

Afstudeerscriptie:
Onderzoek naar diagnose methode voor
spoorstroomlopen

Door:
Datum:

Jamie Teunissen
29/05/2020

Bedrijfsbegeleider:
1^{ste} examiner
2^{de} examiner

Ir. Ronald Helder
Ir. Mehmet Can
Ir. Johan Woudstra

ProRail

DE HAAGSE
HOGESCHOOL

Voorwoord

In dit document is een methode beschreven naar het stellen van diagnose bij de spoorstroomloop (een detectiesysteem om te detecteren of trein op een blok rijdt). Deze scriptie is geschreven in het kader van mijn afstuderen aan de opleiding Elektrotechniek aan De Haagse Hogeschool in Delft en in opdracht van ProRail.

Samen met mijn begeleider ingenieur Ronald Helder en zijn collega's van de afdeling treinbeveiligssystemen heb we deze opdracht geformuleerd. Het was een uitdagend en complex vraagstuk waar ik in het begin nog niet zeker van was of ik hem kon beantwoorden. Maar naar uitgebreid en gestructureerd onderzoek en met mijn domein kennis, kan ik dan toch met trots vertellen dat het gelukt is. Dat is ook zeker te danken aan mijn begeleider en zijn collega's Fred Baauw, Monique Voorderhake en Erwin van Wonderen.

Naast de collega's bij ProRail wil ik bij deze ook graag de docenten van mijn opleiding bedanken voor de kennis die ze op mij hebben overgedragen de afgelopen jaren. Daarnaast wil ook nog graag ingenieur Mehmet Can en ingenieur Johan Woudstra willen bedanken voor de ondersteuning die ze mij hebben geboden gedurende dit traject, en de goede feedback op het verslag die ik van hun heb mogen ontvangen. Als laatste wil ik graag mijn vrienden bedanken voor de wijze raad en hulp bij het redigeren van dit document.

Ondanks de schokkende invloeden die het Covid-19 virus op onze maatschappij heeft uitgeoefend, en de manier waarop het invloed heeft gehad mijn opdracht en de samenwerking. Zal ik toch terugkijken op deze afstudeerperiode als een leuke tijd, waar ik weer nieuwe dingen heb geleerd over mijn vakgebied en over mezelf.

Ik wens u veel leesplezier,

Jamie Teunissen
Delft, 29 mei 2020

Samenvatting

Op het spoor in Nederland worden verscheidene componenten gebruikt om het systeem draaiende te houden. Het systeem komt geregeld in storing en hierdoor kan de normale dienstregeling geen doorgang meer vinden. Dit zorgt uiteindelijk voor vertragingen bij de reiziger. In deze scriptie zal worden ingegaan op een van die systemen die gebruikt worden op het spoor, genaamd de spoorstroomloop. Het spoor is opgedeeld in blokken en in zo een blok mag maar een trein rijden. De spoorstroomloop is het systeem dat detecteert doormiddel van een resonantiecircuit dat er een trein rijdt op zo een blok. Dit systeem stuurt een signaal naar de seinen zodat de andere gebruikers weten of een blok bezet is.

Het doel van dit onderzoek is het ontwikkelen van een methode waarmee het opsporen van fouten in het systeem sneller kan verlopen, en waarbij de uiteindelijke wens van ProRail is een hulpmiddel wat automatische fouten kan identificeren. De kans dat een monteur het defecte component in een keer vindt is 57.6 procent. De hoofdvraag in deze scriptie is: Is er methode waarmee een hogere kans te behalen valt bij het identificeren van defecten componenten aan de resonantiecircuit in de spoorstroomloop?

Er zijn in deze scriptie verschillende mogelijke methodes van meten opgesteld, deze zijn op de literatuur over fout diagnose bij elektrische circuits gebaseerd. De resultaten uit de metingen zijn gebruikt om vier verschillende machine learning modellen te trainen. Deze zijn getraind aan de hand geaccepteerde en afwijkende waarden van het resonantiekring. Aan de hand van de betrouwbaarheidsscores van die machine learning modellen is vergeleken welke combinatie van machine learning model en meet methode de beste resultaten opleveren.

Uit onderzoek is bepaald dat de combinatie waarbij buiten de relaiskast aan de aansturingszijde van de spoorstroomloop moet worden gemeten, in combinatie met een gradient boosted decision tree machine learning model.

Met deze methode is uit simulaties een nauwkeurigheid bereikt van 86 procent. Daarbij is op te merken dat deze methode beter is in het identificeren van problemen in de relaiskast aan de aansturingszijde, dan aan de detectiezijde.

Op basis van de behaalde resultaten kan ProRail overwegen om deze methode verder te ontwikkelen in een hulpmiddel voor monteurs. Omdat de resultaten op basis van simulaties zijn verkregen, zijn verdere testen op installaties in het Railcenter benodigd.

Abstract

Many systems keep the dutch railway system running. But sometimes one of those systems will not work properly. In the event of a malfunction in one of those systems, the normally used timetable will be interrupted. This will result in delays for the end-users of the railway system. In this report one of those systems used is researched, this system is called the track circuit. The rail systems is been divided into several sections. To operate the railway system safely only one train is allowed on a section. The track circuit is a detection system that will register if a train is present on a section. When that's the case a signal will be sent to the signals to indicate to other users of the railway that the section is occupied.

The main purpose of this report is the development of a method to identify certain errors in the track circuit to ultimately reduce maintenance times. From the results of this research, a device can be developed which can automatically detect faulty components. The main research question to be answered in this report is therefore to what extent it is possible to automate fault diagnostic of the track circuit while only considering fault modes that are related to a failure of electrical components.

In this report, several methods of measuring the track circuit are being considered. These methods of measuring the track circuit are based on the literature in the field of electrical component fault diagnostics. The results which are measured with the proposed measuring methods will be used to train four different machine learning models. The different methods and models can be compared with the accuracy results of the machine learning models. With these results, it's possible to determine which combination measuring method and machine learning model is best for the track circuit.

Eventually, a measuring method has been chosen, where the technician measures on one side of the track circuit on the rails outside. In combination with the use of the Gradient Boosted Decision Tree machine learning model.

With this method, an accuracy score of identifying the different proposed fault modes was achieved of 86%. With the remark that the method is better in recognising fault modes that are in the relay box on the control side, then recognising fault modes on the detecting relay box of the track circuit.

Therefore its recommended to do further testing on installations in the Railcentre, to get a clearer picture on how this method will work in the field. Since the method shows promising results in the simulations, it may be useful to set the next steps to developing a device which automates this process.

Inhoudsopgave

Voorwoord	iii
Samenvatting	v
Abstract	vii
1 Inleiding	1
1.1 Aanleiding	2
1.2 Probleemstelling	2
1.3 Doelstelling	2
1.3.1 Onderzoeksvragen	2
1.4 Methode	3
1.4.1 Afbakening	3
1.5 Leeswijzer	3
2 Theoretisch kader	5
2.1 Spoorstroomlopen	5
Spoor als een transmissielijn	6
Gehele systeem	7
2.2 Resonantiecircuits	8
2.3 Foutmodi	9
2.3.1 Theoretisch bepalen van de koppel	10
Spoorrelais	10
Spoorstroomloopmodel	11
2.4 Machine learning	14
2.4.1 Waarom machine learning?	15
2.4.2 Gebruik van een ML model	16
Overzicht	16
2.4.3 Classificatiemodellen	17
Kunstmatige neural network	17
Support Vector Machines	17
Gradient boosted decision tree	19
K-nearest neighbors	19
2.5 Relevante literatuur	20
2.5.1 Huidige diagnose strategie van ProRail	20
Kansberekening	20
2.5.2 Een op conditie gebaseerde diagnose	21
2.5.3 Een op foutmodi gebaseerd model	21
2.5.4 Gedrag gebaseerd model	21
2.5.5 Teststimulus, diagnostische metingen, extractie en selectie	21
2.5.6 Keuze methode	22
2.6 Wavelets	23
2.6.1 Wavelet als filterbank	24

3	Methode	25
3.1	Basisprincipe	25
3.2	Meetopstelling	25
3.2.1	Testsignaal	25
3.2.2	Definitie foutmodi	26
3.3	Voorgestelde methodes	28
3.3.1	Zonder toegevoegd signaal	28
	Methode 1: circuit zonder toevoeging meten	28
3.3.2	Met toegevoegde signaal	28
	Methode 2: vervanging ingangssignaal	28
	Methode 3a: enkelvoudige meting zonder ingangssignaal	28
	Methode 3b: Getrapte meting aan beide zijde zonder ingangs- signaal	29
3.4	Signaalanalyse	29
3.4.1	Overzicht	30
3.5	Analyse van de resultaten	31
3.5.1	Bepalen beste methode en ML-model	31
	Nauwkeurigheid bepalen	31
3.5.2	Componentanalyse van de resultaten	32
4	Resultaten	33
4.1	Uitwerking Methodes	33
4.1.1	Keuze diepgaande analyse	34
4.2	Diepgaande analyse gekozen methode	35
4.2.1	Nauwkeurigheid	36
4.2.2	Componentanalyse fouten	36
5	Conclusie	39
5.1	Literatuur	39
5.2	Methode	39
5.3	Overwegingen	40
5.4	Eindconclusie	40
6	Aanbevelingen	41
A	Experiment: Machine Learning bij LC Resonantie Circuits met DWT	43
A.1	Opzet	43
A.1.1	Parameters	43
A.1.2	Fout modi	44
A.2	Resultaten	45
A.2.1	Ruwe Resultaten	46
A.2.2	Verwerking	48
B	Experiment: Machine Learning bij LC Resonantie Circuits DFT features	49
B.1	Opzet	49
B.1.1	Parameters	49
B.2	Resultaten	50

C	Experiment: Meting fout modi zonder toegevoegd ingang signaal	53
C.1	Opzet	53
C.1.1	Parameters	54
C.1.2	Fout modi	54
C.1.3	Flow Diagram	55
C.2	Resultaten	56
C.3	Conclusie	57
C.4	Data	58
D	Experiment: Meting fout modi vervanging ingangssignaal	59
D.1	Opzet	59
D.1.1	Parameters	59
D.1.2	Fout modi	60
D.1.3	Stroomdiagram	60
D.2	Resultaten	60
D.3	Conclusie	61
D.4	Data	63
E	Experiment: Test met getrapte metingen	65
E.1	Opzet	65
E.1.1	Parameters	66
E.2	Fout modi	66
E.2.1	Volledig methode 3a	66
E.2.2	Prioriteit op resonantie kring methode 3b	66
E.3	Stroomdiagram	66
E.4	Resultaten	67
E.4.1	Volledig	67
E.4.2	Resonantie kring prioriteit	68
E.4.3	Conclusie	69
E.5	Data (Wordt nog uitgebreid met CT info)	70
F	Gebruikte model voor simulaties	71
G	Gebruikte Code - Omzetten Features	73
H	Competenties	75
H.1	Analyseren	75
H.2	Ontwerpen	75
H.3	Realiseren	76
H.4	Managen	76
H.5	Onderzoeken	76
	Bibliografie	77

Lijst van figuren

2.1	Overzicht werking spoorstroomloop met de richting van de stromen © 2016 IEEE [5]	5
2.2	Illustratie van een B-relais [6]	5
2.3	Vervangingsschema spoor © 2015 IEEE	6
2.4	Schematische weergave SSL [8]	7
2.5	Simpel vervangingsschema SSL [6]	7
2.6	Resonantie Circuit SSL	8
2.7	Vereenvoudigd model spoorstroomlopen	11
2.8	Vereenvoudigd vervangingsschema spoorstroomloop	12
2.9	Overzicht machine learning	14
2.10	Stroom diagram standaard proces ML-Algoritmes	16
2.11	KNN [16]	17
2.12	Voorbeeld scheiding datapunten met support vectors	18
2.13	Niet linear scheidbare data	18
2.14	Toepassen van een kernal functie	18
2.15	Voorbeeld gradient boosted decision tree	19
2.16	Voorbeeld KNN	19
2.17	Morlet wavelet [24]	23
2.18	Mexican hat wavelet [25]	23
2.19	Discrete wavelet transformatie [26]	24
3.1	Voorbeeld van een chirp signaal [27]	26
4.1	t-SNE representatie uitgebreide analyse	36
4.2	Staafdiagram waardes condensator	37
4.3	Staafdiagram waardes spoel	38
A.1	Circuit onder test	43
A.2	t-SNE Representatie van Features	45
B.1	Plot magnitude eerste 100 parameters FT signaal	50
B.2	tSNE visualisatie van de features	50
C.1	Schematische weergave meting	53
C.2	Stroomdiagram meting zonder toegevoegde bron	55
C.3	t-SNE Representatie van Features	56
D.1	Schematische weergave meting	59
D.2	Stroomdiagram meting zonder toegevoegde bron	60
D.3	t-SNE weergaven van features	61
E.1	Schematische weergave meting	65
E.2	Stroomdiagram meting zonder toegevoegde bron	66
E.3	t-SNE weergaven van features	67

E.4	t-SNE weergaven van features	68
F.1	Vervangingschema gebruikt voor de simulaties van de SSL	71

Lijst van tabellen

2.1	Foutmodi bij de spoorstroomloop	10
3.1	Koppelwaardes bij variatie component waardes	27
3.2	Component waardes bij ID	27
3.3	Foutmodi condensator C1 en C2	27
3.4	Foutmodi Spoel	28
3.5	Overzicht van signaalanalyse parameters voor ML-modellen	30
3.6	Overzicht correcte indicaties	32
4.1	Overzicht van voorgestelde meet methodes	33
4.2	Samenvatting resultaten experimenten	33
4.3	Samenvatting resonantiekring fouten score	34
4.4	Tijd om probleem op te lossen bij methode 3a	35
4.5	Tijd om probleem op te lossen bij methode 3b	35
4.6	Overzicht gedefinieerde foutmodi	36
4.7	Samenvattingstabel resultaten uitgebreide analyse	37
A.1	Nauwkeurigheid resultaten ML-Algoritmes	45
A.2	Resultaat matrix DWT analyse	48
A.3	Kerngetallen DWT analyse	48
B.1	Nauwkeurigheid resultaten ML-Algoritmes	51
C.1	Nauwkeurigheid resultaten ML-Algoritmes	56
C.2	Verdeling voorspellingen per zijde bij KNN	57
C.3	Verdeling voorspellingen per zijde bij Neural Network	57
C.4	Neural Network	58
C.5	SVM	58
C.6	Gradient Boosted Decision Tree	58
C.7	K Nearest Neighbors	58
D.1	Nauwkeurigheid resultaten ML-Algoritmes	61
D.2	Verdeling voorspellingen per zijde bij KNN	61
D.3	Verdeling voorspellingen per zijde bij Gradient boosted decision tree	62
D.4	Neural Network	63
D.5	SVM	63
D.6	Gradient Boosted Decision Tree	63
D.7	K Nearest Neighbors	63
E.1	Nauwkeurigheid resultaten ML-Algoritmes	67
E.2	Verdeling voorspellingen per zijde bij KNN	68
E.3	Verdeling voorspellingen per zijde bij c	68
E.4	Nauwkeurigheid resultaten ML-Algoritmes	69
E.5	Verdeling voorspellingen in de resonantie kring of daarbuiten bij KNN	69

E.6	Verdeling voorspellingen in de resonantie kring of daarbuiten bij Gradient boosted decision tree	69
E.7	Resultaten Neural Network	70
E.8	Resultaten SVM	70
E.9	Resultaten Gradient Boosted Decision Tree	70
E.10	Resultaten K Nearest Neighbors	70

Lijst van afkortingen

SSL	Spoorstroomloop
ML	Machine-Learning
KNN	K Nearest Neighbors
t.o.b.s.	Ten Onrechte Bezet Spoor
ES	Elektrische Scheiding

Hoofdstuk 1

Inleiding

Op het Nederlandse spoor hebben kleine vertragingen grote gevolgen. Daarom is het van belang vertragingen zo veel mogelijk te voorkomen. Echter, als toch een storing ontstaat in een van de systemen, is het noodzaak deze zo snel mogelijk op te lossen. Een verbeterde diagnosemethode kan helpen de dienstregeling sneller te hervatten. Daarom heeft de groep treinbeveiligssystemen, onderdeel van de afdeling Assetmanagement, gevraagd of een dergelijke methode ontwikkeld kan worden.

Het spoor is onderverdeeld in secties en op een sectie mag nooit meer dan een trein rijden. De spoorstroomloop is een detectiesysteem dat detecteert of er een trein over een sectie rijdt. Wanneer een sectie bezet is stuurt de spoorstroomloop een signaal naar het sein, zodat de andere gebruikers van het spoor weten dat die sectie bezet is.

In deze afstudeerperiode is gezocht naar een methode waarmee monteurs sneller een diagnose kunnen stellen van defecten componenten van de spoorstroomlopen. Dit document presenteert een nieuwe methode om een diagnose te stellen en kan als opzet worden gebruikt om een hulpmiddel te ontwikkelen voor de monteur.

De spoorstroomloop (SSL) is een volledig analoog systeem. Daarom kan het gedrag van de SSL worden voorspeld aan de hand van elektrotechnische domeinkennis. In essentie draait het in dit project om fouten opmerken in de componenten en configuratie van de SSL. Eerder is al onderzoek gedaan naar het identificeren van defecte componenten. Dit is te lezen in de bronnen [1] tot en met [2], waarin wordt gesproken over machine-learning-algoritmes om veranderingen op te merken en om verschillende mogelijke fouten in een model samen te vatten. In de bronnen [3] en [4] staat foutherkenning in de SSL centraal. Daarbij wordt gebruikgemaakt van een machine learning (ML) model. Het verschil tussen dit onderzoek en deze bronnen is dat in de bronnen een fout wordt voorspeld, terwijl dit onderzoek focust op een fout achterhalen in een component wanneer sprake is van een storing in SSL.

1.1 Aanleiding

Dit document presenteert een onderzoek naar een methode om een diagnose te stellen bij de spoorstroomloop. Het doel is om een methode te ontwikkelen die kan worden uitgevoerd door een apparaat of computer, zodat de monteur niet zelf hoeft te zoeken naar het probleem.

1.2 Probleemstelling

Een spoorstroomloop kan om verschillende redenen te maken krijgen met een storing. De ene keer zit er ijzerslijpsel tussen de ES-lassen en de andere keer is het component defect. Met een visuele controle kan een monteur vaak direct zien wat er mis is, maar soms is het probleem complexer en is het lastig om de defecte component te vinden. Het is van belang om een probleem snel op te lossen, want de tijd die het kost om een defecten component te zoeken, heeft impact op de dienstregeling.

1.3 Doelstelling

Om deze reden wil ProRail bekijken of er een meer efficiënte manier is om problemen met de SSL op te sporen. Dit project focust op een methode om problemen op te sporen met de elektrotechnische componenten van de SSL. In de huidige methode is de kans dat een monteur in een keer de storing vindt 57.6 procent dit is toegelicht in paragraaf 2.5.1. Het doel van het onderzoek is een methode ontwikkelen wat een hogere kans geeft om de storing te vinden.

1.3.1 Onderzoeksvragen

Ten behoeve van het onderzoek zijn drie vragen opgesteld:

1. Welke methoden noemt de literatuur in het onderwerp van het identificeren van verschillende fout modi in elektrische circuits?
 - Hoe kan het spoor worden gemodelleerd in simulatie?
 - Hoe werkt het spoorstroomloop systeem?
 - Welke methodes worden door andere onderzoeken toegepast in fout diagnoses?
 - Hoe werkt het systeem?
 - Hoe kan de SSL gemodelleerd worden?
 - Wanneer (en waarom) treedt er een storing op?
 - Welke machine learning modellen zijn toepasbaar op dit probleem
2. Welke methode kan worden toegepast op de spoorstroomloop?
 - Welke methode moet gebruikt worden om het circuit te meten?
 - Welk model uit een set van vier ML-Modellen die toepasbaar zijn op classificatie modellen is het beste toepasbaar op de SSL?
3. In hoeverre werkt deze methode en heeft het een meerwaarde om hiernaar een vervolgonderzoek uit te voeren?
 - Wat is de nauwkeurigheid na het toepassen van een Monte-Carlo analyse

- Zijn er component waardes waarbij het model minder goed werkt?
- Was de wijze van onderzoek optimaal, en versterkt het de business case voor dit project?

