Aguilar BJ, Kiros G-E, Lamango NS. Polyisoprenylated methylated protein methyl esterase as a putative drug target for androgen-insensitive prostate cancer. Ecancermedicalscience. 2014; 8: 459–459.

26. Jin J, Tarrant RD, Bolam EJ, Angell-Manning P, Soegaard M, Pattinson DJ, et al. Production, quality control, stability, and potency of cGMP-produced Plasmodium falciparum RH5.1 protein vaccine expressed in Drosophila S2 cells. npj Vaccines. 2018; 3(1).

27. Harris TK, Keshwani MM. Chapter 7: Measurement of Enzyme Activity. [Internet]. 1st ed. Vol. 463, Methods in Enzymology. Elsevier Inc.; 2009. 57–71 p. Via: http://dx.doi.org/10.1016/S0076-6879(09)63007-X.

28. Wang Y, Wang G, Moitessier N, Mittermaier AK. Enzyme Kinetics by Isothermal Titration Calorimetry: Allostery, Inhibition, and Dynamics. Front Mol Biosci. 2020; 7(October): 1–19.

29. Hadwan MH. Simple spectrophotometric assay for measuring catalase activity in biological tissues. BMC Biochem. 2018; 19(1): 1–8.

30. Hinton-Sheley P. What is ATR-FTIR? [Internet]. AZoLifeSciences. 2021. [Geraadpleegd op 2021 Feb 11]. Via: https://www.azolifesciences.com/article/What-is-ATR-FTIR.aspx.

31. Ausili A, Sánchez M, Gómez-Fernández JC. Attenuated total reflectance infrared spectroscopy: A powerful method for the simultaneous study of structure and spatial orientation of lipids and membrane proteins. Biomed Spectrosc Imaging. 2015; 4(2): 159–70.

32. Paar A. Attenuated total reflectance (ATR). [Internet]. 2021. [Geraadpleegd op 2021 Mar 27]. Via: https://wiki.anton-paar.com/en/attenuated-total-reflectance-atr/#:~:text=In areas where the sample,total reflectance" (ATR).

33. Bradley M. Pathlength Considerations With ATR Sampling in FTIR. [Internet]. 2018. [Geraadpleegd op 2021 Mar 31]. Via: https://www.labcompare.com/10-Featured-Articles/352695-Pathlength-Considerations-With-ATR-Sampling-in-FTIR/.

34. Chemistry LibreTexts. Infrared Spectroscopy. [Internet]. 2020. [Geraadpleegd op 2021 Feb 11]. Via: https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Spectroscopy/Vibrational_Spectroscopy/Infrared_Spectroscopy/Infrared_Spectroscpy#:~:text=Absorption of IR r.

35. Schelden V. FT-IR studie van afbraakprocessen van vluchtige organische stoffen via atmosferische plasma's. UGent. UGent; 2008.

36. Williams M. What is the Speed of Light? [Internet]. 2016. [Geraadpleegd op 2021 Feb 15]. Via: https://www.universetoday.com/38040/speed-of-light-2/.

37. Stein J. Planck's Constant: The Number That Rules Technology, Reality, and Life. [Internet]. The nature of reality. 2011. [Geraadpleegd op 2021 Feb 15]. Via: https://www.pbs.org/wgbh/nova/article/plancks-constant/.

38. Van de Voorde I. Spectroscopische technieken: infrarood. 2019th–2020th ed. Gent; 2019.

39. Van de Ven MFC, Qiu J, Zhang Y. Infrarood metingen op bitumen: Voorbeeld van mogelijkheden . 2012.

40. Winter A. How to Find Functional Groups in the IR Spectrum. [Internet]. 2021. [Geraadpleegd op 2021 Feb 12]. Via: https://www.dummies.com/education/science/chemistry/how-to-find-functional-groups-in-the-ir-spectrum/.

41. Ed Vitz, John W. Moore, Justin Shorb, Xavier Prat-Resina, Tim Wendorff & AH. 21.5: The Spectra of Molecules- Infrared. [Internet]. 2020. [Geraadpleegd op 2021 Feb 11]. Via: https://chem.libretexts.org/Bookshelves/General_Chemistry/Book%3A_ChemPRIME_(Moore_et_al.)/21%3A_Spectra_and_Structure_of_Atoms_and_Molecules/21.05%3A_The_Spectra_of_Molecules-_Infrared.

42. MacKie DM, Jahnke JP, Benyamin MS, Sumner JJ. Simple, fast, and accurate methodology for quantitative analysis using Fourier transform infrared spectroscopy, with bio-hybrid fuel cell examples. MethodsX. [Internet]. 2016 ;3: 128–38. Via: http://dx.doi.org/10.1016/j.mex.2016.02.002.

43. Klein O, Roth A, Dornuf F, Schöller O, Mäntele W. The good vibrations of beer. The use of infrared and UV/Vis spectroscopy and chemometry for the quantitative analysis of beverages. Zeitschrift fur Naturforsch - Sect B J Chem Sci. 2012; 67(10): 1005–15.

44. ThermoFischer. FTIR FAQs: Using the Beer-Lambert law in FT-IR ATR for quantitative analysis of a time-sensitive, migrating substance (e.g., erucamide) in a polymer is difficult. How can this be overcome? [Internet]. 2020. [Geraadpleegd op 2021 Mar 31]. Via: https://www.thermofisher.com/be/en/home/industrial/spectroscopy-elemental-isotope-an alysis/spectroscopy-elemental-isotope-analysis-learning-center/molecular-spectroscopy-information/ftir-information/ftir-faqs.html.

45. Jiménez-Carvelo AM, Osorio MT, Koidis A, González-Casado A, Cuadros-Rodríguez L. Chemometric classification and quantification of olive oil in blends with any edible vegetable oils using FTIR-ATR and Raman spectroscopy. LWT - Food Sci Technol. 2017; 86: 174–84.

46. Mallah MA, Sherazi STH, Bhanger MI, Mahesar SA, Bajeer MA. A rapid Fourier-transform infrared (FTIR) spectroscopic method for direct quantification of paracetamol content in solid pharmaceutical formulations. Spectrochim Acta - Part A Mol Biomol Spectrosc [Internet]. 2015; 141 :64–70. Via: http://dx.doi.org/10.1016/j.saa.2015.01.036.

47. Dreissig I, Machill S, Salzer R, Krafft C. Quantification of brain lipids by FTIR spectroscopy and partial least squares regression. Spectrochim Acta - Part A Mol Biomol Spectrosc. 2009; 71(5): 2069–75.

48. Luo J, Ying K, Bai J. Savitzky-Golay smoothing and differentiation filter for even number data. Signal Processing. 2005; 85(7): 1429–34.

49. Alrezj OA, Patchava K, Benaissa M, Alshebeili SA. Pre-processing to Enhance the Quantitative Analysis of Glucose from NIR and MIR Spectra. [Internet]. Vol. 65, IFMBE Proceedings. 2018. Via: http://link.springer.com/10.1007/978-981-10-5122-7.

50. Crochet RB. Data processing data processing flowchart. 2017; 1–31.

51. Sigma-Aldrich. D-Sorbitol. [Internet]. 2021. [Geraadpleegd op 2021 Feb 12]. Via: https://www.sigmaaldrich.com/catalog/ product/sigma/S1876?lang=fr&region=BE.

52. TGI Chemicals. Glycerol. [Internet]. 2021. [Geraadpleegd op 2021 Feb 12]. Via: https://www.tcichemicals.com/BE/en/p/G0316v.

53. Danish M, Mumtaz MW, Fakhar M, Rashid U. Response surface methodology based optimized purification of the residual glycerol from biodiesel production process. Chiang Mai J Sci. 2017; 44(4): 1570–82.

54. Sigma-Aldrich. 1,2-Propanediol. [Internet]. 2021. [Geraadpleegd op 2021 Feb 12]. Via: https://www.sigmaaldrich.com/catalog/product/sigma/p1009?lang=fr&region=BE.