Hoofdvraag: Is er methode waarmee een hogere kans om een defect component te identificeren aan de resonantiecircuiten in de spoorstroomloop, dan de huidige methode die is toegepast door ProRail?

1.4 Methode

Aan de hand van deze drie vragen kan de meerwaarde van dit project worden ingeschat. Daarnaast brengen de vragen structuur in het verslag. Om te beginnen wordt literatuuronderzoek gedaan naar de verschillende mogelijkheden om fouten te diagnosticeren in circuits. Daarbij wordt achtergrondkennis gebruikt, die nodig is om de in de literatuur beschreven methodes toe te passen. Op basis van de literatuur worden methodes opgesteld, getest en vergeleken. De methode die het beste werkt, wordt uitgebreid geanalyseerd. Deze analyse gaat in op de prestaties van het systeem. Uiteindelijk blijkt in hoeverre de methode werkt en of het een meerwaarde heeft om vervolgonderzoek te doen naar bijvoorbeeld een product waarbij deze methode toegepast kan worden.

1.4.1 Afbakening

Om het project binnen de daarvoor gestelde termijn af te kunnen ronden, zijn de onderstaande grenzen opgesteld.

1. Dit project is theoretisch van karakter. De methode wordt niet getest op de werkelijke installatie, dit in verband met COVID-19.
2. In dit project wordt enkel gezocht naar een methode om fouten sneller op te sporen. Er wordt dus geen totaaloplossing ontwikkeld.
3. Het onderzoek focust slechts op één ontwerp van de spoorstroomloop.

1.5 Leeswijzer

In deze paragraaf zal worden uitgelegd hoe dit document is opgebouwd en waar de antwoorden op de gestelde onderzoeksvragen te vinden zijn.

Eerst zal in dit rapport het theoretisch kader van de gebruikte machine-learning-modellen en de spoorstroomloop worden gegeven, dit gebeurd in hoofdstuk 2. Daarbij zal eerst worden ingegaan op de werking van de SSL, daar zullen ook de vragen worden beantwoord over de werking van de SSL en hoe het gemodelleerd moet worden. In dat hoofdstuk zal ook worden behandeld wanneer er een storing optreedt in de SSL. Verder zal dieper in worden gegaan op de werking van de gebruikte ML-modellen en op machine learning in het algemeen. Tot slot wordt het theoretisch kader afgesloten een vergelijking tussen voorgestelde methodes in de literatuur, en wordt ook vanuit die methodes een kader opgesteld voor de methode die ontwikkeld wordt in dit verslag.

Daarna zal in hoofdstuk 3 verschillende methodes worden gepresenteerd op basis van de opgedane kennis uit de literatuur, hoe het circuit gemeten dient te worden. Verder zal in hoofdstuk 3 ook worden beschreven hoe het signaal geanalyseerd wordt, en hoe een keuze gemaakt gaat worden tussen de vier gekozen ML-modellen.

In hoofdstuk 4 zullen de resultaten worden gegeven van de vier methodes die zijn voorgesteld zijn in hoofdstuk 3. Die zullen daarna ook met elkaar worden vergeleken, dan zal resulteren in de keuzen de methode van meten in combinatie met een van de ML-modellen die het beste toepasbaar is op de SSL. Deze methode zal dan nog wat uitgebreider getest worden en er zal ook worden onderzocht bij welke component waardes het model slechter werkt.

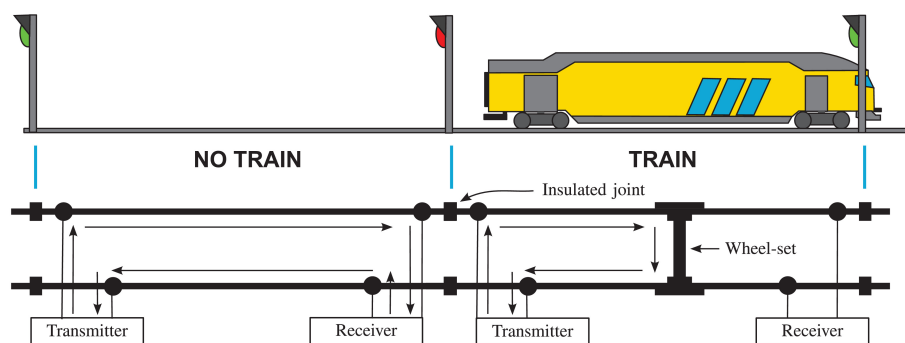
In Hoofdstuk 6 zullen aan de hand van de resultaten aanbevelingen worden gegeven over het nut van vervolgonderzoek met de voorgestelde methodes.

Hoofdstuk 2

Theoretisch kader

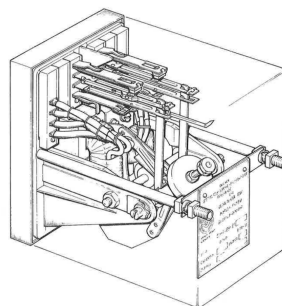
Twee onderwerpen zijn nodig om een begrip te krijgen van de achterliggende werking van de methodes, namelijk de werking van de gebruikte ML-algoritmes en de technische aspecten van de spoorstroomloop. Dit hoofdstuk bespreekt beide onderwerpen.

2.1 Spoorstroomlopen



FIGUUR 2.1: Overzicht werking spoorstroomloop met de richting van de stromen © 2016 IEEE [5]

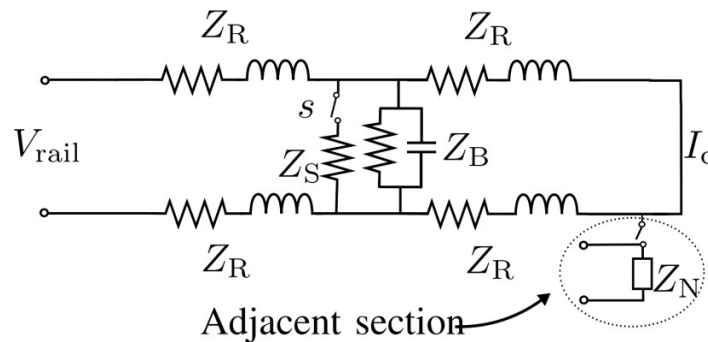
Een spoorstroomloop bestaat uit drie componenten: de zender, het spoor en de ontvanger. Daarnaast heeft de trein een functie in de werking. Het concept werkt als volgt: de zender zet een spanning tussen de twee spoorstaven. Wanneer er geen trein op het spoor aanwezig is, is er een spanningsverschil tussen de twee spoorstaven bij de ontvanger. Die spanning stuurt een B-relais aan, zodat deze aangetrokken wordt. In deze positie zal een veilig sein worden getoond. Wanneer een trein in een sectie rijdt, wordt deze verbinding kortgesloten voordat het de ontvanger bereikt. Daarop valt het B-relais af en wordt een rood sein getoond.



FIGUUR 2.2: Illustratie van een B-relais [6]

Een baanvak is opgeknipt in verschillende blokken en elk blok is onderverdeeld in secties. Elke sectie bevat de onderzochte schakeling en elk blok heeft een sein. Een schematisch overzicht staat weergegeven in figuur 2.1. Daarin is een situatie afgebeeld waarin er geen trein is (te zien aan linkerkant). Hier is dus niks kortgesloten tussen de zender en de ontvanger en wordt een veilig sein getoond. Aan de rechterkant van het figuur is sprake van een situatie met een trein op het spoor. Hier is te zien dat de verbinding is kortgesloten over de as van de trein en dat dus een rood sein wordt getoond aan het begin van het blok.

Spoor als een transmissielijn



FIGUUR 2.3: Vervangingsschema spoor © 2015 IEEE

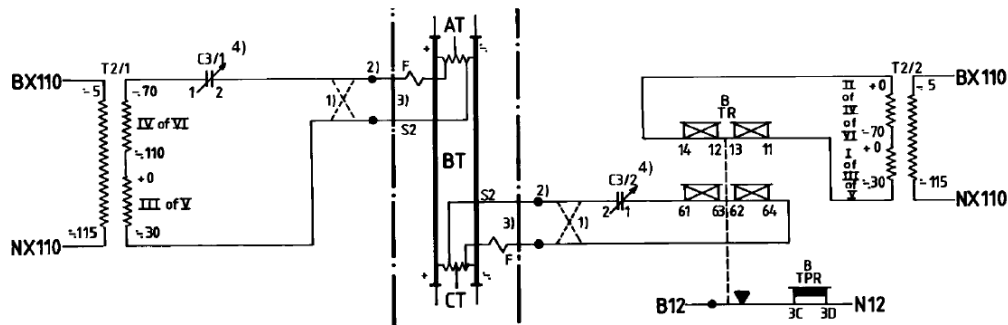
Zoals beschreven in bron [4], kan de verbinding tussen de zender en ontvanger gemodelleerd worden zoals in figuur 2.3. Daarbij is V_{rail} de spanning die door de zender op het spoor wordt gezet. Het signaal aan de kant van de ontvanger is dan stroom I_c . Het isolerende plaatje dat tussen de secties wordt geplaatst, zorgt ervoor dat het oscillerende signaal van 75 Hz dat geproduceerd wordt door de zender, niet overloopt naar het andere blok. Om de grote stromen die geproduceerd worden door de tractie-units van de treinen door te laten, zijn tussen de blokken rail-spoelen aangebracht. Deze impedantie zorgt ervoor dat de gelijkstroom van de tractie-units vrij kan stromen tussen de blokken.

Om een beter begrip te krijgen van welke effecten er bij de verbinding over de rail meespelen, wordt hieronder uitgelegd wat de effecten zijn van de elektrotechnische aspecten van de rails, zoals de ballastweerstand.

Rails liggen op de grond met dwarsliggers ertussen. Dit zorgt ervoor dat tussen de rails stromen kunnen lopen. Deze stromen kunnen lopen via de dwarsliggers, de bevestigingen en de grond [7]. Deze ballastimpedantie wordt aangegeven met Z_B . De ballastimpedantie is afhankelijk van de constructie en van weersomstandigheden. Zo is de ballastimpedantie in regenachtige omstandigheden een stuk lager dan wanneer het droog is. Volgens [8] varieert de impedantie tussen de $1.5\Omega km$ tot $\infty\Omega km$. Verder heeft elke spoorstaaf een eigen impedantie. Deze wordt gerepresenteerd door Z_R .

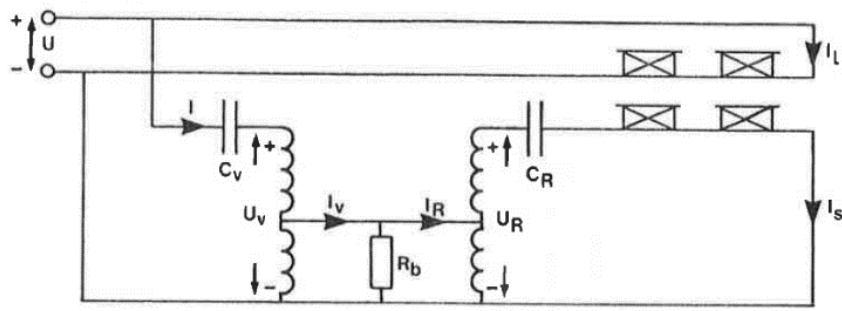
In figuur 2.3 wordt ook gesproken van een Z_S , deze wordt in dit project niet meegenomen. Z_S is namelijk de impedantie die wordt veroorzaakt door de trein shunt. Maar omdat in dit project wordt uitgegaan van een vrije sectie, dus zonder trein op het spoor. Is er geen trein shunt aanwezig.

Gehele systeem



FIGUUR 2.4: Schematische weergave SSL [8]

In figuur 2.4 staat een schematische weergaven van een spoorstroomloopvariant die ProRail toepast op het Nederlandse spoor. Het systeem werkt als volgt: aan de linkerkant wordt een 110v wisselspanning van 75Hz geleverd. Deze sluit aan op een spoel met meerdere secundaire aftakkingen, afgebeeld als T2/1. C3/1 zorgt er samen met C3/2 voor dat de fasedraaiing wordt gecompenseerd die wordt veroorzaakt door het spoor en de spoorspoelen. Dit zorgt er uiteindelijk voor dat het werkelijke vermogen het hoogst is, zodat bij de spoelen die in het b-relais zitten, de hoogste koppel wordt gehaald [6].



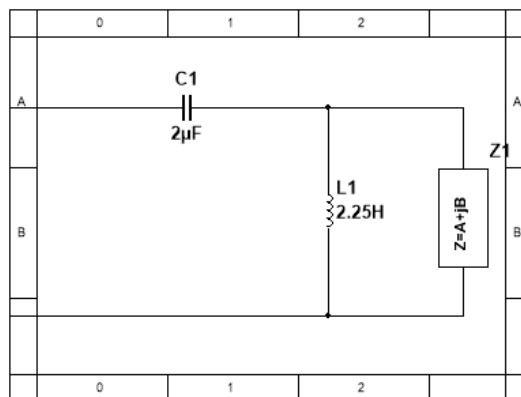
FIGUUR 2.5: Smpel vervangingsschema SSL [6]

Om dit in detail uit te leggen, kan het bovenstaande schema worden vereenvoudigd tot het schema in figuur 2.5. Hierbij is aangenomen dat het spoor geen weestand heeft en wordt het systeem bekeken nadat de spanning is getransformeerd. R_b stelt hier de ballastweerstand voor, omdat deze wel effect heeft op de werking van het circuit. Zoals hierboven benoemd, is het de bedoeling dat een maximaal koppel wordt bereikt over het spoorrelais. Het circuit is verder te vereenvoudigen door aan te nemen dat de transformatie die plaatsvindt bij de railspoel, ideaal is met een spoel parallel aan de ingang.

2.2 Resonantiecircuits

De spoorstroomloop bestaat uit verschillende componenten. Een van de belangrijkste onderdelen van de SSL, zijn de twee resonantiecircuits die zich aan de aanstuur- en relaiszijde bevinden. Deze paragraaf gaat in op de werking van deze resonantiecircuits.

Als vanuit de aanstuurzijde wordt gekeken, kan het resonantiecircuit worden beschouwd zoals weergegeven in figuur 2.6. Daarbij stelt C_1 de capaciteit van C3/1 voor, L_1 de inductie van de spoorspoel en Z_1 de rest van het circuit.



FIGUUR 2.6: Resonantie Circuit SSL

Elektrische resonantie betekent in een LC-circuit dat bij de resonantiefrequentie de imaginaire vector van het circuit nul wordt. Wat er in praktische zin gebeurt, is dat op enig moment het magnetische veld van de spoel afbouwt en een stroom creëert. Die stroom laadt dan de condensator op. Wanneer de spoel ontladen is en de condensator vol, ontladt de condensator weer en creëert wederom een stroom die een magnetisch veld om de spoel creëert. In het ideale geval, dus zonder weerstand, blijft dit proces zich eeuwig herhalen. De relatie tussen de resonantiefrequentie en de waardes van L en C is beschreven in vergelijking 2.1.

$$\omega_f = \frac{1}{\sqrt{LC}} \quad (2.1)$$

Waar:

L = inductieve waarde in H

C = capacatieve waarde F

ω_f = resonantie hoekfrequentie

Een andere belangrijke parameter die wordt geassocieerd met de resonantiecircuit, is de Q-factor. De Q-factor is een dimensieloze parameter die een maat geeft hoe ongedempt een systeem is. Wanneer de Q-factor groter is zal dat resulteren in kleinere bandbreedte bij de oscillatie. In essentie geeft deze maat dus aan de ratio tussen de energie opgeslagen en de energie die wordt gedissipeerd. Deze relatie is beschreven in vergelijking 2.2.

$$Q = \frac{1}{R} \sqrt{\frac{L}{C}} \quad (2.2)$$

Waar:

R = Weerstand in Ω

2.3 Foutmodi

Zoals beschreven in de literatuur, is al veelvuldig onderzoek gedaan naar diagnoses stellen bij elektrische circuits. Echter, een diagnose stellen waarbij van buiten het circuit een uitspraak moet worden gedaan over welke specifieke component defect is, is lastiger. De voornaamste reden hiervoor is dat de waarde waar een component in ligt, toleranties heeft. Een andere reden is dat de waardes kunnen afwijken van de toleranties, maar dat de functie van het circuit dan nog steeds behouden blijft. Voor het opstellen van de foutmodi moet ook worden gekeken bij welke componenten het circuit in storing treedt [9].

Uit een analyse van de storingsmeldingen gepresenteerd in tabel 2.1, blijkt dat vanuit de meeste storingen een onbekende oorzaak hebben. Dit betekent dat toen de monteur aankwam, de storing al was verholpen. Wat de reden van deze storing is, is dan meestal niet meer te achterhalen.

Voor dit onderzoek zijn de storingen die worden veroorzaakt door componenten en de instelling van het systeem het meest interessant, omdat deze kunnen worden afgeleid van een fout die wordt veroorzaakt door het gebruikte circuit. Fouten die worden veroorzaakt door een van de andere categorieën worden buiten beschouwing gelaten, omdat ze veroorzaakt worden door externe factoren, zoals weersomstandigheden, menselijke fouten en verwerking.

In dit onderzoek wordt ervan uitgegaan dat het systeem in storing treedt wanneer sprake is van een ten onrechte bezet spoor (t.o.b.s.). Dit is dus wanneer een rood sein wordt getoond, maar er geen trein aanwezig is op de sectie. De meldingen in tabel 2.1 zijn dan ook meestal reacties op t.o.b.s. Echter, wanneer een t.o.b.s. niet voor een vertraging zorgt, dan wordt er geen melding van gedaan. Als het circuit wordt bekeken, dan is aannemelijk dat het systeem in storing treedt wanneer het systeem niet in staat is de spoel in het B-relais op een correcte manier van energie te voorzien. Dit zorgt er dan voor dat het relais afvalt. Om een goed systeem te creëren dat deze zaken kan diagnosticeren, moet er begrip zijn van de foutmeldingen en van hoe deze ontstaan.

TABEL 2.1: Foutmodi bij de spoorstroomloop

Rijlabels	Som van No.
Onbekend	431
Kabel	185
ES-las splinter/slijpsel	156
Vandalisme (muntje)	111
Werkzaamheden derden	71
<i>Instelling</i>	53
Kortsluiting door overige objecten	42
<i>Condensator</i>	23
<i>Trafo</i>	23
<i>Zekering</i>	22
<i>Railspoel</i>	16
CR/VTB	15
Bedrading (incl. klemmen/pluggen)	14
Vervuiling	13
Bliksem	7
DSV	5
Overig	5
Energievoorziening	3
ZKL	3
<i>Weerstand</i>	1
Totaal	1199

2.3.1 Theoretisch bepalen van de koppel

Om te begrijpen waarom bij een SSL het spoorrelais niet wordt aangetrokken, moet eerst duidelijk zijn waarom een relais afvalt wanneer bijvoorbeeld de instelling niet goed is geconfigureerd. Verschillende systemen op het spoor worden aangestuurd door het spoorrelais. Zoals alle relais, moet het spoorrelais ook worden bekrachtigd om het te schakelen.

Spoorrelais

Het spoorrelais bestaat uit twee spoelen die onafhankelijk van elkaar een magnetisch veld kunnen opwekken. Het ene paar spoelen heeft een hoge impedantie en wordt op de voedingsbron aangesloten. Dit wordt de lokale fase genoemd. Het andere paar heeft een lage impedantie en wordt aangesloten op de spoorsectie. Dit wordt de spoorfase genoemd. Het relais wordt pas opgetrokken als beide fasen worden bekrachtigd worden. Daarbij is het van belang dat de signalen dezelfde frequentie hebben en dat er een bepaald fase verschil is.