55. He S, Zhang W, Lijuan L, Yu H, Jiming H, Wanyi X, et al. Baseline correction for Raman spectra using an improved asymmetric least squares method. Anal Methods. 2014;(6): 4402–7.

56. Eilers PHC. A perfect smoother. Anal Chem. 2003; 75(14): 3631–6.

57. He S, Fang S, Liu X, Zhang W, Xie W, Zhang H, et al. Investigation of a genetic algorithm based cubic spline smoothing for baseline correction of Raman spectra. Chemom Intell Lab Syst. [Internet]. 2016; 152: 1–9. Via: http://dx.doi.org/10.1016/j.chemolab. 2016.01.005.

58. Shao L, Griffiths PR. Automatic baseline correction by wavelet transform for quantitative open-path fourier transform infrared spectroscopy. Environ Sci Technol. 2007; 41(20): 7054–9.

59. Zhang F, Tang X, Tong A, Wang B, Wang J. An automatic baseline correction method based on the penalized least squares method. Sensors (Switzerland). 2020; 20(7).

60. Zhang ZM, Chen S, Liang YZ. Baseline correction using adaptive iteratively reweighted penalized least squares. Analyst. 2010; 135(5): 1138–46.

61. Héberger K. Chemoinformatics — multivariate mathematical – statistical methods. Med Appl mass Spectrom. 2008; 141–69.

62. Grisanti E, Totska M, Huber S, Calderon CK, Hohmann M, Lingenfelser D, et al. Dynamic localized SNV, Peak SNV, and partial peak SNV: Novel standardization methods for preprocessing of spectroscopic data used in predictive modeling. J Spectrosc. 2018; 2018.

63. Syvilay D, Wilkie-Chancellier N, Trichereau B, Texier A, Martinez L, Serfaty S, et al. Evaluation of the standard normal variate method for Laser-Induced Breakdown Spectroscopy data treatment applied to the discrimination of painting layers. Spectrochim Acta - Part B At Spectrosc. [Internet]. 2015; 114: 38–45. Via: http://dx.doi.org /10.1016/j.sab.2015.09.022.

64. Bock T. What is a Latent Variable. [Internet]. 2020. DisplayR. [Geraadpleegd op 2021 Feb 22]. Via: https://www.displayr.com/what-is-a-latent-variable/.

65. Saikat M, Jun Y, Maitra S, Yan J. Principle Component Analysis and Partial Least Squares : Two Dimension Reduction Techniques for Reegression. Casualty Actuar Soc. 2008; 79–90.

66. Javatpoint. Purpose of Normalization. [Internet]. Javatpoint. 2018. [Geraadpleegd op 2021 May 31]. Via: https://www.javatpoint.com/dbms-purpose-of-normalization.

67. Jaadie Z. WHEN AND WHY TO STANDARDIZE YOUR DATA? A simple guide on when it is necessary to standardize your data. Builtin. 2021.

68. Géron A. Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems. O'Reilly Media. 2017.

69. Chollet F. Deep Learning with Python. MITP-Verlags GmbH & Co. KG. Manning Publications Co.; 2018. 373 p.

70. Missing Link AI. Deep Learning for Big Data. [Internet]. 2020. [Geraadpleegd op 2021 Feb 20]. Via: https://missinglink.ai/guides/neural-network-concepts/deep-learning-for-big-data/.

71. Frankenfield J. Artificial intelligence AI. [Internet]. 2021. [Geraadpleegd op 2021 Feb 15]. Via: https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp.

72. Sperling E. Deep Learning Spreads. [Internet]. 2018. [Geraadpleegd op 2021 Feb 15]. Via: https://semiengineering.com/deep-learning-spreads/.

73. AISOLAB. INTRO TO AI #1: ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, NEURAL NETWORKS AND DEEP LEARNING: WHAT ARE THE DIFFERENCES? [Internet]. 2017. [Geraadpleegd op 2021 Feb 20]. Via: https://aiso-lab.com/intro-to-ai-1-artificial-intelligence-machine-learning-neural-networks-and-deep-learning-what-are-the-differences/.