In essentie komt het erop neer dat de maximale koppel bereikt moet kunnen worden in het spoorrelais. Daarbij moet sprake zijn van een verschil van 90 graden tussen de stroom die door de spoorfase loopt en de stroom die door lokale fase loopt. Bron [6] gaat in op waarom dit nodig is.

De relatie tussen de koppel en de spoorfase en de lokale fase is gegeven in vergelijking 2.3.

$$K = I_l I_s \sin \phi \quad (2.3)$$

Waar:

K = koppelwaarde van het spoorrelais

I_l = lokale fasestroom

I_s = spoorfasestroom

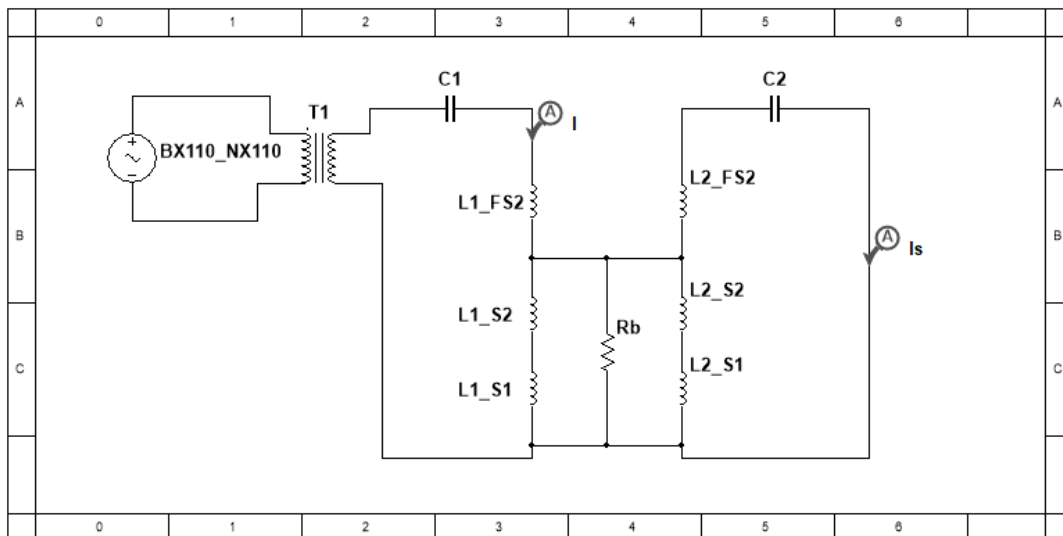
ϕ = Faseverschil I_l en I_s vanuit I_l

Omdat I_l een vaste waarde heeft, kan deze weggelaten worden om de relatie aan te geven tussen de verschillende parameters. Dit is te zien in vergelijking 2.4.

$$K = I_s \sin \phi \quad (2.4)$$

Spoorstroomloopmodel

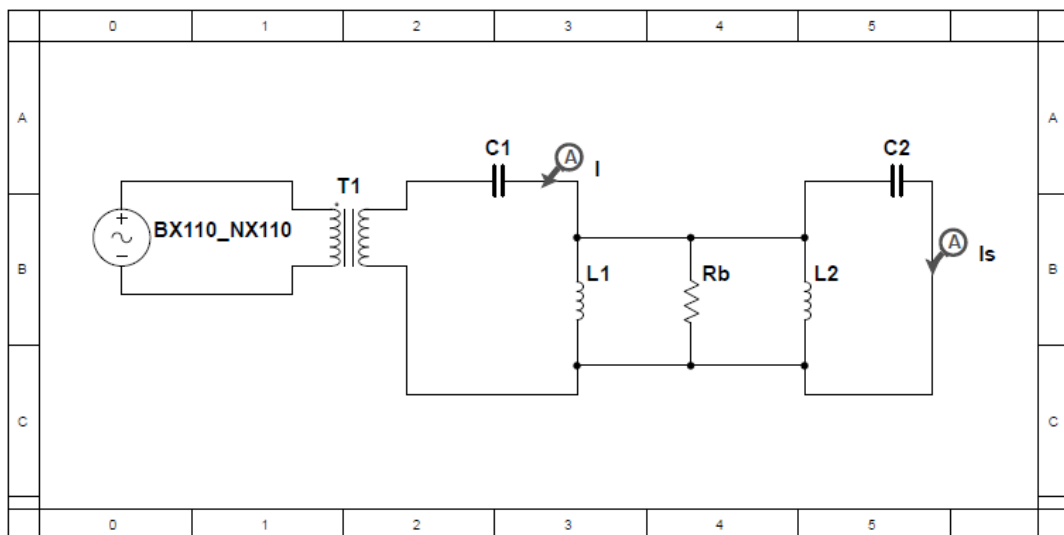
In figuur 2.7 is in een model weergegeven hoe de spoorstroomloop gemodelleerd kan worden om de verschuiving in fase en de stroom te berekenen aan de relaiskant. Dit circuit kan nog verder worden vereenvoudigd, omdat L1 niet alleen een inductieve eigenschap heeft, maar ook een transformatoreigenschap. Dit is zodat de spanning op het spoor naar beneden wordt getransformeerd.



FIGUUR 2.7: Vereenvoudigd model spoorstroomlopen

Dit circuit kan nog verder worden vereenvoudigd omdat L1 naast een inductieve eigenschap ook een transformator eigenschap heeft. Die transformatorverhouding is 80:1 bij de spoorspoelen. Omdat aan beide kanten wordt getransformeerd met dezelfde verhouding, hoeft alleen de ballastweerstand getransformeerd te worden. Omdat de verhouding 80:1 is, is de transformatorverhouding 80^2 . Dus R_b wordt $80^2 R_b = 6400 R_b$. Na deze transformaties krijgt het vervangingsschema de vorm zoals in figuur 2.8.

Door het circuit door te rekenen, kan berekend worden wat de faseverschuiving bij I_s . Hieronder staat een voorbeeld van hoe die verschuiving kan worden berekend. De volgende waarden worden gehanteerd voor het vervangingsschema in figuur 2.8:



FIGUUR 2.8: Vereenvoudigd vervangingsschema spoorstroomloop

$$\begin{aligned}
 C_1 &= 2\mu F \\
 L_1 &= 2.20H \\
 L_{1ESR} &= 0.1\Omega \\
 R_b &= 15 \cdot 80^2 = 96000\Omega \\
 C_2 &= C_1 \\
 L_2 &= L_1 \\
 L_{2ESR} &= L_{1ESR} \\
 E_{in} = BX110_NX110 &= 130V \ 75Hz \ \text{sinus}
 \end{aligned}$$

Om te beginnen worden de complexe impedanties van de verschillende componenten uitgerekend. Daarbij worden de volgende vergelijkingen gebruikt:

$$X_C = \frac{1}{j\omega C} \quad (2.5)$$

$$X_L = L_{ESR} + j\omega L \quad (2.6)$$

$$\omega = 2\pi f \quad (2.7)$$

waar:

X_C = reactantie condensator
 X_L = reactantie spoel
 j = imaginaire unit
 f = Frequenties in Hz
 ω = hoekfrequentie

Als de bovenstaande vergelijkingen worden uitgerekend, komen daar de volgende impedantiewaardes uit: $X_{C1} = X_{C2} = 1061\Omega \angle -90^\circ$ en $X_{L1} = X_{L2} = 1037\Omega \angle 89.99^\circ$.

Om I_s te kunnen uitrekenen, is het nodig om de totale stroom I uit te rekenen. Daarvoor moet de ingangsimpedantie eerst vanaf de bron worden berekend. Daarbij worden de volgende waardes gedefinieerd:

$$Z_1 = X_{L1} // R_b // X_{L2} // X_{C2} \rightarrow \frac{1}{Z_1} = \frac{1}{X_{L1}} + \frac{1}{R_b} + \frac{1}{X_{L2}} + \frac{1}{X_{C2}} \quad (2.8)$$

waar:

Z = complexe impedantie

Vergelijking 2.8 resulteert in de volgende waarde voor dit circuit: $Z_1 = 10.89 + 1013.39j$. Nu de ingangsimpedantie bekend is, kan de totale stroom door het circuit worden berekend.

$$Z_t = X_{C1} + Z_1 \quad (2.9)$$

Vergelijking 2.9 resulteert in de volgende waarde voor dit circuit $Z_t = 10.89 - 47.64j$. Nu de ingangsimpedantie bekend is kan de totale stroom door het circuit worden berekend.

$$I = \frac{E}{Z} = \frac{E_{in}}{Z_t} \quad (2.10)$$

De bovenstaande vergelijking resulteert voor de in dit voorbeeld gekozen waardes in $I = 2.66 \angle 77.12^\circ A$. Daarna is de spanning die over de parallel geschakelde componenten bekend, Deze is:

$$E_{Z1} = IZ_1 \quad (2.11)$$

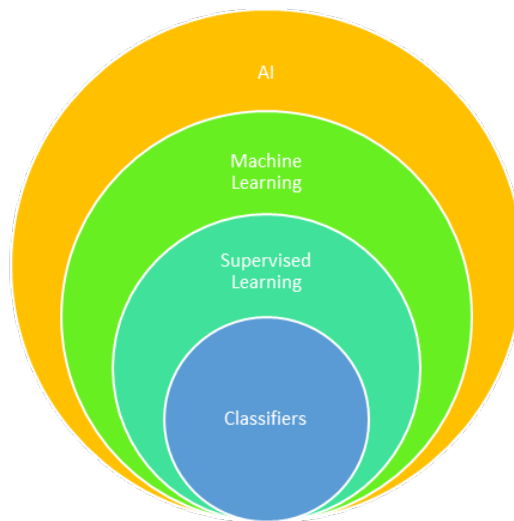
Vergelijking 2.11 resulteert in $E_{Z1} = 2696 \angle 166.51^\circ V$, omdat alle componenten parallel zijn geschakeld, zijn de spanningen over alle componenten gelijk. Daardoor is direct duidelijk welke spanning en dus ook welke stroom (I_s) door C2 loopt.

$$I_s = \frac{E_{Z1}}{X_{C2}} \quad (2.12)$$

Dus de stroom is $I_s = 2.54 \angle -103.49^\circ A$. Nu de hoek van de stroom aan de relaiszijde bekend is, kan de koppel worden berekend. Het is bekend dat de lokale fase 65° achterloopt op de stroom I . Dit betekent dat de lokale fasestroom 115.61° voorloopt op de spoorfasestroom. Omdat een positief koppel nodig is om het relais correct aan te sturen, wordt de spanning omgewisseld aangesloten aan het relais. Dit betekent dat de fase 180° gedraaid is. Daardoor is in dit voorbeeld het faseverschil ten opzichte van I_s 64.38° . Dit betekent dat de koppelmaat voor dit circuit uitgerekend is met vergelijking 2.4 $K = 2291.01$.

2.4 Machine learning

Uit de literatuur blijkt dat verschillende technieken berusten op het gebruik van machine learning [10]. Machine learning is een technologie die de afgelopen jaren zeer populair is geworden, omdat het mensen en bedrijven in staat stelt om grote hoeveelheden data te analyseren en om er patronen in te herkennen. Daarbij kan het gaan om een grote verscheidenheid aan data, zoals persoonlijke informatie, financiële transacties, biologische reacties of sensordata uit elektronische systemen. ML-algoritmes zijn een efficiënte manier om modellen te maken van data, waardoor vrij eenvoudig parameters voorspeld kunnen worden. Voorbeelden van toepassingen zijn te vinden in bron [10]. Binnen machine learning is sprake van een



FIGUUR 2.9: Overzicht machine learning

hiërarchie, zoals te zien in figuur 2.9. In dit project wordt gebruikgemaakt van supervised learning. Hierbij leert een algoritme een bepaalde relatie tussen data door te laten zien wat de verwachte uitkomst is. Bij deze techniek wordt bij het “trainen” het algoritme dus meegegeven wat de uitkomst is van de gegeven set van parameters. Wanneer bijvoorbeeld iemands kredietscore bepaald moet worden, krijgt het algoritme in de trainingsfase voorbeelden van mensen die hun schulden vaak niet kunnen afbetalen en mensen die dit wel kunnen. Na het trainen, kan het model met een bepaalde betrouwbaarheid voorspellen hoe goed iemand zijn schulden kan afbetalen.

Een andere tak van machine learning is unsupervised learning. Hier wordt in de trainingsfase niet meegegeven wat de uitkomst is van een combinatie van parameters, maar moet het algoritme leren van de onderliggende patronen van een bepaalde dataset. Een voorbeeld is voorspellen wat iemand graag wil kopen in een webshop.

Van de besproken technieken wordt in dit onderzoek enkel gebruikt voor supervised learning. En binnen dat domein zal dan worden gekozen voor classificatiemodellen. Dergelijke modellen zetten inputdata om in een categorie. Dit is dus iets anders dan een regressiemodel, waar een numerieke waarde de uitkomst is. Om te begrijpen hoe deze ML-algoritmes kunnen bijdragen diagnoses stellen bij spoorstroomlopen, is het van belang om te begrijpen hoe een dergelijk probleem kan worden aangepakt.

Bron [11] geeft een samenvatting van de mogelijke manieren om fouten op te sporen in analoge circuits.

2.4.1 Waarom machine learning?

Voordat wordt ingegaan op de details van de gebruikte ML-modellen, is het van belang om aandacht te besteden aan de waarom-vraag. De vraag is namelijk waarom machine learning beter zou zijn dan een wiskundig model dat gebaseerd is op de overdracht van het systeem?

Machine learning is enkel een middel om modellen te maken. Daarbij is het nodig om te weten wat er gebeurt in een dergelijk model en waarom er wordt gekozen voor een bepaald model. Het heeft geen zin om zomaar de volledige data van een signaal in te voeren in een ML-model. Het model levert dan waarschijnlijk geen goede resultaten op.

De vraag is dan of het wel nuttig is om dieper in te gaan op ML in het geval van de SSL. De gebruikte methode gaat namelijk veel te maken hebben met signaalanalyse. In principe is het geen probleem om ML-modellen te gebruiken om signalen te analyseren. Echter, om een ML-model optimaal te gebruiken, wordt het signaal omgezet in karakteriserende parameters voor het signaal. In hoofdstuk 3.4 wordt dieper ingegaan op de manier waarop het signaal wordt geanalyseerd.

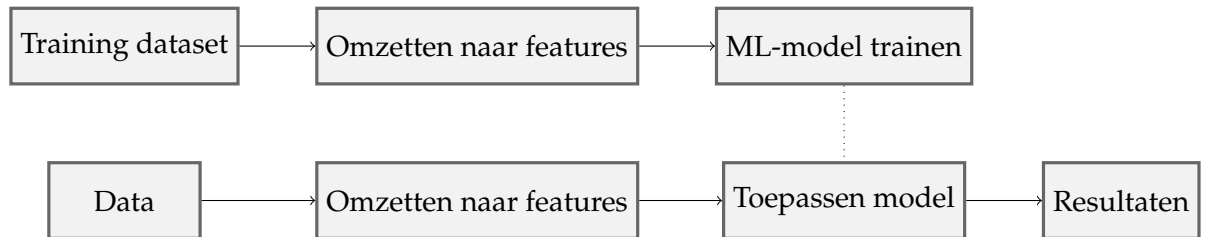
Als een ML-model wordt afgewogen tegen bijvoorbeeld een wiskundig model van een systeem, dan biedt ML een voordeel, omdat het onbekende variabelen kan meenemen. De SSL lijkt op het eerste gezicht eenvoudig, maar het spoor is een variabel en slecht te voorspellen component. Het spoor is te zien als transmissielijn, zoals uitgelegd in paragraaf 2.1. In deze paragraaf is uitgelegd dat steeds veel verandert door verscheidene moeilijk te voorspellen parameters, zoals het weer. In het proces van een ML-model maken, kan vrij eenvoudig worden meegenomen dat sprake is van een groot bereik waartussen de resultaten kunnen liggen. Bij het opstellen van een traditioneel wiskundig model kan dit voor veel onbekende variabelen zorgen die dan benaderd moet worden, waardoor het model minder goed presteert onder de verschillende omstandigheden van de SSL.

Waarom een ML-model hier een betere toepassing is, is omdat het gespecialiseerd is in data classificeren. Dit betekent dat de wiskunde die erin gebruikt wordt, gemaakt is om verschillende mogelijkheden af te wegen. Daarentegen zijn traditionele wiskundige modellen meer gebaseerd op uitkomsten voorspellen, dus terug berekenen welke categorie lastig kan zijn. In dit project wordt ervoor gekozen om de literatuur te volgen en gebruik te maken van een ML-model.

2.4.2 Gebruik van een ML model

Hoewel de manier waarop een ML-algoritme wordt toegepast per soort verschilt, moeten voor de meeste in dit project gebruikte ML-algoritmen een aantal dezelfde stappen worden gezet. Deze paragraaf bespreekt deze stappen. De stappen zijn alleen van toepassing op supervised learning-algoritmes.

Overzicht



FIGUUR 2.10: Stroom diagram standaard proces ML-Algoritmes

In figuur 2.10 staat een globaal overzicht van de stappen. Het stroomdiagram kent twee fasen: de trainingsfase en de toepassingsfase.

Trainingsfase In de trainingsfase worden de parameters van het model bepaald. Hierbij is veel trainingsdata nodig die een goede representatie is van de dataset die gebruikt gaat worden. Specifiek moet er dus voor worden gezorgd dat hetgeen gemeten wordt, correleert met de resultaten. Dus wanneer parameter x stijgt of daalt, moet dat effect hebben op uitkomst y .

Om een model nog optimaler te gebruiken, kunnen data omgezet worden in correlerende parameters. Hierbij valt te denken aan de lengte en het gewicht van iemand omzetten naar een BMI. Een ander voorbeeld is een fouriertransformatie op een signaal toepassen, zodat data uit het frequentiedomein beschikbaar is. Uiteindelijk moet ervoor worden gezorgd dat de verschillende categorieën die het model moet herkennen, van elkaar gescheiden kunnen worden.

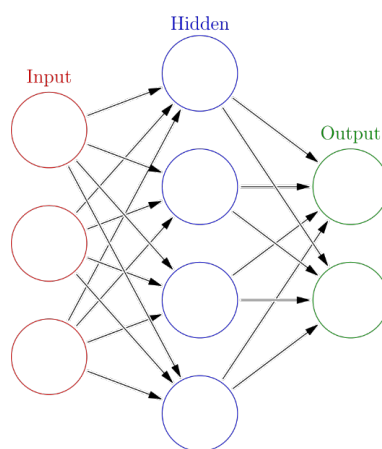
Nadat de dataset gereed is, kan een model worden aangeboden aan een trainingsalgoritme van een ML-model. De meeste modellen linken datapunten aan elkaar en stellen daar voorwaardes of gewichten aan. Categorie n is dan bijvoorbeeld behaald wanneer feature x groter is dan y . In het genoemde voorbeeld wordt de y bepaald via een optimalisatiefunctie. Daarbij wordt wanneer een willekeurige waarde wordt gekozen, gekeken in welke mate deze afwijkt van de gewenste waarde. Als bekend is in welke richting de parameter afwijkt, kan waarde y naar boven of naar onder worden bijgesteld. Uiteindelijk zal de waarde niet meer veranderen en dan is waarschijnlijk de beste configuratie behaald. In bron [12] staan meer voorbeelden van het trainen van modellen.

Uitvoeringsfase Nadat een model getraind en getest is, kan het gebruikt worden om data te classificeren. Daarbij moet bijvoorbeeld het signaal om dezelfde manier worden omgezet als in de trainingsfase.

2.4.3 Classificatiemodellen

Deze paragraaf bespreekt de werking van een classificatiemodel. Daarbij zijn verschillende modellen uiteengezet die het mogelijk maken om aan de hand van een set parameters te voorspellen wat de uitkomst is. In de literatuur over fout diagnose bij elektrische circuits [1], [2], [13]–[15] staan vooral kunstmatige neurale netwerken en support vector machines centraal.

Kunstmatige neural network



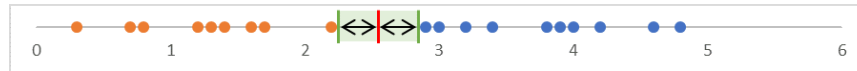
FIGUUR 2.11: KNN [16]

Een neural network is een van de meest bekende ML-algoritmes. Het model is gebaseerd op een abstracte en versimpelde versie van het menselijk brein. Een simpele vorm bestaat uit drie lagen. In de input-laag worden alle waarden die zijn gemeten en bepaald, ingevoerd. De output-laag geeft het antwoord op de vraagstelling waarin het netwerk getraind is. Tussen deze twee lagen zit de *hidden* laag. Al deze lagen bestaan uit *nodes* en op basis van de waarden die zijn ingevoerd, wordt aan de hand van gewichten tussen de *nodes* bepaald wat de waarde is van de volgende node. Een node uit de *hidden* laag is dus verbonden met alle *nodes* van de *input*-laag. Deze node neemt de waarde aan van de som van de waarde van de *input*-waarde, vermenigvuldigd met het gewicht van de verbinding.

Support Vector Machines

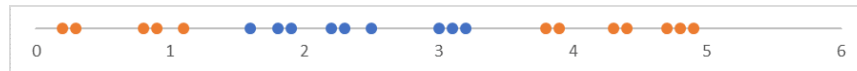
Eéndimensionale data lijkt op het eerste gezicht niet zo moeilijk. Als een parameter een bepaalde waarde heeft gehaald, dan valt deze in een bepaalde categorie. Deze scheiding wordt in de literatuur een *decision boundry* genoemd, zie bron [17] en [18].

Het support-vector-algoritme geeft een algoritmische oplossing om de plaats van de scheidslijn tussen twee categorieën te bepalen. De werking is eenvoudig te begrijpen. Eerst worden bij de uiterste waarden van een categorie support vectors geplaatst. De plaats van de scheidslijn is dan de plaats met de maximale afstand tot deze support vectors.



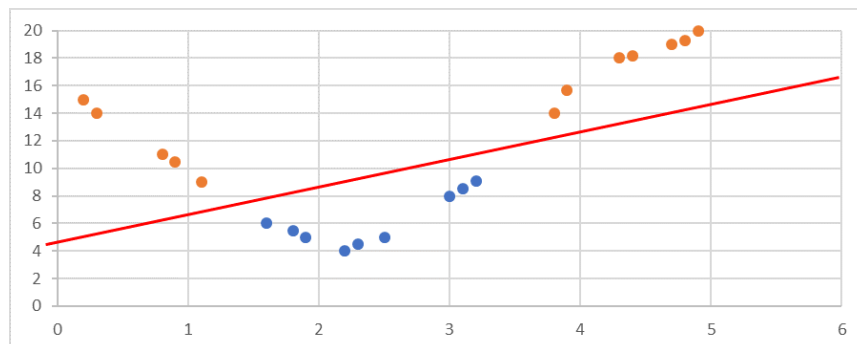
FIGUUR 2.12: Voorbeeld scheiding datapunten met support vectors

Bij een dataset zoals in figuur 2.13, ontstaat een niet-lineair probleem. Hier kan namelijk niet eenvoudig worden gezegd dat wanneer de waarde van de parameter groter is dan 1.5, deze bij categorie blauw hoort. Er zouden twee decision boundaries getekend kunnen worden, maar daarbij moet onthouden worden dat hier een simpel ééndimensionaal voorbeeld wordt gegeven.



FIGUUR 2.13: Niet linear scheidbare data

De vraag is hoe een lineaire scheiding kan worden aangebracht in deze dataset. Het antwoord is door de dimensies op te schalen waarin gewerkt wordt. Als voorbeeld kan gedacht worden dat men een parabool pakt en over die parabool de datapunten plot. Als dan eendimensionaal van onder wordt gekeken ziet men nog steeds het figuur zoals weergegeven in figuur 2.12. Wanneer de data wordt geprojecteerd op de parabool ontstaat een figuur zoals weergegeven in figuur 2.13.

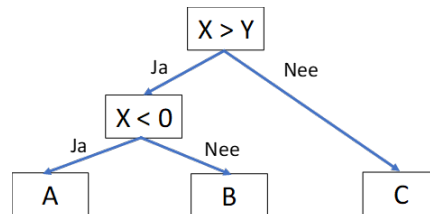


FIGUUR 2.14: Toepassen van een kernal functie

Het aantal dimensies wordt verhoogd via een kernalfunctie [17]. Dit heeft overeenkomsten met convolutie toepassen op een signaal, maar dan met een dataset. Het uiteindelijke resultaat is een scheidingslijn waarmee kan worden bepaald of een datapunt in een bepaalde categorie valt.

Gradient boosted decision tree

Dit model bestaat uit twee onderdelen: een decision tree en het “boosten” van de resultaten van die decision tree. Een decision tree is een reeks condities die uiteindelijk uitkomen op een categorie. Als er bijvoorbeeld drie categorieën zijn (A, B en C) en er zijn twee parameters (X en Y), dan is de vraag welke condities nodig zijn om tot een categorie te komen. In figuur 2.15 staat een voorbeeld van een decision tree. Wanneer dit model wordt getraind, wordt de afweging van elke node berekend en de conditie van de verbinding tussen deze nodes. Boosten betekent dat meerdere



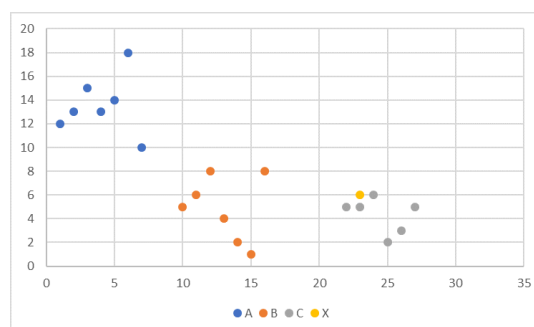
FIGUUR 2.15: Voorbeeld gradient boosted decision tree

kleinere modellen achter elkaar worden geplaatst, die eindelijk gescoord en vergeleken worden. Zo worden fouten verkleind en wordt ervoor gezorgd dat het model niet te specifiek wordt getraind op de trainingsdataset. Door het model te boosten wordt voorkomen dat het model te specifiek wordt op een categorie.

K-nearest neighbors

K-nearest neighbors (KNN) is een van de meest eenvoudige ML-modellen. De werking ervan berust op ingevoerde data vergelijken met de al bekende en gelabelde trainingsdata. Eenvoudig gesteld probeert dit algoritme een voorspelling te baseren op de datapunten die rondom het ingevoerde datapunt liggen. In figuur 3.15 staat een voorbeeld van een dataset met twee variabelen. Hier zijn de datapunten A, B en C bekend, maar moet achterhaald worden welke categorie datapunt X is. KNN kijkt dan naar de punten die ernaast liggen. Dus in dit geval is datapunt X waarschijnlijk categorie C.

Bij dit algoritme is K een belangrijke variabele. Deze variabele geeft aan hoeveel datapunten om de ingevoerde data moeten worden meegenomen om een voorspelling te maken. Bij $K = 1$ wordt enkel gekeken naar de directe burens van het ingevoerde datapunt, terwijl bij $K = 2$ ook de burens van de directe burens van het datapunt worden meegenomen. Zo kan K steeds verhoogd worden, totdat er een beter resultaat uit het model komt.



FIGUUR 2.16: Voorbeeld KNN

2.5 Relevante literatuur

Dit hoofdstuk bespreekt literatuur over het stellen van een diagnose van componenten in circuits. Deze informatie is relevant, omdat het als springplank gebruikt kan worden om een techniek te ontwikkelen die toe te passen is op de spoorstroomloop. Uit de literatuur blijkt dat er vier hoofdstromen zijn die ingezet kunnen worden om diagnoses te stellen op circuits. Hieronder worden deze methodes beschreven en met elkaar vergeleken. Op die manier ontstaat een kader voor een methode die in dit project ontwikkeld kan worden.

2.5.1 Huidige diagnose strategie van ProRail

De huidige methode die ProRail toepast op het spoor staat beschreven in [19]. Deze methode berust voornamelijk op de kwaliteiten van de monteurs. Deze gaan aan de hand van een visuele inspectie na of er problemen zijn met het spoor. Dit is een vrij logische stap, want als een kabel niet goed is aangesloten, dan heeft het geen zin om uitgebreid te sleutelen aan de componenten.

Echter, als een monteur merkt dat sprake is van een probleem dat wordt veroorzaakt door het relaïskast, dan zal hij om te beginnen aan de condensator schroeven om na te gaan of het relaï daarop reageert. Wanneer hij merkt dat dit geen effect heeft, zal hij een nieuw relaï plaatsen om te kijken of dat wel effect heeft. Als blijkt dat het relaï dan wel opkomt, kan ervan uit worden gegaan dat de condensator defect was. Vervolgens gaat de monteur de componenten in de spoorstroomloop af, om te controleren of deze nog naar behoren werken. Zo kan bij een railspoel worden gemeten of de overdracht nog correct is. De diagnosetechniek van de monteurs berust kortom op ervaring en trial and error.

Kansberekening

In tabel 2.1 wordt het aantal keer dat een bepaald defect voorkomt aan spoorstroomloop weergegeven. Wanneer er vanuit wordt gegaan dat defecten gelijk verdeeld zijn over de twee resonantiekringen van de spoorstroomloop, en wordt uitgegaan van de foutmodi die worden onderzocht in deze scriptie. Namelijk defecten aan de condensator, spoorspoel en een verkeerd geconfigureerde instelling. Kan worden bepaald hoe vaak monteurs in een keer een defect aan een component kunnen identificeren. Monteurs zullen bij een defect aan het elektrische circuit vrijwel altijd eerst de instelling van het systeem proberen te veranderen. Wanneer dus die kans ten opzichte van de kans dat het probleem aan een ander component zit vergelijkt. Is de kans dat de monteur in een keer het probleem herkent 57.6 procent.

$$P_{herken} = \frac{D_{In}}{D_{In} + D_{Co} + D_{Rs}} \quad (2.13)$$

waar:

- P_{herken} = kans dat een foutmodi in een keer wordt herkend
- D_{In} = aantal defecten aan de instelling (53)
- D_{Co} = aantal defecten aan de condensator (23)
- D_{Rs} = aantal defecten aan de railspoel (16)

2.5.2 Een op conditie gebaseerde diagnose

De techniek gepresenteerd in bron [20] komt het meest overeen met de strategie die monteurs toepassen. De monteurs herkennen een bepaalde indicator en linken deze aan een fout. Bijvoorbeeld wanneer een SSL foutief aangeeft dat het spoor bezet is en de monteur het systeem controleert, komt hij erachter dat er een muntstuk op de scheiding tussen de baanvakken is gelegd. De monteur weet dan dat het probleem is ontstaan door het muntje. Een diagnose stellen wordt echter complexer wanneer het gaat over componenten in het systeem. Als bijvoorbeeld de waarde van een condensator veranderd is door degradatie en het systeem niet optimaal werkt, moet de monteur de relaiskast in en de componenten onderzoeken. Deze techniek werkt dus om vast te stellen of een systeem defect is, maar niet zo goed om te kunnen bepalen of een component defect is.

2.5.3 Een op foutmodi gebaseerd model

Het model in bron [1] is gebaseerd op het vooraf vaststellen van verschillende fout-hypotheses, dus wanneer een systeem niet meer functioneert. Dit gebeurt meestal doordat een foutanalyse wordt uitgevoerd op het circuit en door te kijken naar historische foutinformatie. Een bekende vorm van een dergelijk model is de *fault dictionary*-techniek. Een dergelijke foutbibliotheek bevat metingen van gesimuleerde situaties waarin steeds een fout wordt toegevoegd. Wanneer bij een diagnose van een systeem metingen worden gedaan, worden deze metingen vergeleken met de resultaten van de simulaties. Daaruit kan worden afgeleid bij welke gesimuleerde combinatie van fouten bij componenten, de gemeten waardes het beste aansluiten. Dit vertoont gelijkenissen met een classificatieprobleem en kan dus gecombineerd worden met een ML-algoritme.

2.5.4 Gedrag gebaseerd model

Een *behavioral*-model zoals beschreven in [21] en [22] werkt op de tegenovergestelde manier als een *fault*-model. De techniek is gefundeerd op een schatting van het gedrag van een circuit. Nadat vervolgens metingen zijn verricht van het systeem waarvoor een diagnose gesteld moet worden, wordt geprobeerd om via simulatie eenzelfde reactie te verkrijgen als de gemeten reactie van het circuit. Nadat een match is gevonden, kan worden geïdentificeerd welke componenten voor problemen zorgen. Een nadeel van deze techniek is dat het qua computerkracht een zware operatie is; er moet immers worden gezocht naar een combinatie van waardes.

2.5.5 Teststimulus, diagnostische metingen, extractie en selectie

In de manier van diagnose stellen zoals gepresenteerd in [13], wordt een bepaald inputsignaal gegeven aan het circuit (bijvoorbeeld ruis, zaagtand of ramp) en wordt de reactie van het systeem gemeten. Het signaal wordt vervolgens getransformeerd tot een paar parameters. Om dit model te laten werken, moeten goede parameters worden uitgezocht. Er zijn verschillende manieren om aan deze parameters te komen. Voorbeelden zijn het ARMA-model, zoals beschreven in [23], of een rampsignaal, zoals gebruikt in [13]. Verder zijn er verschillende manieren om de data te bewerken. Zo wordt soms gekeken naar intervallen in de tijd, maar worden in andere gevallen de coëfficiënten van een wavelet-transformatie gebruikt, in combinatie met andere factoren die uit het signaal naar voren komen.

2.5.6 Keuze methode

Hierboven zijn vier methodes beschreven die als kader kunnen dienen voor de te ontwikkelen methode in dit project. Echter, in dit project wordt ervoor gekozen om verder te werken met de methode die gebaseerd is op een testsignaal en waarbij de meting in een ML-model wordt geladen, zoals beschreven in [13]. Het voordeel van deze methode is dat het voor controle zorgt in de testopstelling, omdat wordt bepaald wat het ingangssignaal is. Zo is er geen afhankelijkheid van het ingangssignaal van de spoorstroomloop en kan beter worden gegarandeerd dat deze methode werkt bij de wisselende spooromstandigheden.

Het op gedrag gebaseerde model is niet geschikt als methode voor de SSL, omdat verwacht wordt dat vanwege de aard van het spoor zoals beschreven in paragraaf 2.1, omdat verwacht wordt dat deze methode teveel computer kracht kost om snel en efficiënt te werken. Een op conditie gebaseerd model zou kunnen werken wanneer alle variabelen op het spoor gemeten worden, zoals de weeromstandigheden. Het is mogelijk om foutmodi voor deze spoorstroomlopen te testen en op te stellen, maar dit kost veel tijd.

Onderdelen van het op foutmodi gebaseerde model worden in dit project gebruikt, omdat de methode in [13] ook uitgaat van het van tevoren opstellen van fouthypothesen. In dit project wordt dus verdergegaan met een methode die een testsignaal het circuit instuurt. Vervolgens wordt daarvan de reactie gemeten. Deze reactie wordt dan naast de al van tevoren gesimuleerde reacties van de foutmodi gehouden.

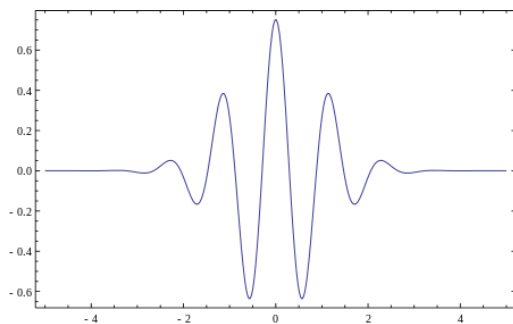
2.6 Wavelets

In bron [13] waarop de gekozen methode uit paragraaf 2.5.6 is gebaseerd staat beschreven dat een signaal getransformeerd wordt naar karakteristieke features aan de hand van wavelet-transformaties. Om meer duidelijk te geven op deze vorm van signaal analyse, zal in deze paragraaf worden ingegaan op de toepasbare kennis van deze transformatie voor dit project.

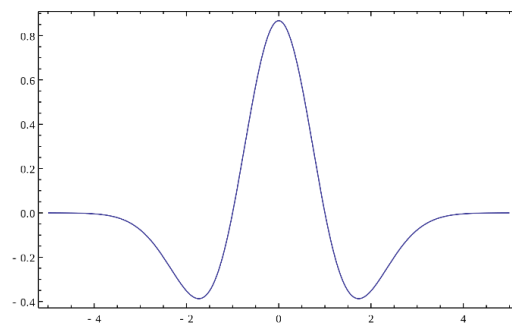
Wavelets zijn functies die beginnen op nul, vervolgens in amplitude toenemen en daarna weer afnemen tot nul. Deze functies worden toegepast om de verschillende niveaus van een signaal te benaderen.

Wavelet-transformaties worden bijvoorbeeld ingezet om foto's te bewerken. In een aantal opzichten lijkt de wavelet-transformatie op de fouriertransformatie. Beide zijn immers technieken om een signaal te ontleden in zijn frequentiecomponenten. Het verschil tussen beide transformaties is dat bij een fouriertransformatie de tijdsinformatie verloren gaat. Dus wanneer een verandering in het signaal plaatsvindt, kan wel worden waargenomen dat een verandering heeft plaatsgevonden, omdat dan veel hoge frequenties waargenomen kunnen worden, maar kan niet worden aangegeven wanneer die verandering heeft plaatsgevonden. De wavelet-transformatie biedt hier uitkomst, omdat deze het wel mogelijk maakt om te achterhalen wanneer sprake was van een bepaalde verandering.

De wavelet-transformatie vindt plaats door een bepaalde wavelet van een vooraf gedefinieerde golf te schalen en te verplaatsen, totdat deze het signaal representeert. Wavelets worden gedefinieerd als $\psi(t)$ en komen in verschillende vormen voor. Hieronder zijn twee voorbeelden gegeven.



FIGUUR 2.17: Morlet wavelet [24]



FIGUUR 2.18: Mexican hat wavelet [25]

De definitie van een wavelet waarbij de continue schaling en verschuiving is gegeven, staat in 2.14. Daarbij zijn (a, b) de schalings- en verschuivingsparameters.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (2.14)$$

waar:

ψ = wavelet functie

a = schalingsparameter

b = verschuivingsparameter

t = tijd in seconden

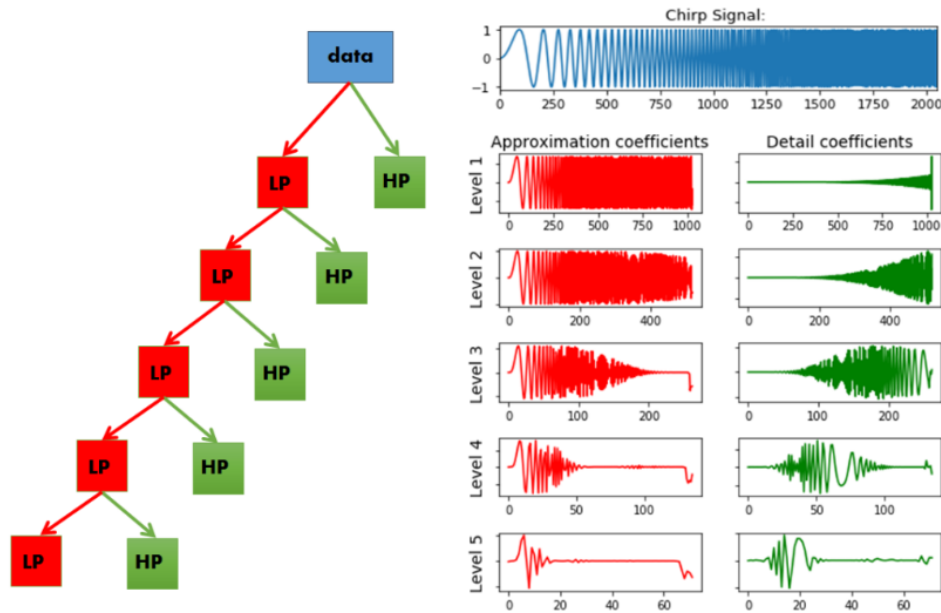
Voor een beter begrip is het nodig om te begrijpen wat de schalings- en verschuivingsparameters doen met het signaal. De schalingsparameter geeft informatie over of de breedte van de wavelet wordt vergroot of verkleind om de wavelet te laten passen in het signaal. Deze schalingsparameter is gelinkt aan de frequentie van de wavelet, omdat de breedte van een cyclus in een signaal de frequentie bepaalt. Deze breedte zorgt ervoor dat het signaal gefilterd wordt, omdat een specifieke frequentie zo goed mogelijk benaderd wordt. In essentie is een bandpass filter gecreëerd. De verschuivingsparameter geeft informatie over verandering en over waar deze plaatsvindt in de tijd.