74. Technology. Differences between machine learning and software engineering. [Internet]. 2018. [Geraadpleegd op 2021 Feb 15]. Via: https://futurice.com/blog/differences-between-machine-learning-and-software-engineering.

75. Grossfield B. Deep learning vs machine learning: a simple way to understand the difference. [Internet]. 2020. [Geraadpleegd op 2021 Feb 15]. Via: https://www.zendesk.com/blog/machine-learning-and-deep-learning/#:~:text=To recap the differences between,intelligent decisions on its own.

76. IBM Cloud Education. Neural networks. [Internet]. 2020. [Geraadpleegd op 2021 Feb 15]. Via: https://www.ibm.com/cloud/learn/neural-networks.

77. Rashid M, Singh H, Goyal V. The use of machine learning and deep learning algorithms in functional magnetic resonance imaging—A systematic review. Expert Syst. 2020; 37(6): 1–29.

78. Harikrishna B. Deep Learning. [Internet]. 2018. [Geraadpleegd op 2021 Feb 16]. Via: https://medium.com/datadriveninvestor/deep-learning-2025e8c4a50.

79. Amini A. 6S191_MIT_DeepLearning_L1.pdf. [PowerPoint presentation]. MIT; 2020.

80. Github Pages. Deep Learning From Scratch. [Internet]. Zied HY's Data Science Blog. 2021. [Geraadpleegd op 2021 Feb 16]. Via: https://ziedhy.github.io/Introduction_Deep_Learning.html.

81. Homenick C. What's The Role Of Weights And Bias In a Neural Network? [Internet]. 2020. [Geraadpleegd op 2021 Feb 18]. Via: https://morioh.com/p/eb23d31bb742.

82. Feng J, He X, Teng Q. Commonly-used-activation-functions-a-Sigmoid-b-Tanh-c-ReLU-and-d-LReLU. [Internet]. 2019. [Geraadpleegd op 2021 Mar 30]. Via: https://www.researchgate.net/figure/Commonly-used-activation-functions-a-Sigmoid-b-Tanh-c-ReLU-and-d-LReLU_fig3_335845675.

83. Doshi S. Various Optimization Algorithms For Training Neural Network. [Internet]. Towards data science. 2019. [Geraadpleegd op 2021 Mar 2]. Via: https: //towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6.

84. Vieira S, Lopez Pinaya WH, Garcia-Dias R, Mechelli A. Deep neural networks. [Internet]. Machine Learning: Methods and Applications to Brain Disorders. Elsevier Inc.; 2019. 157–172 p. Via: http://dx.doi.org/10.1016/B978-0-12-815739-8.00009-2.

85. Subir V, Sanjiv D. Chapter 7 Training Neural Networks Part 1. [Internet]. 2018. [Geraadpleegd op 2021 Feb 18]. Via: https://srdas.github.io/DLBook/.

86. Subir V, Sanjiv D. Chapter 8 Training Neural Networks Part 2. [Internet]. 2020. [Geraadpleegd op 2021 Feb 18]. Via: https://srdas.github.io/DLBook/ ImprovingModelGeneralization.html#detecting-underfitting.

87. Edspresso Team. Overfitting and underfitting. [Internet]. Edspresso. 2021. [Geraadpleegd op 2021 Feb 18]. Via: https://www.educative.io/edpresso/overfitting-and-underfitting.

88. Varma S, Sanjiv D. Chapter 9 Training Neural Networks Part 3. [Internet]. 2018. [Geraadpleegd op 2021 Feb 18]. Via: https://srdas.github.io/DLBook /HyperParameterSelection.html.

89. Amini A. 6S191_MIT_DeepLearning_L3.pdf. [PowerPoint presentation]. MIT; 2020.

90. Bharath R, Reza BZ. Chapter 4. Fully Connected Deep Networks. [Internet]. 2021. [Geraadpleegd op 2021 Feb 18]. Via: https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html.