Op basis van deze definitie is het mogelijk om de continue wavelet-transformatie te definiëren als vergelijking 2.15. Daarbij representeert $f(t)$ het signaal.

$$w_f(a, b) = \left\langle f(t), \psi_{(a,b)}(t) \right\rangle = \int f(t) \psi_{(a,b)}(t) dt \quad (2.15)$$

2.6.1 Wavelet als filterbank

In deze scriptie zal de discrete wavelet transformatie gebruikt worden om de karakteristieke parameters uit een signaal te verkrijgen. Daarbij zal het signaal in verschillende decompositie niveaus worden opgebroken. Dit betekent dat er eerst van de hoge frequentie een stuk wordt geanalyseerd en zo steeds een bandbreedte naar beneden wordt gewerkt om een beeld van alle frequenties in het signaal te krijgen.



FIGUUR 2.19: Discrete wavelet transformatie [26]

In figuur 2.19 wordt weergegeven hoe een signaal wordt opgebroken per decompositie niveau. De analyse begint met een lage schaal om de hoge frequenties te analyseren. Per laag wordt de schaal verdubbeld om zo steeds een stuk op te schuiven in de frequentie bandbreedte en steeds lagere frequenties te analyseren. Deze analyse levert uiteindelijk de schalings- en verschuivingsparameters oftewel wavelet-coëfficiënten per decompositieniveau op, gedefinieerd als $w_{j,k}$. De wavelet-coëfficiënten zijn laten de verschillende frequentie eigenschappen zien van de reactie van het systeem.

Hoofdstuk 3

Methode

In dit hoofdstuk wordt de voorgestelde methode uitgewerkt. De methode berust voornamelijk op de in bron [13] besproken methode. Eerst zal het basisprincipe worden toegelicht, en daarna zal worden ingegaan op de meetopstelling

3.1 Basisprincipe

De basis van het onderzoek is opgebouwd uit drie onderdelen, namelijk:

1. verschillende onderdelen van het circuit activeren;
2. de resultaten omzetten naar parameters die uniek zijn voor de verschillende foutmodi;
3. een model toepassen dat aan de hand van de verschillende parameters bepaalt in welke foutmodus het systeem zich bevindt.

Specifiek wordt dus door een signaal toe te passen op een bepaalde praktische aansluiting in het circuit, een response gemeten die het model wat kan vertellen over de verschillende onderdelen in het circuit.

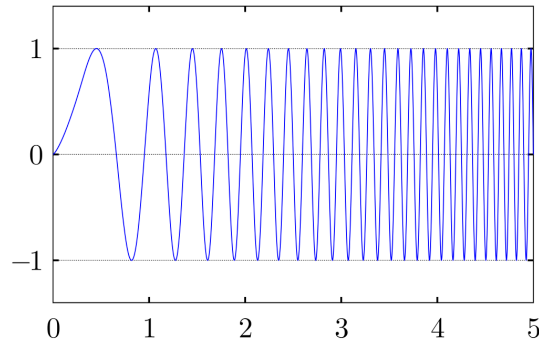
3.2 Meetopstelling

Bij de meetopstelling moet rekening worden gehouden met een aantal praktische eisen. Op die manier is de businesscase van dit project te versterken.

- Voor de monteur moet het praktisch zijn om de testapparatuur aan te sluiten. Het moet bijvoorbeeld niet zo zijn dat de monteur van de ene relaiskast naar de andere relaiskast moet lopen.
- De effecten van de meetopstelling mogen niet resulteren in een onveilige situatie op het spoor.

3.2.1 Testsignaal

Naast deze praktische eisen is ervoor gekozen om als testsignaal een sinus te gebruiken die lineair van frequentie verandert, een zogenaamd chirp-signaal. Voor dit testsignaal wordt gekozen, omdat wordt aangenomen dat dit signaal een interessante reactie kan opwekken bij de verschillende componenten in het circuit. Een andere optie is om gebruik te maken van het 75Hz-signaal dat wordt gebruikt om het systeem aan te sturen, maar dan is het niet mogelijk om een volledig beeld te krijgen van de transiënte eigenschappen van het systeem.



FIGUUR 3.1: Voorbeeld van een chirp signaal [27]

Daarom wordt ervoor gekozen om, zoals beschreven in bron [13], gebruik te maken van een chirp-siginaal dat lineair oploopt van 50Hz naar 150Hz. Deze range is gekozen, omdat dan nog steeds resonantie waargenomen kan worden over de verschillende gedefinieerde foutmodi.

3.2.2 Definitie foutmodi

De methode zoals gepresenteerd in paragraaf 2.3.1, kan worden gebruikt om voor verschillende componentwaarden de koppelmaat te berekenen. Op basis daarvan kan uiteindelijk worden bepaald bij welke waarde het systeem niet werkt. In tabel 3.1 is te zien wat de koppelwaarde is als een van de componenten varieert. Voor de rest zijn de volgende waarden gebruikt:

$$L_1 = L_2 = 2.25H$$

$$L_{ESR} = 0.1\Omega$$

$$C_1 = C_2 = 2\mu F$$

$$R_b = 5 \cdot 80^2 (1000m \text{ Sectielengte})$$

$$E_{in} = 130V \text{ 75Hz Sinus}$$

In tabel 3.2 staan de componentwaarden die corresponderen met de nummers in tabel 3.1.

TABEL 3.1: Koppelwaardes bij variatie component waardes

ID	Koppelwaarde			
	C1 en C2	C1	C2	L
1	174.0876	351.0942	351.9961	230.1521
2	260.8838	465.3876	466.3227	281.8222
3	434.1665	653.2731	654.1874	351.7176
4	924.4601	1013.6394	1014.3821	451.3913
5	3273.5806	1900.0281	1900.0281	604.4624
6	1156.3167	3427.0834	3424.3126	866.8382
7	655.2666	1917.8008	1914.5305	1399.2240
8	479.4857	1160.2858	1157.1467	2654.4456
9	391.6720	837.3087	834.1034	2866.7702
10	339.6920	664.0103	660.6256	1641.2580
11	305.7351	556.7613	553.1182	1097.9125
12	282.0836	484.0527	480.0807	830.6771
13	264.8493	431.5801	427.2050	675.2958

TABEL 3.2: Component waardes bij ID

Comp.\ID	1	2	3	4	5	6	7	8	9	10	11	12	13
C (in nF)	1.6	1.7	1.8	1.9	2	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8
L (in H)	1.55	1.65	1.75	1.85	1.95	2.05	2.15	2.25	2.35	2.45	2.55	2.65	2.75

Met de gevonden waardes kunnen de verschillende foutmodi worden bepaald voor dit systeem. Door COVID-19 kunnen geen fysieke testen worden uitgevoerd. Daarom wordt een aanname gedaan van bij welke koppelwaarde het systeem nog werkt. Daarom wordt de literatuur gevolgd en wordt gekeken bij welke component waardes in bron [6] het systeem in storing treedt. De meeste overeenkomst met bron [6] wordt waargenomen bij een koppelwaarde van 1000.

TABEL 3.3: Foutmodi condensator C1 en C2

	1.6	1.7	1.8	1.9	2	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8
C1 en C2	F	F	F	F	G	G	F	F	F	F	F	F	F
C1	F	F	F	G	G	G	G	F	F	F	F	F	F
C2	F	F	F	G	G	G	G	F	F	F	F	F	F

In tabel 3.3 en 3.4 is met waardes weergegeven wanneer het systeem in storing treedt en wanneer niet. De "F" representeert de storingsgevallen en de "G" representeert de gevallen dat het systeem naar behoren werkt.

Om meer inzicht te krijgen in de capabiliteit van het ML-model, is ervoor gekozen om een klein vooronderzoek te verrichten. Dit vooronderzoek is uitgewerkt in bijlage A. In het onderzoek is getest of het mogelijk is om de methode die is bestudeerd in de literatuur, toe te passen op een resonantiecircuit. Dit is gedaan, omdat in essentie de twee belangrijkste componenten van de SSL, de resonantiekringen zijn.

Daarnaast is onderzocht of de weg die is ingeslagen met de wavelet-transformatie, de juiste is. De wavelet-transformatie is naast een andere signaalanalysetechniek gelegd, namelijk de discrete-fourieranalyse. Deze is uitgewerkt in bijlage B.

TABEL 3.4: Foutmodi Spoel

	1.75	1.85	1.95	2.05	2.15	2.25	2.35	2.45	2.55	2.65	2.75
L	F	F	F	F	G	G	G	G	G	F	F

Uit A is de eerste indicatie gekomen dat de wavelet-transformatie het tot ongeveer 79 procent nauwkeurig mogelijk maakt om de verschillende foutmodi te identificeren. Door de resultaten van A en B met elkaar te vergelijken, blijkt dat in dit geval de wavelet-transformatie beter werkt dan de discrete-fouriertransformatie. Met deze kennis is met enige zekerheid te stellen dat de technieken uit de literatuur toegepast kunnen worden op de SSL.

3.3 Voorgestelde methodes

Paragraaf 3.1 beschrijft waar de methode aan moet voldoen. Deze paragraaf bespreekt drie voorstellen om de beschreven doelen te behalen.

3.3.1 Zonder toegevoegd signaal

Als een basismeting wordt een methode ontwikkeld waarbij geen toevoegingen gebruikt hoeven te worden.

Methode 1: circuit zonder toevoeging meten

In de eerste methode wordt gewerkt met het bestaande ingangssignaal van 75Hz 110V. Daarbij wordt de spanning over de condensator aan de relaiszijde gemeten. Bij deze methode wordt gekeken naar de faseverschuiving tussen de spoorfase en de lokale fase en naar de amplitude. Deze methode maakt in essentie de overdracht van het systeem meetbaar.

3.3.2 Met toegevoegde signaal

Methode 2: vervanging ingangssignaal

Bij deze methode wordt het ingangssignaal van het systeem vervangen door het testsignaal. Het idee achter deze methode is dat het mogelijk wordt om de volledige overdracht van het systeem waar te nemen. Daarom wordt de spanning over de condensator in de tweede resonantiekring gemeten. Het gebruikelijke ingangssignaal van 110v wordt wel losgekoppeld, want deze wordt vervangen.

Methode 3a: enkelvoudige meting zonder ingangssignaal

Bij deze methode wordt op één plek in het circuit gemeten en wordt geprobeerd om op die manier alle foutmodi te herkennen. Daarvoor wordt het bestaande ingangssignaal verwijderd en wordt in plaats daarvan een testsignaal aangebracht in het circuit. Het doel van deze methode is om enkelvoudig in te grijpen in het SSL-circuit.

Methode 3b: Getrapte meting aan beide zijde zonder ingangssignaal

Bij deze methode wordt een getrapte meting uitgevoerd aan de aanstuur- en relaiszijde van de SSL. Daarvoor wordt het bestaande ingangssignaal van 75Hz 130V losgekoppeld en wordt getrapd een chirp-signaal ingestuurd aan beide zijde van de SSL. Daarvan wordt het gedrag gemeten. Als niet direct resultaat zichtbaar is, dan wordt hetzelfde gedaan aan de relaisziden. Deze methode bestudeert dus de overdrachtsreactie aan beide zijden. De verwachting is dat op deze manier het beste de verschillende foutmodi op te merken zijn.

Belangrijke kanttekening bij deze methode is het feit dat deze methode tegenstrijdig is met de uitgangspunten. Omdat de monteur bij deze methode als nog langs beide relaiskasten moet.

3.4 Signaalanalyse

Voor de metingen met een toegevoegd signaal wordt gebruikgemaakt van een aantal signaalanalysetechnieken. Het hoofddoel van de analyse is om de tijddomeingegevens van een signaal om te zetten naar de karakteriserende frequentie-eigenschappen van het signaal, zodat uiteindelijk de ML-modellen beter functioneren. Het belangrijkste instrument om het signaal te analyseren, is de wavelet-transformatie. Deze is besproken in paragraaf 2.6.

Voor de metingen wordt dezelfde strategie gebruikt als in bron [13]. Zoals uitgelegd in paragraaf 2.6, is het voordeel van een wavelet transformatie dat de belangrijke signaal eigenschappen naar boven komen. Op die manier kan uitspraak worden gedaan over veranderingen in het signaal. Zoals eerder uitgelegd, moet de grondwavelet worden geschaald en verplaatst totdat deze overeenkomt met het signaal. Deze relatie tussen de grondwavelet is weergegeven in vergelijking 2.14 in paragraaf 2.6.

Eerst wordt een discrete wavelet-transformatie uitgevoerd op het signaal. Daaruit worden de wavelet-coëfficiënt $w_{j,k}$ verkregen, er is voor gekozen om van de eerst vier decompositielagen de coëfficiënt te gebruiken. Om ruis van het signaal te onderscheiden worden de volgende 4 vergelijkingen gebruikt. De eerst stap daarin is om de ruisvariantie te bepalen, volgens de relatie gegeven in vergelijking 3.1

$$\hat{\sigma}_j = \frac{\text{med}(|w_{j,k}|)}{0.6745} \quad (3.1)$$

waar:

$w_{j,k}$ = wavelet-coëfficiënten per decompositie niveau
 $\hat{\sigma}_j$ = ruisvariantie
 $\text{med}(\ast)$ = mediaan

Daarna wordt een grenswaarde voor de ruis bepaald per decompositieniveau van de wavelet transformatie.

$$T_j = \hat{\sigma}_j \sqrt{2 \cdot \ln(2^j)} \quad (3.2)$$

T_j = grenswaarde per decompositieniveau

Vervolgens wordt met deze grenswaarde bepaald welke componenten ruis zijn en welke karakteristiek zijn voor het signaal.

$$d_{j,k} = \begin{cases} \text{sgn}(w_{j,k})(|w_{j,k}| - T_j) & \text{als } |w_{j,k}| > T_j \\ 0 & \text{Anders} \end{cases} \quad (3.3)$$

waar:

$\text{sgn}(\ast)$ = signum functie

$d_{j,k}$ = gefilterde wavelet-coëfficiënten per decompositie niveau

Nu de ruis van het signaal verwijderd is, kan per decompositieniveau de energie worden uitgerekend. Deze waarde kan worden gebruikt om uitspraak te doen over het gedrag van het circuit op verschillende frequenties. Deze waarde wordt als volgt berekend:

$$E_j = \sum_k |d_{j,k}|^2 \quad (3.4)$$

Parameter E_j wordt als input gebruikt voor de verschillende ML-modellen.

Verder worden, zoals beschreven in bron [13], twee statistische waardes uit de data gehaald. De kurtosis geeft informatie over het aantal extremen in het signaal. Zo kan worden waargenomen dat er bij een hoge kurtosis weinig extremen zijn en dat het signaal dus een lage variatie heeft. De kurtosis wordt bepaald met de relatie weergegeven in vergelijking 3.5.

$$\text{kurt}(x) = \frac{E[x - E[x]]^4}{[E[x - E[x]]^2]^2} \quad (3.5)$$

Uiteindelijk blijven vijf features over die gebruikt worden als input voor de ML-algoritmes. Voordat deze in het model worden geladen, worden de gevonden features geschaald, zodat ze niet meer afhankelijk zijn van de schaal van de meting.

3.4.1 Overzicht

In tabel 3.5 staat een overzicht van de parameters (*features*) die worden gebruikt om de ML-modellen te trainen en te gebruiken. De energie is de onderscheidende parameter als het gaat om de reactie van het systeem op de verschillende aangeboden frequenties. De kurtosis geeft informatie over de hoeveelheid extremen in het signaal en de entropie geeft de mate van informatiedichtheid aan van de te verwachten uitkomsten.

TABEL 3.5: Overzicht van signaalanalyse parameters voor ML-modellen

Naam	Symbool
WT energie op decomposities niveaus	E_j
Kurtosis	$\text{Kurt}(x)$

3.5 Analyse van de resultaten

Deze paragraaf bespreekt de manier waarop de resultaten die uit de ML-modellen naar komen, geanalyseerd dienen te worden. Om het verslag compact en leesbaar te houden, is ervoor gekozen om de verschillende methoden uit te werken in bijlage C, D en E. Alleen de best werkende methode wordt uitgewerkt in het rapport. Daardoor vindt de analyse dus plaats in twee stappen.

3.5.1 Bepalen beste methode en ML-model

Om de beste methode te bepalen, wordt de nauwkeurigheid van de modellen en de methode bestudeerd. Daarbij worden alleen de twee best presterende modellen van elke methode meegenomen. Die twee methoden worden vervolgens tegen elkaar afgewogen, om na te gaan hoe goed ze presteren en om fouten in de juiste resonantiekring te vinden.

Nauwkeurigheid bepalen

De nauwkeurigheid bepalen, gebeurt door na te gaan hoe vaak het model de juiste categorie identificeert. De manier waarop dit wordt bepaald, is weergegeven in vergelijking 3.6.

$$h = \frac{\sum_{k=1}^n T_k}{\sum_{k=1}^n T_k + \sum_{i=1}^m F_i} \quad (3.6)$$

waar:

T = de correct geïdentificeerde waardes per categorie;
 F = de fout geïdentificeerde waardes per categorie.

Om verder onderscheid te kunnen maken tussen de methodes en de ML-modellen, wordt gekeken of het algoritme in staat is om te bepalen of het model kan bepalen waar de fout op een hoger niveau zich bevindt. Omdat de SSL is opgebouwd uit twee resonantiekringen, wordt bestudeerd wat het vermogen is van het model om te bepalen of wel of niet sprake is van een fout in de resonantiekring.

$$j = \frac{\sum_{k=1}^{n_c} (R_{at})}{\sum_{i=1}^{n_f} (F_i) + \sum_{k=1}^{n_c} (Rc_k + Rd_k)} \quad (3.7)$$

waar:

R_{at} = de set van alle resultaten die correct geïdentificeerd zijn in de resonantie kring
 F = de set van alle resultaten van fout geïdentificeerde data buiten de resonantie kring

Om duidelijk te maken wat met vergelijking 3.7 wordt bedoeld, staat hieronder een tabel waarin gelabeld is wat wel en niet moet worden meegenomen in de vergelijking. In tabel 3.6 zijn de volgende afkortingen gebruikt:

- F.I. = foute indicatie
- G.I. = goede indicatie
- G.B. = geen besluit
- AT = resonantie kring 1
- CT = resonantie kring 2

Een foute indicatie (F.I.) geeft alle gevallen weer waarbij het model de monteur op het verkeerde spoor heeft gezet. Een goede indicatie (G.I.) geeft aan dat de monteur wel bij de juiste resonantiekering aan het zoeken was. Deze vergelijking moet met een maat aangeven hoe vaak de monteur naar een fout moet zoeken die niet bestaat, of hoe vaak hij, zonder dat sprake is van een fout, naar een andere relais kast is gestuurd.

Naast deze twee categorieën is er de categorie geen besluit (G.B.). Als de monteur het bericht krijgt dat er geen probleem is met de SSL, maar na een visuele inspectie kan aantonen dat wél sprake is van een fout, kan worden beredeneerd dat de methode geen uitsluitsel kan geven.

TABEL 3.6: Overzicht correcte indicaties

werkelijk\voorspelling	A	B	C	D	E
A	* G.I.	F.I.	F.I.	F.I.	F.I.
B	G.B.	CT G.I.	F.I.	F.I.	CT G.I.
C	G.B.	F.I.	AT G.I.	AT G.I.	F.I.
D	G.B.	F.I.	AT G.I.	AT G.I.	F.I.
E	G.B.	CT G.I.	F.I.	F.I.	CT G.I.

Uit het overzicht zijn twee parameters te verkrijgen, namelijk het percentage dat het algoritme een monteur op het verkeerde zet en het percentage waaruit blijkt hoe goed hij presteert per resonantiekering.

3.5.2 Componentanalyse van de resultaten

De bovenstaande analyse kan worden gebruikt om te bepalen welk model en welke methode het beste werken. De meer diepgaande analyse gaat in op de prestaties van de beste methode en het ML-model. Zo moet bekeken worden bij welke componentwaardes het algoritme het beste en het slechts presteerde. Dit wordt gedaan door te vergelijken bij welke waardes een foute voorspelling is gedaan. Op basis van die data wordt vervolgens gezocht naar patronen. Zo kan bijvoorbeeld opgemerkt worden dat een bepaald bereik altijd foute waardes geeft. Uit deze patronen kan afgeleid worden hoe betrouwbaar de methode is bij verschillende fouten. In de eerste testen wordt enkel een component gedraaid om een foutmodus te simuleren, maar om een beter beeld te krijgen van hoe goed de methode en het ML-model werken, is het ook van belang om de onvoorspelbaarheid van het spoor mee te nemen. Dit wordt gedaan door een Monte-Carlo analyse uit te voeren. Daarbij worden tien keer binnen een bepaald bereik random de componenten van het spoor gewijzigd.

Hoofdstuk 4

Resultaten

Zoals beschreven in het vorige hoofdstuk, worden drie methodes getest om te achterhalen wat het beste werkt om de foutmodi van de SSL vast te stellen. In dit hoofdstuk worden eerst de verschillende methodes uitgewerkt. Daarna wordt ingegaan op de best werkende methode. Deze methode wordt gekozen op basis van de resultaten van de verschillende experimenten. Specifiek dient onderzocht te worden in hoeverre deze methode werkt, zodat uiteindelijk de toegevoegde waarde van het systeem voor ProRail bepaald kan worden.

4.1 Uitwerking Methodes

Om het rapport compact te houden, is ervoor gekozen de methodes uit te werken in de bijlagen. Methode 1 is uitgewerkt in bijlage C, methode 2 is uitgewerkt in bijlage D en methode 3a en 3b zijn uitgewerkt in bijlage D.

TABEL 4.1: Overzicht van voorgestelde meet methodes

<i>Methode</i>	<i>Korte beschrijving</i>
1	Geen toegevoegde testsignaal
2	Vervanging van het inganssignaal door testsignaal
3a	Enkelzijdige meting van buiten de relaïskast
3b	Getrapte metingen van buiten de relaïskasten

In tabel 4.2 is een samenvatting te zien van de resultaten van de experimenten die zijn gedaan op basis van de verschillende methodes.

TABEL 4.2: Samenvatting resultaten experimenten

ML - model \Nauwkeurigheid	Methode 1	Methode 2	Methode 3a	Methode 3b
Neural Network	70%	55%	71%	80%
Support vector machine	60%	55%	74%	80%
Gradient boosted decision tree	69%	64%	74%	82%
K-nearest neighbors	72%	56%	76%	83%

Uit de resultaten blijkt dat methode 3b het meest nauwkeurig is, maar, zoals beschreven in de bijlage en in paragraaf 3.3, is bij deze methode sprake van een *trade-off*. Deze methode herkent namelijk niet alle foutmodi tegelijk. Daarom moet in twee stappen worden gewerkt om te bepalen waar de fout exact zit. Wel is het zo dat wanneer de monteur bij de resonantiekring staat waar de fout zich bevindt, hij niet verder hoeft te zoeken.

Methode 3a scoort ook relatief goed en heeft het voordeel dat het direct kan identificeren welke component waarschijnlijk de storing in het systeem geeft. Het voordeel daarvan is dat de monteur niet onnodig heen en weer hoeft te lopen tussen relaïskasten.

Tot slot is nagegaan hoe nauwkeurig de methode is om fouten te onderscheiden op het niveau van de resonantiekringen. Deze resultaten staan hieronder in tabel 4.3:

TABEL 4.3: Samenvatting resonantiekring fouten score

Model \ Score	Methode 1	Methode 2	Methode 3a	Methode 3b
Neural Network	81%	-	-	-
Support vector machine	-	-	-	-
Gradient boosted decision tree	-	56%	82%	70%
K-nearest neighbors	83%	86%	76%	75%

Over het algemeen werkt het KNN-model het beste. Alleen bij methode 3a werkt de gradient boosted decision tree het beste.

4.1.1 Keuze diepgaande analyse

Deze paragraaf bespreekt welke methode en welk algoritme gekozen worden om verder te analyseren. Dit wordt gedaan aan de hand van de bovenstaande resultaten.

Methode 3b scoort het beste, maar zoals eerder uitgelegd, is het bij deze methode niet mogelijk om alle foutmodi in één keer te herkennen. Dit roept de vraag op of methode 3a misschien sneller is in de praktijk. Het gaat er dus om met welke methode de monteur sneller het probleem kan vinden, daarbij rekening houdend met hoe vaak het algoritme het fout heeft. In dit onderzoek is dat vrij moeilijk te bepalen, omdat het niet mogelijk is om de twee methodes in het veld te testen en dan de resultaten te vergelijken. Daarom moet een aanname worden gedaan over wat het effect hiervan is. De volgende aannames worden gedaan over een reparatie aan de SSL:

- de kans dat een storing optreedt aan een van de twee zijdes is 50 procent;
- een meting kost vijf minuten;
- lopen naar het andere relaïshuis duurt tien minuten;
- als de methode niks oplevert, moet de monteur het systeem onderzoeken;
- na onderzoek heeft de monteur altijd het probleem gevonden;
- het onderzoek duurt vijftien minuten.

Met de bovenstaande gegevens kan per methode worden berekend hoelang het duurt voordat het probleem gevonden is. In tabel 4.4 en 4.5 staan de resultaten:

De verkeerde zijde is de kans dat de monteur het andere relaïshuis is gaan meten, omdat dit relaïshuis als resultaat uit de meting kwam, en dat de monteur daarna moest teruglopen omdat hij het probleem niet kon waarnemen bij dit relaïshuis.

TABEL 4.4: Tijd om probleem op te lossen bij methode 3a

<i>Methode A</i>	Kans	Tijd tot vinden probleem (min)
Goed zijde 1:	37.5%	5.00
Goed zijde 2:	37.5%	10.00
Goed na onderzoek zijde 1:	11.38%	20.00
Goed na onderzoek zijde 2:	11.38%	25.00
Verkeerde zijde 2	1.12%	45.00
Verkeerde zijde 1	1.12%	50.00

Gemiddelde tijd bezig		11.81
------------------------------	--	-------

TABEL 4.5: Tijd om probleem op te lossen bij methode 3b

<i>Methode A</i>	Kans	Tijd tot vinden probleem (min)
Goed zijde 1:	41.5%	5.00
Goed zijde 2:	41.5%	15.00
Goed na onderzoek zijde 1:	7.40%	20.00
Goed na onderzoek zijde 2:	7.40%	30.00
Verkeerde zijde 2	1.10%	45.00
Verkeerde zijde 1	1.10%	55.00

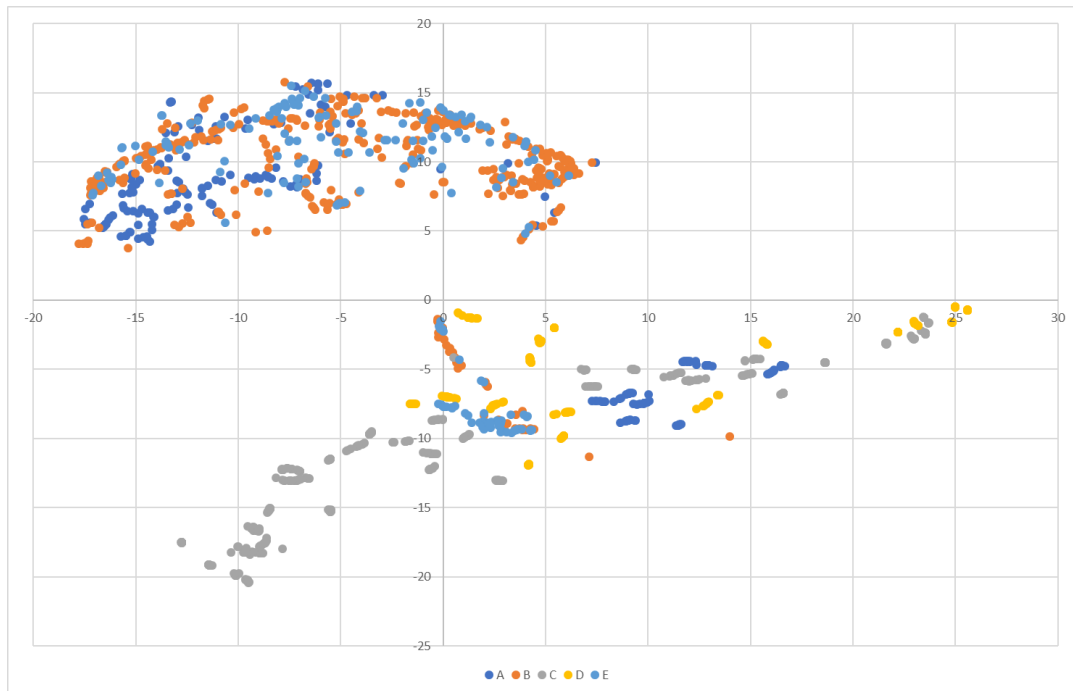
Gemiddelde tijd bezig		13.1
------------------------------	--	------

De tijden zijn verkregen door de kans te berekenen dat een bepaalde situatie zich voordoet en deze kans te vermenigvuldigen met een geschatte tijd. Via deze methode is gebleken dat methode A de snelste manier is om een diagnose te stellen van de SSL. Daarom wordt in de rest van dit onderzoek alleen methode 3a diepgaand geanalyseerd. Hierbij wordt gebruikgemaakt van de gradient boosted decision tree.

4.2 Diepgaande analyse gekozen methode

In de diepgaande analyse is methode 3a nogmaals onderworpen aan simulaties. Naast dat de verschillende foutmodi zijn gesimuleerd, zijn ook op elke foutmodus tien Monte-Carlo simulaties toegepast. Bij deze simulaties variëren de verschillende componenten van het spoor, zoals de ballastweerstand. Hierdoor zijn tevens de hoge mate van variabiliteit van het spoor en meer datapunten meegenomen, waardoor het model nog beter te trainen is.

Wanneer de data worden omgezet naar features, ontstaat de volgende t-SNE-visualisatie van de data.



FIGUUR 4.1: t-SNE representatie uitgebreide analyse

In het figuur is te zien dat twee resonantiekringen ver uit elkaar liggen. Dus wanneer een fout in de resonantiekring zich aan de aanstuurzijde bevindt (categorie C en D), dan ligt die onder de X-as. Andersom geldt dit voor fouten aan de relaiszijde (categorie B en E). De aanstuurzijde beter te scheiden dan de relaiszijde. Aan de relaiszijde is te zien dat de resultaten meer door elkaar heen liggen en dat ze lijken op het gedrag dat zichtbaar wordt wanneer geen sprake is van een fout in het circuit. Dit komt waarschijnlijk omdat de relaiszijde verder van het meetpunt ligt.

TABEL 4.6: Overzicht gedefinieerde foutmodi

<i>Categorie</i>	<i>Beschrijving</i>
A	Geen storing aan de SSL
B	Condensator relaiszijden defect
C	Condensator aansturingszijden defect
D	Spoorspoel aansturingszijden defect
E	Spoorspoel relaiszijden defect

4.2.1 Nauwkeurigheid

Wanneer wordt gekeken naar de nauwkeurigheid van de grote dataset, valt op dat deze een hogere score laat zien. Uit tabel 4.7 blijkt dat met deze methode een score kan worden gehaald van 86 procent. Zoals verwacht, heeft het model meer moeite met de categorieën B en E, omdat deze in de resonantiekring liggen die ver weg ligt ten opzichte van het meetpunt.

4.2.2 Componentanalyse fouten

Deze paragraaf behandelt de resultaten die fout geclassificeerd zijn. Zo is na te gaan of sprake is van een patroon in de resultaten die als fout worden geclassificeerd.

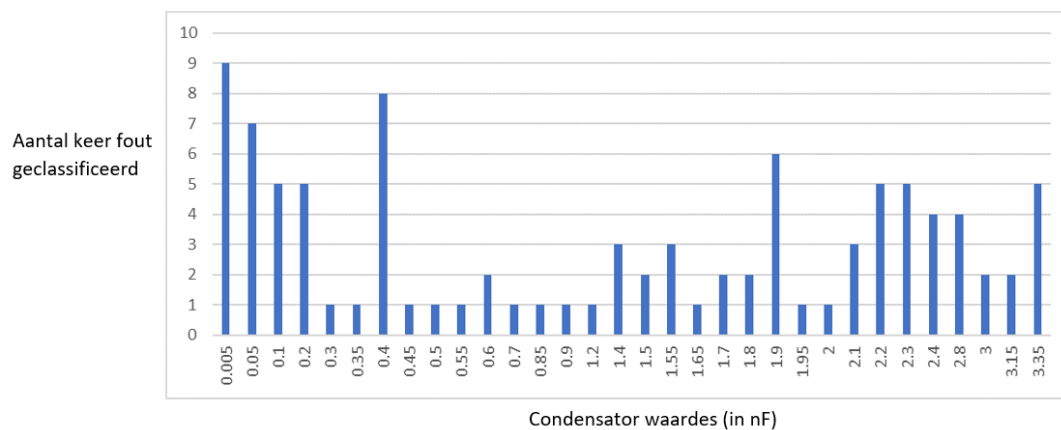
TABEL 4.7: Samenvattingstabel resultaten uitgebreide analyse

Werkelijk \ Voorspelling	A	B	C	D	E
A	40	1	0	1	1
B	4	68	0	1	10
C	2	0	87	1	0
D	0	0	0	28	0
E	1	18	0	0	23

Nauwkeurigheid	86.014%
Error Rate	13.986%
Aantal Goed	246
Aantal Fout	40

Daarbij worden ook de data meegenomen die zijn gebruikt om het ML-model te trainen.

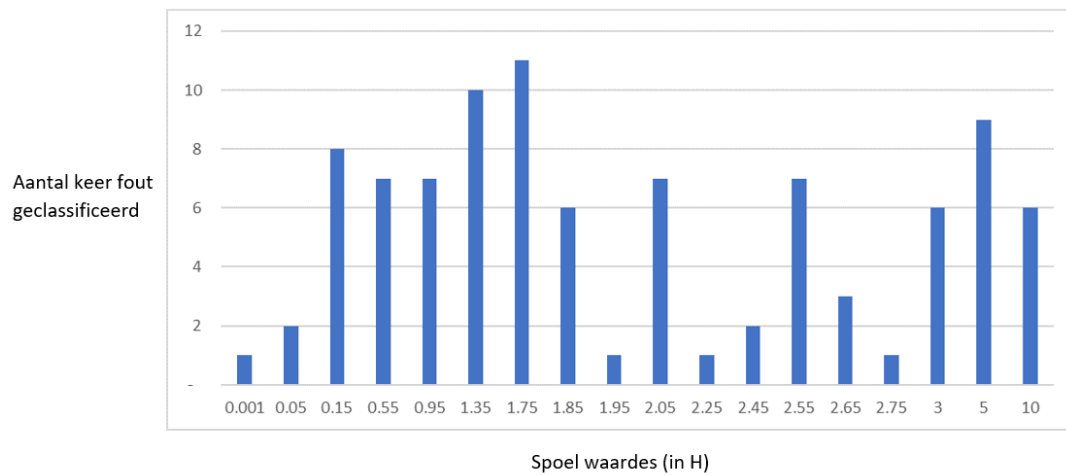
Hieronder staan in figuur 4.2 de fout geclassificeerde condensatorwaardes.



FIGUUR 4.2: Staafdiagram waardes condensator

Het blijkt dat vooral bij hoge en lage condensatorwaardes veel fouten zijn waar te nemen.

Hieronder staan in figuur 4.3 de fout geclassificeerde spoelwaardes.



FIGUUR 4.3: Staafdiagram waardes spoel

Uit figuur 4.3 blijkt juist dat de foutwaardes meer verspreid liggen en dat niet een duidelijk patroon te herkennen is in de fouten.

Uit alle resultaten blijkt dat dit algoritme goed fouten herkent dichtbij de resonantiekring waar gemeten wordt. De methode en het model hebben echter meer moeite om foutmodi te onderscheiden in verder weg gelegen resonantiekringen.

Hoofdstuk 5

Conclusie

Aan het begin van het onderzoek zijn vragen opgesteld. Deze hadden tot doel om het project te structureren. Daarnaast is aan de hand ervan duidelijk te maken in hoeverre het project is geslaagd. De onderzoeksvragen zijn:

1. Welke methoden noemt de literatuur in het onderwerp van het identificeren van verschillende fout modi in elektrische circuits?
2. Welke methode kan worden toegepast op de spoorstroomloop?
3. In hoeverre werkt deze methode en heeft het een meerwaarde om hiernaar een vervolgonderzoek uit te voeren?

Hoofdvraag: Is er methode waarmee een hogere kans om een defect component te identificeren aan de resonantiecircuiten in de spoorstroomloop, dan de huidige methode die is toegepast door ProRail?

5.1 Literatuur

In dit onderzoek is ervoor gekozen om literatuur te gebruiken die meer informatie geeft dan alleen over spoorssystemen en spoorstroomlopen. Op basis daarvan is een methode ontwikkeld en is een breder besef ontstaan van de verschillende mogelijkheden. Uiteindelijk is gekozen voor een methode waarbij het systeem onderworpen wordt aan teststimulus. Dit is gedaan, omdat in literatuur waarin machine-learning-algoritmes zijn gebruikt, goede resultaten naar voren kwamen. Verder heeft deze methode het mogelijk gemaakt om op specifieke frequenties te testen en om meer plaatselijke testen in het circuit uit te voeren. De andere onderzochte methodes leken minder geschikt voor de spoorstroomloop, vanwege het onvoorspelbare karakter van het spoor. Een overzicht van de overweging is te vinden in paragraaf 2.5.6.

5.2 Methode

De gekozen methode (zie paragraaf 3.3.2) waarbij een testsignaal wordt aangesloten aan een zijde van het circuit en de reactie van het circuit wordt gemeten, haalt een acceptabele betrouwbaarheid. In simulaties is de betrouwbaarheid 86 procent. Dit betekent dat wanneer een monteur deze methode toepast en een meting uitvoert aan de spoorstaven buiten het relaishuis, hij met een acceptabele betrouwbaarheid direct kan bepalen welke component in het systeem defect is. Daarbij is wel op te merken dat deze methode beter is in het identificeren van problemen in het relaishuis aan de zijde van de aansturing dan bij de zijde waar de detectie plaatsvindt.

5.3 Overwegingen

Bij de interpretatie van de resultaten is het van belang ervan bewust te zijn dat de resultaten gebaseerd zijn op simulaties. Daardoor zijn de gepresenteerde betrouwbaarheidsscores minder betrouwbaar. Het gedrag van het spoor is ingeschat. Daarbij is uitgegaan van een model dat beschreven is in de literatuur, maar door COVID-19 was het niet mogelijk om deze gegevens te verifiëren.

Daarnaast was het niet mogelijk om de gekozen methoden in het veld te testen. Het is dus niet duidelijk of de methode daadwerkelijk werkt wanneer deze wordt toegepast op de opstellingen die bij het spoor liggen.

5.4 Eindconclusie

Wanneer deze methode wordt vergeleken met de geschatte kans dat een monteur in een keer een defect component kan identificeren. Kan worden geconcludeerd dat deze methode in simulatie een hogere kans heeft om in een keer een defect component binnen de resonantiekringen van de spoorstroomloop te identificeren.