91. Amini A. 6S191_MIT_DeepLearning_L2.pdf. [PowerPoint presentation]. MIT; 2020.

92. NumPy. NumPy. [Internet]. 2020. [Geraadpleegd op 2021 Mar 23]. Via: https://numpy.org/.

93. Matiplot. Matplotlib: Visualization with Python. [Internet]. 2021. [Geraadpleegd op 2021 Mar 22]. Via: https://matplotlib.org/.

94. TensorFlow. TensorFlow Core. [Internet]. 2021. [Geraadpleegd op 2021 Mar 23]. Via: https://www.tensorflow.org/guide/tensor.

95. Pydata. Pandas: Package overvieuw. [Internet]. 2021. [Geraadpleegd op 2021 Mar 23]. Via: https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html.

96. Peng J, Peng S, Jiang A, Wei J, Li C, Tan J. Asymmetric least squares for multiple spectra baseline correction. Anal Chim Acta. [Internet]. 2010; 683(1): 63–8. Via: http://dx.doi. org/10.1016/j.aca.2010.08.033.

97. Github Gist: Perimosocordiae. Asymmetric Least Squares. [Internet]. 2015. [Geraadpleegd op 2021 Feb 20]. Via: https://gist.github.com/perimosocordiae/ efabc30c4b2c9afd8a83.

98. Liu X, Zhang Z, Sousa PFM, Chen C, Ouyang M, Wei Y, et al. Selective iteratively reweighted quantile regression for baseline correction. Anal Bioanal Chem. 2014; 406(7): 1985–98.

99. Github: zmzhang. AirPLS. [Internet]. github. 2016. [Geraadpleegd op 2021 Feb 20]. Via: https://github.com/zmzhang/airPLS/blob/master/airPLS.py.

100. Pelliccia D. Two scatter correction techniques for NIR spectroscopy in Python. [Internet]. 2018. [Geraadpleegd op 2021 Feb 20]. Via: https://nirpyresearch.com/two-scatter-correction-techniques-nir-spectroscopy-python/.

101. Galarnyk M. PCA using Python (scikit-learn). [Internet]. 2017. [Geraadpleegd op 2021 Mar 1]. Via: https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60.

102. Sealens M. Interpretatie en modellering van multi instrumentele analytische data met Deep Learning. KU Leuven; 2020.

103. Halvorsen KB. Relationship between SVD and PCA. How to use SVD to perform PCA? [Internet]. StackExchange. 2015. [Geraadpleegd op 2021 Apr 24]. Via: https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca.

104. Beleites C. PCA and the train/test split. [Internet]. StackExchange. 2021. [Geraadpleegd op 2021 Apr 23]. Via: https://stats.stackexchange.com/questions/ 55718/pca-and-the-train-test-split.