Hoofdstuk 6

Aanbevelingen

De uiteindelijke wens van ProRail is een hulpmiddel waarmee de monteur sneller defecte componenten aan de spoorstroomloop kan herkennen. Echter zijn op dit moment de resultaten van het onderzoek nog niet voldoende geverifieerd, om een betrouwbare uitspraak te kunnen doen over deze vraag.

Dus voor vervolgonderzoek wordt aangeraden om testen uit te voeren op de installaties in het Railcenter met de gepresenteerde methode in dit verslag. Als ook dan nog een hoge betrouwbaarheidsscore wordt gehaald, is verder vervolgonderzoek naar een hulpmiddel aan te bevelen. Daarnaast moet ook nog worden getest of deze methode goed is toe te passen op de andere varianten van de spoorstroomloop.

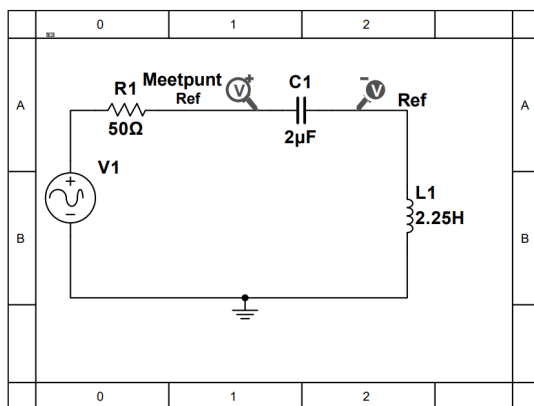
Bijlage A

Experiment: Machine Learning bij LC Resonantie Circuits met DWT

In dit experiment zal worden getest of er voor de gebruikte ML algoritmes¹ in dit project een goede score kan worden gehaald. Dus in essentie hoe goed de verschillende fout modi bij deze resonantie circuits te onderscheiden zijn. Daarbij wordt gebruik gemaakt van de Discrete Wavelet Transform om de features te creëren.

A.1 Opzet

Het circuit dat in dit experiment gebruikt zal worden is weergegeven in figuur A.1. In het circuit worden dezelfde waarden gebruikt als in het SSL circuit². De meting zal bestaan uit de spanning die staat over de condensator, er zal net zoals in de voorgestelde methode en zoals in [13] gebruik worden gemaakt van een *chirp* signaal als ingang spanning bron. De frequentie zal vanaf 1 naar 150Hz gaan in 0.2 seconden met een spanning van 10V.



FIGUUR A.1: Circuit onder test

A.1.1 Parameters

Het gemeten signaal zal worden omgezet in twee parameters die ook al eerder zijn toegelicht in paragraaf 3.4. Deze parameters zijn:

¹Het ML-Algoritme zal worden uitgevoerd via KNIME Analytics Platform for Windows 4.1.2 op een Microsoft Surface Pro 7 met Windows 10

²Het circuit zal worden gesimuleerd in NI Multisim versie 14.1 op een Microsoft Surface Pro 7 met Windows 10

Naam	Symbool
WT energie op decomposities niveaus	E_j
Kurtosis	$Kurt(x)$

A.1.2 Fout modi

Er zijn drie fout modi gedefinieerd die herkend moeten worden door de ML-modellen, deze zijn voor het grootste deel gedefinieerd in hoofdstuk 3.2.2. De componenten worden veranderd van de nominale waardes door ze naar boven (\uparrow) of naar beneden (\downarrow) bij te stellen. Alle veranderingen die getest worden zijn hieronder weergegeven.

Component	Verandering	Categorie
Alle	=	A
C1	\uparrow	B
C1	\downarrow	B
L1	\uparrow	C
L1	\downarrow	C

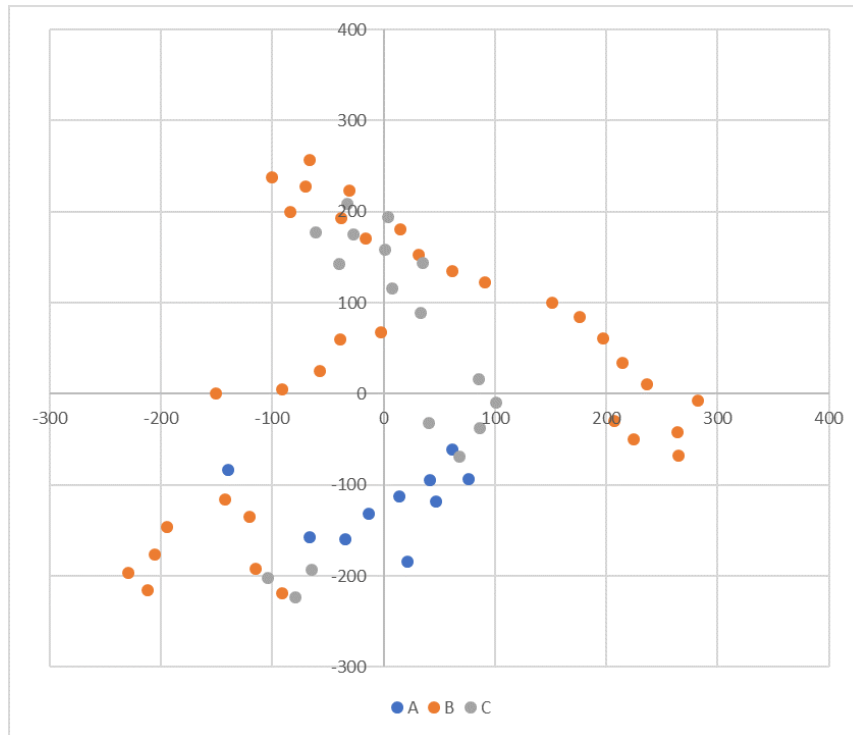
Naast de in de hoofdstuk 3.2.2 vermelden verschillende fout modi zijn nog extra aangevuld met meer mogelijkheden voor de spoel. Deze zijn hieronder weergegeven:

Waarde L1	Categorie
0.001	F
0.005	F
0.01	F
0.05	F
0.15	F
0.55	F
0.95	F
1.35	F
1.75	F
1.85	F
1.95	F

Waarde L1	Categorie
2.05	F
2.15	G
2.25	G
2.35	G
2.45	G
2.55	G
2.65	F
2.75	F
3	F
5	F
10	F

A.2 Resultaten

De in figuur A.2 weergegeven weergave representeert het omgezette signaal in de features zoals beschreven in deze bijlage. Deze figuur is een *t-distributed Stochastic Neighbor Embedding* weergaven van de features. Dit is om hogere dimensionale data om te zetten naar een visualisatie in een 2 of 3 dimensies.



FIGUUR A.2: t-SNE Representatie van Features

Wat al direct opvalt is dat het lijkt dat de data goed te scheiden valt zeker categorie A van de rest, daar zit maar een marginale overlap met categorie B en C. Echter tussen categorie B en C zit wel een overlap, dit zorgt er waarschijnlijk ook voor dat deze minder goed presenteren in een ML-model.

Hieronder zijn de resultaten weergegeven van de verschillende gebruikte ML-algoritmes: Hier is duidelijk te zien dat met een *Gradient Boosted Decision tree* het beste resultaat

Model\Resultaat	Correct Geclassificeerd	Fout Geclassificeerd	Nauwkeurigheid
<i>Neural Network</i>	47	15	75.8 %
<i>Support Vector Machine</i>	41	21	66.1 %
<i>Gradient Boosted Tree</i>	49	12	79.0 %
<i>K Nearest Neighbors</i>	39	23	62.9 %

TABEL A.1: Nauwkeurigheid resultaten ML-Algoritmes

behaald kon worden. Daarom zal hieronder verder worden ingegaan op de resultaten van de *Gradient Boosted Decision tree*.

A.2.1 Ruwe Resultaten

Werkelijk	Voorspeld	Goed voorspeld	C waarde (in F)
A	A	WAAR	2.00E-06
A	A	WAAR	1.90E-06
A	A	WAAR	1.95E-06
A	A	WAAR	2.10E-06
A	A	WAAR	2.20E-06
A	A	WAAR	2.30E-06
B	C	ONWAAR	2.40E-06
B	B	WAAR	2.80E-06
B	B	WAAR	3.00E-06
B	B	WAAR	3.10E-06
B	B	WAAR	3.15E-06
B	B	WAAR	2.35E-06
B	C	ONWAAR	1.80E-06
B	B	WAAR	1.70E-06
B	C	ONWAAR	1.75E-06
B	C	ONWAAR	1.65E-06
B	B	WAAR	8.00E-07
B	B	WAAR	1.20E-06
B	C	ONWAAR	1.40E-06
B	B	WAAR	1.50E-06
B	C	ONWAAR	1.55E-06
B	B	WAAR	1.00E-06
B	B	WAAR	1.10E-06
B	B	WAAR	1.15E-06
B	B	WAAR	1.00E-06
B	B	WAAR	1.10E-06
B	B	WAAR	1.15E-06
B	B	WAAR	9.00E-07
B	B	WAAR	9.50E-07
B	B	WAAR	8.50E-07
B	B	WAAR	4.00E-07
B	B	WAAR	6.00E-07
B	B	WAAR	7.00E-07
B	B	WAAR	7.50E-07
B	B	WAAR	5.00E-07
B	B	WAAR	5.50E-07
B	B	WAAR	4.50E-07
B	B	WAAR	2.00E-07
B	B	WAAR	3.00E-07
B	B	WAAR	3.50E-07

Werkelijk	Voorspeld	Goed voorspeld	L waarde (in H)
A	A	WAAR	2.15
A	A	WAAR	2.25
A	A	WAAR	2.35
A	A	WAAR	2.45
A	A	WAAR	2.55
C	C	WAAR	0.001
C	C	WAAR	0.005
C	C	WAAR	0.01
C	C	WAAR	0.05
C	C	WAAR	0.15
C	C	WAAR	0.55
C	C	WAAR	0.95
C	B	ONWAAR	1.35
C	B	ONWAAR	1.75
C	B	ONWAAR	1.85
C	B	ONWAAR	1.95
C	B	ONWAAR	2.05
C	A	ONWAAR	2.65
C	C	WAAR	2.75
C	C	WAAR	3
C	B	ONWAAR	5
C	C	WAAR	10

A.2.2 Verwerking

Hieronder zijn de ruwe resultaten verwerkt tot een matrix met de sommen van goede en foute voorspellingen per categorie. Dit kan nog verder verwerkt worden

TABEL A.2: Resultaat matrix DWT analyse

		Voorspelling		
		A	B	C
Werkelijk	A	11	0	0
	B	0	28	6
	C	1	6	10

tot de volgende kern getallen: Wat duidelijk blijkt uit deze dat is dat het model goed

TABEL A.3: Kerngetallen DWT analyse

	Sensitiviteit	Specificiteit	Nauwkeurigheid
A	100%	98%	92%
B	82%	79%	82%
C	59%	87%	63%

is met het onderscheiden van de A categorie met de rest, maar dat het moeite heeft met het onderscheiden van de B en C categorie. En dat valt het meest op met voorspellingen die vallen onder de C categorie dus met de spoel. Als wordt gekeken in tabel A.3, dan valt al snel op dat het model voornamelijk moeite heeft met het onderscheiden op waarden die dicht bij de A categorie liggen. Tussen de 1.35 en 0.95 ligt de ondergrens waarnaar het algoritme beter gaat presteren, ook is er een bovengrens waarnaar het model beter presteert.

Hier uit kan worden geconcludeerd dat deze techniek goed werkt om nominale waarden en afwijkende waarden van elkaar te onderscheiden en om verschillen in spoelwaarden te onderscheiden. Echter is er wel te zien om fouten bij spoelwaarden te detecteren dat het model daar meer moeite mee heeft, behalve als de waarden ver afwijken van de nominale waarde.

Bijlage B

Experiment: Machine Learning bij LC Resonantie Circuits DFT features

In dit experiment zal worden getest of er voor de gebruikte ML algoritmes¹ in dit project een goede score kan worden gehaald. Dus in essentie hoe goed de verschillende fout modi bij deze resonantie circuits te onderscheiden zijn. Daarbij wordt gebruik gemaakt van de Discrete Fourier Transform om de features te creëren.

B.1 Opzet

De opzet is bijna exact hetzelfde als in het experiment bij bijlage A. Het verschil tussen deze twee experimenten is de manier waarop het signaal wordt omgezet in parameters.

B.1.1 Parameters

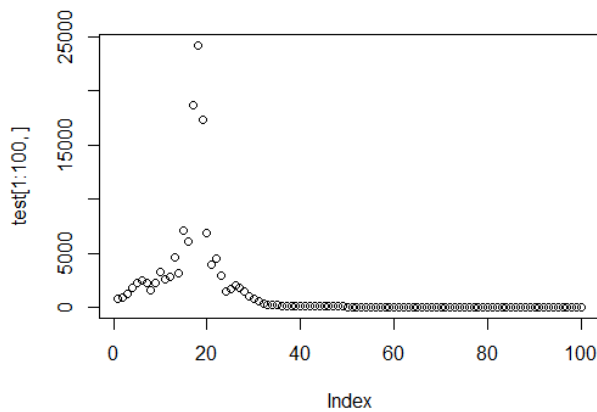
Het signaal dat wordt gemeten in Multisim² zal met behulp van een Discrete Fourier Transformatie worden omgezet naar de magnitude per harmonische.

Naam	Symbool
FT magnitude van parameters	M_n

De magnitude van de eerste 40 parameters van de fourier transformatie zullen worden gebruikt als input voor de verschillende ML-Algoritmes. Er is gekozen voor de eerste 40 omdat in dit bereik de grootste spreiding van resultaten waar te nemen was. Dit is bijvoorbeeld te zien in figuur B.1

¹Het ML-Algoritme zal worden uitgevoerd via KNIME Analytics Platform for Windows 4.1.2 op een Microsoft Surface Pro 7 met Windows 10

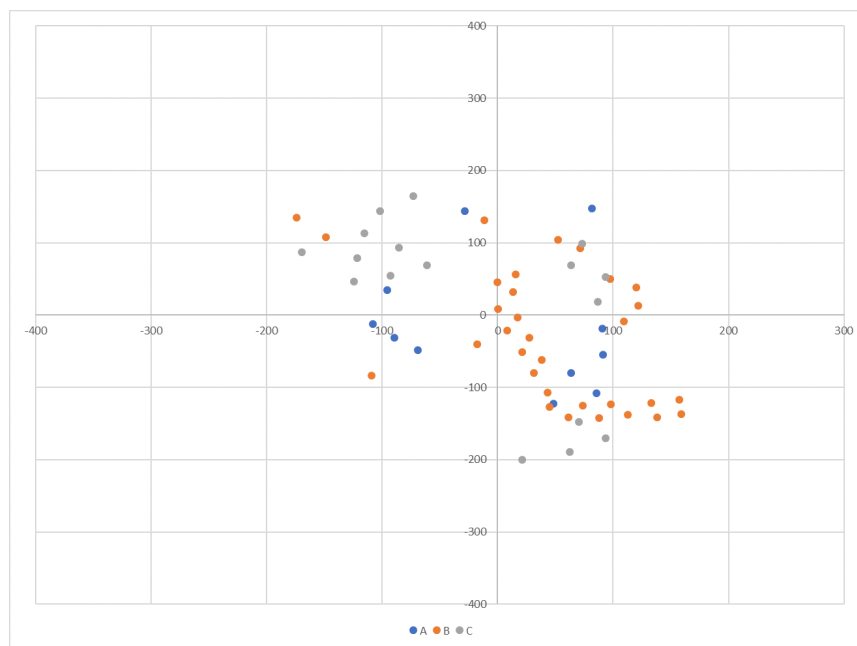
²Het circuit zal worden gesimuleerd in NI Multisim versie 14.1 op een Microsoft Surface Pro 7 met Windows 10



FIGUUR B.1: Plot magnitude eerste 100 parameters FT signaal

B.2 Resultaten

Net zoals in bijlage A zal hieronder een tSNE visualisatie worden gepresenteerd van de resultaten:



FIGUUR B.2: tSNE visualisatie van de features

Zoals te zien in figuur B.2 liggen de resultaten nog meer door elkaar heen dan bij de resultaten van de test in A. Wel is te zien dat er een betere clustering is bij categorie B. Maar aan dit figuur is wel af te leiden dat de resultaten van de ML-modellen slechter zullen zijn dan wanneer een DWT gebruikt wordt om de features te maken.

Hieronder zijn de resultaten gepresenteerd van de verschillende ML-modellen: Het is duidelijk te zien dat de resultaten over het algemeen slechter zijn dan bij het vorige experiment in A. Deze verandering in prestatie is ook volledig te verwijzen naar de gebruikte techniek om het signaal te analyseren.

Model\Resultaat	Correct Geclassificeerd	Fout Geclassificeerd	Nauwkeurigheid
<i>Neural Network</i>	37	25	59.7 %
<i>Support Vector Machine</i>	42	20	67.7 %
<i>Gradient Boosted Tree</i>	46	16	74.2 %
<i>K Nearest Neighbors</i>	29	33	46.8 %

TABEL B.1: Nauwkeurigheid resultaten ML-Algoritmes

Bijlage C

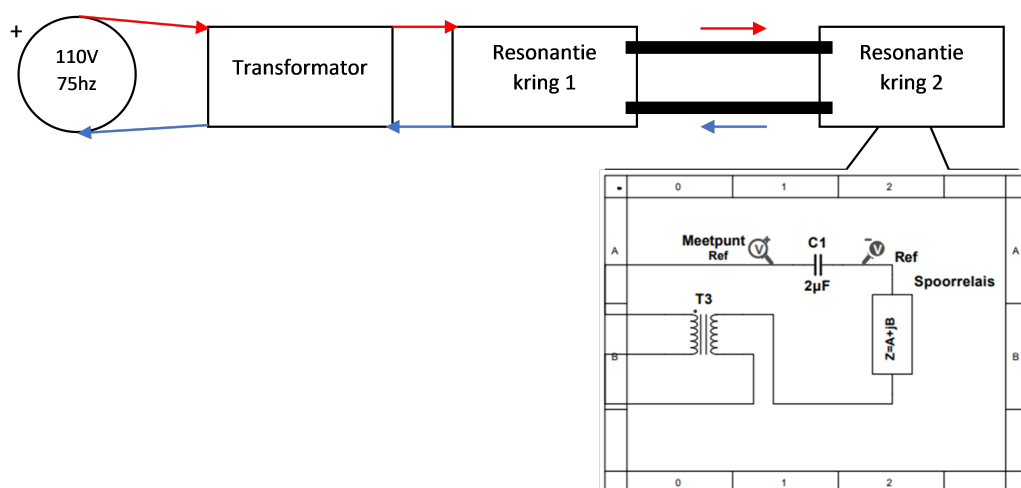
Experiment: Meting fout modi zonder toegevoegd ingang signaal

In dit experiment zal worden getest of het mogelijk is om een meetbaar verschil te kunnen opmerken wanneer alleen gebruik wordt gemaakt van het 75 Hz 110V signaal dat het systeem in wordt gestuurd aan de aansturing kant.

C.1 Opzet

Om een vergelijking te kunne maken tussen de resultaten van de verschillende experimenten die te maken hebben met het bepalen van het ingang signaal, is er voor gekozen om de resultaten te vergelijken door het toepassen van 4 ML-algoritmes deze zijn de K-nearest neighbor, support vector machine, gradient boosted tree en het neural network.

Bij het meten voor dit experiment worden de eisen van hoofdstuk 3.2 in acht genomen. Daarom wordt er gekozen om de spanning over de condensator C3 te meten als input signaal. Deze reactie moet worden omgezet naar parameters die ingevoerd kunnen worden in een ML-model. Het circuit zal worden gesimuleerd in NI Multisim versie 14.1. Het ML-Algoritme zal worden uitgevoerd via KNIME Analytics Platform for Windows 4.1.2. Alle twee de programma zijn uitgevoerd op een Microsoft Surface Pro 7 met windows 10 erop geïnstalleerd.