105. Uhrovčík J. Strategy for determination of LOD and LOQ values - Some basic aspects. Talanta. 2014; 119: 178–80.

Bijlagen

Bijlage A: Python-code voor de AsLS-basislijncorrectie

Bijlage B: Python-code voor de AirPLS-basislijncorrectie

Bijlage C: Python-code voor SNV pre-processing

Bijlage D: Python-code voor PCA pre-processing

Bijlage E: Python-code voor de opbouw van een neuraal netwerk

Bijlage A PYTHON-CODE VOOR DE ASLS-BASISLIJNCORRECTIE

```
def AsLS_baseline(intensities, asymmetry_param=0.0001,
                  smoothness_param=1000, max_iters=1000, conv_thresh=0.1,
                  verbose=False):

  smoother = WhittakerSmoother(intensities, smoothness_param,
                               deriv_order=2)
  p = asymmetry_param
  w = np.ones(intensities.shape[0])
  for i in range(max_iters):
    z = smoother.smooth(w)
    mask = intensities > z
    new_w = p*mask + (1-p)*(~mask)
    conv = scipy.linalg.norm(new_w - w)
    if verbose:
      print (i+1, conv)
    if conv < conv_thresh:
      break
    w = new_w
  else:
    print ('ALS did not converge in %d iterations' % max_iters)
  return z
class WhittakerSmoother(object):
  def __init__(self, signal, smoothness_param, deriv_order=1):
    self.y = signal
    assert deriv_order > 0
    d = np.zeros(deriv_order*2 + 1, dtype=int)
    d[deriv_order] = 1
    d = np.diff(d, n=deriv_order)
    n = self.y.shape[0]
    k = len(d)
    s = float(smoothness_param)
        diag_sums = np.vstack([
        np.pad(s*np.cumsum(d[-i:]*d[:i]), ((k-i,0),), 'constant')
        for i in range(1, k+1)])
    upper_bands = np.tile(diag_sums[:,-1:], n)
    upper_bands[:,:k] = diag_sums
    for i,ds in enumerate(diag_sums):
      upper_bands[i,-i-1:] = ds[::-1][:i+1]
    self.upper_bands = upper_bands

  def smooth(self, w):
    foo = self.upper_bands.copy()
    foo[-1] += w  # last row is the diagonal
    return solveh_banded(foo, w * self.y, overwrite_ab=True,
                         overwrite_b=True)
def loopAsLS(df):
     Resultaat = pd.DataFrame()
     for column in df:
          kolommen = pd.DataFrame(df[column])
          kolommen.columns = ['Absorbantie']
          y = kolommen['Absorbantie']
          y = np.array(y).astype(float)
          baseline = AsLs_baseline(y)
          resetBaseline = y - baseline
          resetBaseline = pd.DataFrame(resetBaseline)
          Resultaat = pd.concat([Resultaat,resetBaseline], axis = 1)
     return Resultaat
```

Bijlage B PYTHON-CODE VOOR DE AIRPLS-BASISLIJNCORRECTIE

```
def WhittakerSmooth(x,w,lambda_,differences=1):
    X=np.matrix(x)
    m=X.size
    i=np.arange(0,m)
    E=eye(m,format='csc')
    D=E[1:]-E[:-1]
    W=diags(w,0,shape=(m,m))
    A=csc_matrix(W+(lambda_*D.T*D))
    B=csc_matrix(W*X.T)
    background=spsolve(A,B)
    return np.array(background)

def airPLS(x, lambda_=100, porder=50, itermax=15):

    m=x.shape[0]
    w=np.ones(m)
    for i in range(1,itermax+1):
        z=WhittakerSmooth(x,w,lambda_, porder)
        d=x-z
        dssn=np.abs(d[d<0].sum())
        if(dssn<0.001*(abs(x)).sum() or i==itermax):
            if(i==itermax): print ('WARING max iteration reached!')
            break
        w[d>=0]=0
        w[d<0]=np.exp(i*np.abs(d[d<0])/dssn)
        w[0]=np.exp(i*(d[d<0]).max()/dssn)
        w[-1]=w[0]
    return z
def loopAirPLS(df):
    Resultaat = pd.DataFrame()
    for column in df:
        kolommen = pd.DataFrame(df[column])
        kolommen.columns = ['Absorbantie']
        y = kolommen['Absorbantie']
        y = np.array(y).astype(float)
        baseline = airPLS(y)
        resetBaseline = y - baseline
        resetBaseline = pd.DataFrame(resetBaseline)
        Resultaat = pd.concat([Resultaat,resetBaseline], axis = 1)
    return Resultaat
```

Bijlage C PYTHON-CODE VOOR SNV PRE-PROCESSING

```
def snv(input_data):
 input_data = np.asarray(input_data)
 data_snv = np.zeros_like(input_data)
 for i in range(len(data_snv)):
  data_snv [i] =
  (input_data[i]- np.mean(input_data)) / np.std(input_data)
 return data_snv
```

Bijlage D PYTHON-CODE VOOR PCA PRE-PROCESSING

```
def PCA_Spectrale_Data (df):
    pca = PCA(n_components=2)
    principalComponents = pca.fit_transform(df)
    principalDf = pd.DataFrame(data = principalComponents, columns =
    ['principal component 1', 'principal component 2'])
    print (principalDf)
    print (pca.explained_variance_ratio_)
    return principalDf
```

Bijlage E PYTHON CODE VOOR DE OPBOUW VAN EEN NEURAAL NETWERK

Stap in schema (paragraaf 3.2.5)	Python-code Bron (69)

1 en 2	`#Importeren van de packages`
	`from numpy import loadtxt`
	`from keras.models import load_model`
	`import pandas as pd`
	`import numpy as np`
	`import sys`
	`from numpy.random import seed`
	`from keras import models`
	`from keras import layers`
	`from keras.utils import to_categorical`
	`from keras import optimizers`
	`import matplotlib.pyplot as plt`
	`import xlsxwriter`
	
	`#Data laden en in dataframe plaatsen`
	`xls_file = pd.ExcelFile('Bestandsnaam')`
	`df = xls_file.parse('Data')`
	`df = pd.DataFrame(df)`
	Code voor het laden van de pakketten en het inlezen van het Excel-bestand met de spectrale data

3	`#Manipuleren van de data`
	`X=df.drop(["% MPG","% Glycerol","% Sorbitol"],axis=1)`
	`S=df["% Sorbitol"]`
	`M=df["% MPG"]`
	`G=df["% Glycerol"]`
	`S_train, S_test, M_train, M_test, G_train, G_test =`
	`S[:319], S[319:], M[:319], M[319:], G[:319], G[319:]`
	
	`#Data omzetten naar een rij`
	`s_train = np.array(S_train).astype("float32")`
	`s_test= np.array(S_test).astype("float32")`
	`m_train = np.array(M_train).astype("float32")`
	`m_test= np.array(M_test).astype("float32")`
	`g_train = np.array(G_train).astype("float32")`
	`g_test= np.array(G_test).astype("float32")`
	Code voor het indelen in training- en testset voor een ternair mengsel

4.I.i	`model = models.Sequential()`
	`model.add(layers.Dense(32, activation='relu',input_shape=(2,)))`
	`model.add(layers.Dense(32, activation = 'relu'))`
	`model.add(layers.Dense(1))`
	`model.compile(optimizer = 'rmsprop', loss='mse',`
	`metrics=['mean_absolute_error'])`
	Opbouw van een neuraal netwerk met drie lagen voor een binair mengsel

4.I. ii	`inputs=Input((3,))`
	`x=Dense(32,activation='relu')(inputs)`
	`x=Dense(32, activation = 'relu')(x)`
	`MPG_prediction = layers.Dense(1, name='MPG')(x)`
	`Glycerine_prediction = layers.Dense(1, name='Glycerine')(x)`
	`Sorbitol_prediction = layers.Dense(1, name='Sorbitol')(x)`
	`model = Model(inputs=inputs,outputs=`
	`[MPG_prediction, Glycerine_prediction,Sorbitol_prediction])`
	`model.compile(optimizer = 'rmsprop', loss=['mse','mse','mse'],`
	`metrics=['mean_absolute_error'])`
	Opbouw van een neuraal netwerk voor een ternair mengsel

4.II	`history = model.fit(partial_x_train, partial_y_train,`
	`epochs=50, batch_size=1,validation_data=(x_val,y_val),verbose=1)`
	Code voor het meegeven van hyperparameters aan het model

5	`model.save("model_Deep_Learning")`
	`Model = load_model('model_Deep_Learning')`
	Opslaan en laden van een model op de harde schijf

6	`xls_file = pd.ExcelFile('Data_ter_validatie.xlsx')`
	`df = pd.DataFrame(xls_file.parse('Data'))`
	`X_Val = df.drop(["% MPG",'% Glycerine'],axis=1)`
	`predictions = DL_model.predict(X_Val)`
	Code voor de validatie van een model

FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN
TECHNOLOGIECAMPUS GENT
Gebroeders De Smetstraat 1
9000 GENT, België
tel. + 32 92 65 86 10
iiw.gent@kuleuven.be
www.iiw.kuleuven.be