FIGUUR C.1: Schematische weergave meting

C.1.1 Parameters

Omdat er maar op een enkele frequentie wordt gemeten kan er geen gedrag frequentie afhankelijk gedrag worden waargenomen. Daardoor heeft het uitvoeren van een FFT of DWT geen zin. Voor deze test wordt daarom gekozen om de gemiddelde piek waarde van het signaal te kiezen. En de fase verschuiving ten opzichte van de lokale fase.

Naam	Symbool
Gemiddelde piek spanning	μ_{Vpp}
Faseverandering tov lokale fase	ϕ

$$\mu_{Vpp} = \frac{\sum_{a \in A} a}{|A|} \quad (C.1)$$

Waar

$$A = \{m : m \text{ is de piek waarde van een cyclus van de gemeten sinusoïde}\}$$

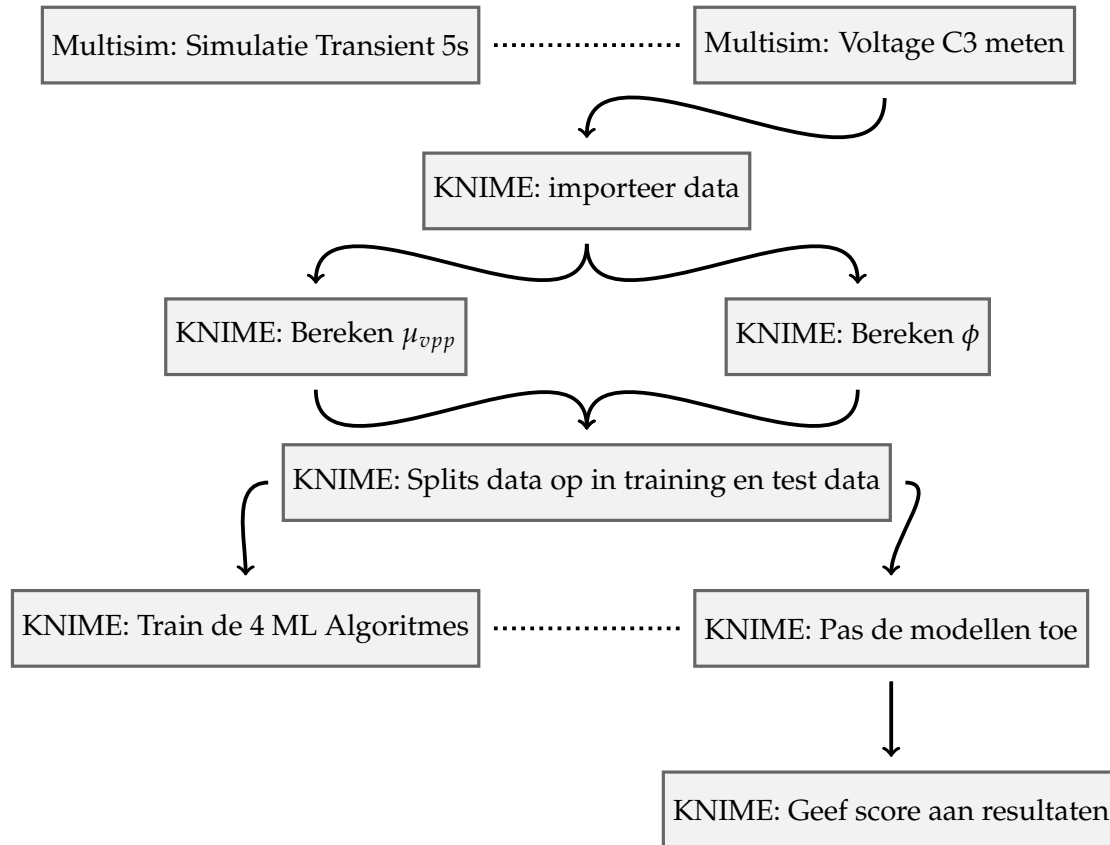
C.1.2 Fout modi

Voor de test worden de volgende componenten worden veranderd van de nominale waarden door ze naar boven (\uparrow) of naar beneden (\downarrow) bij te stellen. Er worden geen fout modi gecombineerd, dit betekent dat er dus vanuit wordt gegaan dat er maar een fout tegelijk ontstaat in het spoorstroomloop systeem.

Component	Verandering	Categorie
Alle	=	A
C1	\uparrow	B
C1	\downarrow	B
C3	\uparrow	C
C3	\downarrow	C
Railspoel AT	\uparrow	D
Railspoel AT	\downarrow	D
Railspoel CT	\uparrow	E
Railspoel CT	\downarrow	E

C.1.3 Flow Diagram

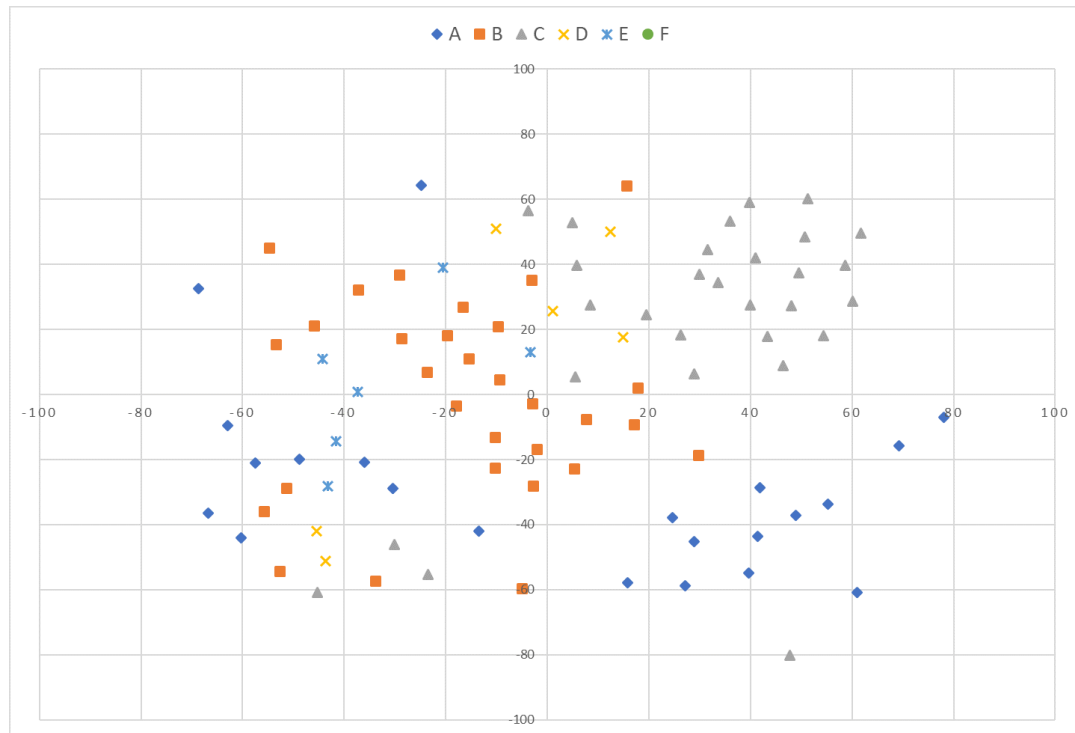
Hieronder is een visuele representatie van de in KNIME en Multisim geprogrammeerde logica om tot een vergelijkbaar resultaat tussen de verschillende algoritmes te komen.



FIGUUR C.2: Stroomdiagram meting zonder toegevoegde bron

C.2 Resultaten

Zoals te lezen in de flowchart worden de gemeten waardes over tijd omgezet naar kenmerkende features voor het ML-Algorithm. Deze zijn voor dit experiment Gemiddelde piek spanning en de faseverandering ten opzichte van de lokale fase. Dit levert de volgende grafiek op:



FIGUUR C.3: t-SNE Representatie van Features

De resultaten van de verschillende ML-algoritmes zijn samengevat in tabel C.1.

Model\Resultaat	Correct Geclassificeerd	Fout Geclassificeerd	Nauwkeurigheid
<i>Neural Network</i>	83	35	70 %
<i>Support Vector Machine</i>	71	47	60 %
<i>Gradient Boosted Tree</i>	82	36	69 %
<i>K Nearest Neighbours</i>	85	33	72 %

TABEL C.1: Nauwkeurigheid resultaten ML-Algoritmes

Aan de resultaten is duidelijk te zien dat de algoritmes nog niet volledig optimaal werken, al kan wel een nauwkeurigheid worden behaald tussen de 67% en de 75%,. In de tabellen af te lezen in hoofdstuk C.4 is duidelijk te zien dat de meeste fouten worden gemaakt bij fout modi die te maken hebben met de spoelen aan beide zeide. Dit komt omdat het gedrag van de spoelen en het gedrag van de condensatoren hetzelfde veroorzaken in amplitude.

Verder als wordt gekeken hoe erg deze methode ervoor zorgt dat de monteur in verwarring komt. Hier zullen de twee best presterende modellen worden vergeleken.

TABEL C.2: Verdeling voorspellingen per zijde bij KNN

	<i>Goed</i>	<i>Fout</i>	<i>Percentage</i>
Fout aan aansturingszijde	62	13	83%
Fout aan de relaiszijde	58	13	82%
Totaal	120	26	82%

Geen Fout	19	4	83%
------------------	----	---	-----

TABEL C.3: Verdeling voorspellingen per zijde bij Neural Network

	<i>Goed</i>	<i>Fout</i>	<i>Percentage</i>
Fout aan aansturingszijde	64	14	82%
Fout aan de relaiszijde	56	14	80%
Totaal	120	28	81%

Geen Fout	21	5	81%
------------------	----	---	-----

C.3 Conclusie

Uit de resultaten kan worden geconcludeerd dat bij deze methode de K Nearest Neighbours model het beste werkt. Deze kan met een zekerheid 72% de verschillende categorieën onderscheiden. Daarnaast presteert het KNN model ook het beste bij het bepalen welke resonantie kring de storing is opgetreden.

C.4 Data

Label\Voorspelling	A	B	C	D	E
A	21	1	0	0	0
B	1	24	3	1	2
C	0	3	23	4	1
D	0	1	8	8	0
E	4	2	3	1	7

TABEL C.4: Neural Network

Label\Voorspelling	A	B	C	D	E
A	16	1	0	1	4
B	1	25	2	2	1
C	1	2	22	6	0
D	0	2	13	2	0
E	3	5	3	0	6

TABEL C.5: SVM

Label\Voorspelling	A	B	C	D	E
A	19	0	0	0	3
B	1	24	1	2	3
C	0	0	20	8	3
D	0	2	4	11	0
E	3	4	2	0	8

TABEL C.6: Gradient Boosted Decision Tree

Label\Voorspelling	A	B	C	D	E
A	19	0	0	0	3
B	0	25	1	2	3
C	0	2	24	4	1
D	0	2	4	11	0
E	4	5	2	0	6

TABEL C.7: K Nearest Neighbors

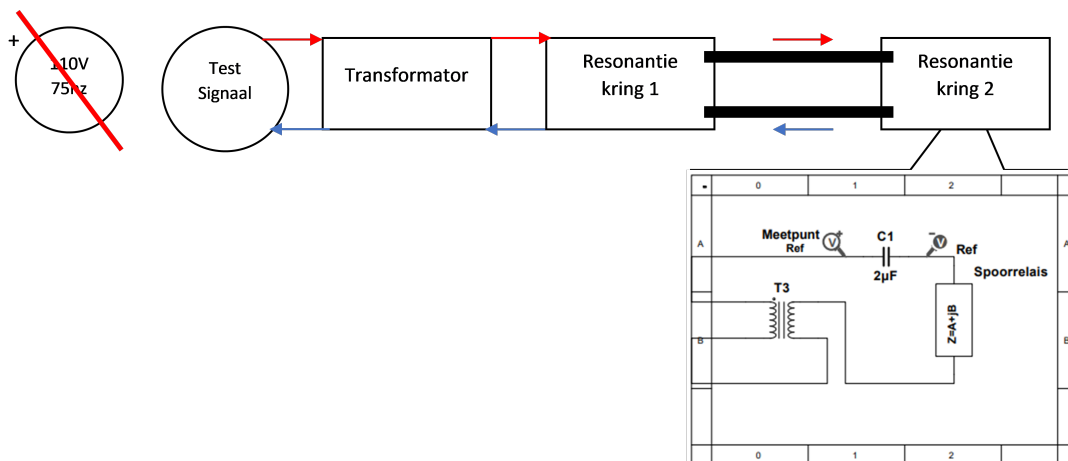
Bijlage D

Experiment: Meting fout modi vervaning ingangssignaal

In dit experiment zal worden getest of het veranderen van de 110v bron 75hz naar een chrip signaal een verbetering van de resultaten gepresenteerd in bijlage C kan worden behaald. Hierbij zal weer over de condensator C3 worden gemeten. Deze bron zal een signaal van 100v geven, en zal van 50Hz lineair oplopen tot 150 Hz.

D.1 Opzet

Zoals beschreven in C.1 is het uiteindelijke resultaat van dit experiment een nauwkeurigheid score voor vier ML-modellen. In figuur D.1 is een schematische weergave weergegeven van de opstelling in dit experiment. Zo is te zien dat de originele 110v 75Hz signaal wordt losgekoppeld, verder wordt ook weer over dezelfde condensator gemeten als in het experiment gepresenteerd in bijlage C.



FIGUUR D.1: Schematische weergave meting

D.1.1 Parameters

Voor dit experiment wordt een veranderde frequentie gebruikt om het gedrag van het circuit te bestuderen. Dit betekent dus dat over verschillende frequenties kan worden bekeken hoe het circuit reageert. Om het signaal te analyseren zal gebruik worden gemaakt van twee parameters: de energie van het signaal op verschillende decompositie niveaus en de verdeling van de resultaten genaamd de kurtosis.

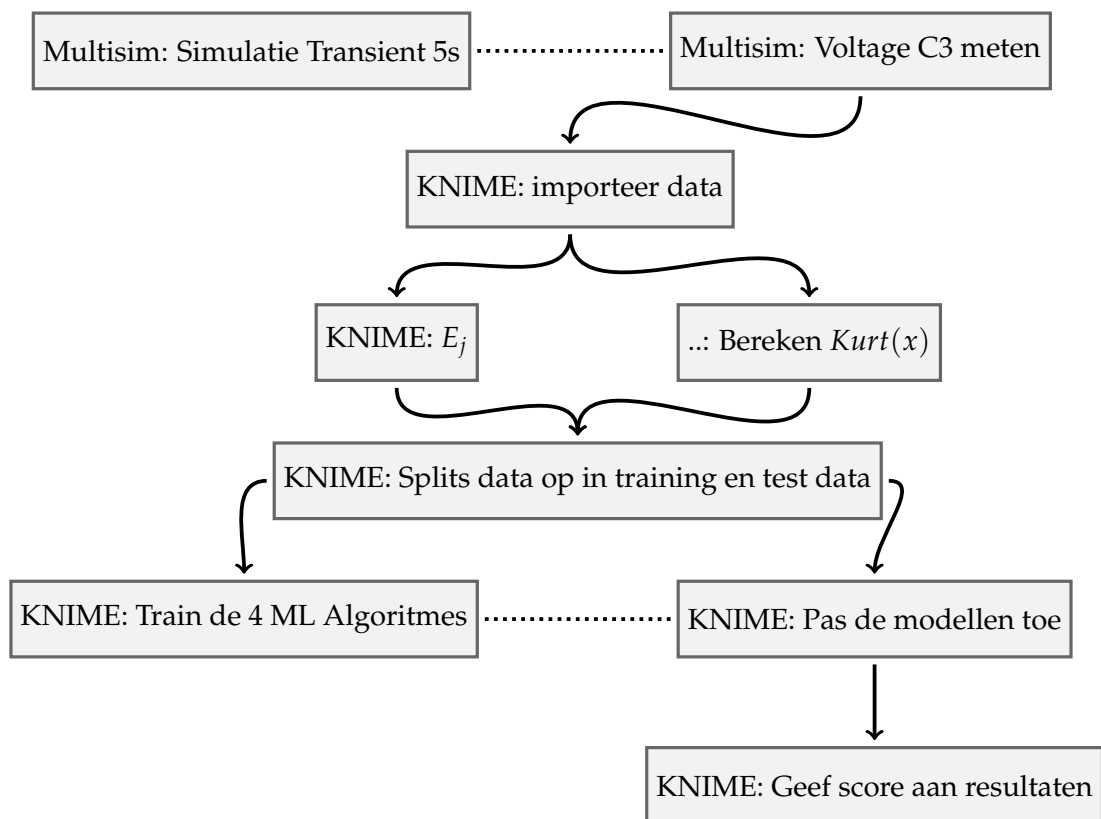
Naam	Symbool
WT energie op decomposities niveaus	E_j
Kurtosis	$Kurt(x)$

D.1.2 Fout modi

Dezelfde foutmodi zoals gepresenteerd in experiment weergegeven in bijlage C zullen gebruikt worden.

D.1.3 Stroomdiagram

Hieronder is een stroomdiagram weergegeven van het proces.



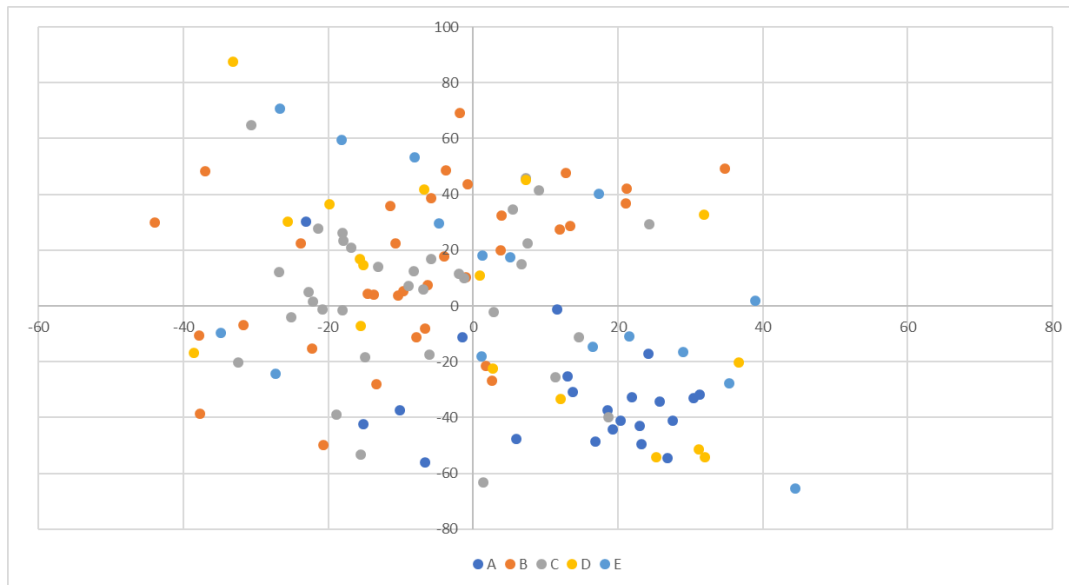
FIGUUR D.2: Stroomdiagram meting zonder toegevoegde bron

D.2 Resultaten

De verkregen features E_j en $Kurt(x)$ kunnen worden gerepresenteerd in een t-SNE weergaven figuur D.3. In deze weergaven is weer te zien dat de resultaten van fout modi A, B en C in zekere zin te scheiden is van elkaar. Maar wat opvalt is dat D en E dichtbij elkaar liggen en ook overlap hebben. Hierdoor is het dus moeilijk om een model te ontwikkelen die goed deze resultaten kan scheiden. Daarom wordt verwacht dat het identificeren van deze fout modi problemen gaat opleveren.

De ML-algoritmes hebben de volgende resultaten:

Er is duidelijk te zien dat deze resultaten minder goed zijn dan de resultaten in bijlage C, dit komt waarschijnlijk door de bovengenoemde redenen. Als ook wordt



FIGUUR D.3: t-SNE weergaven van features

Model\Resultaat	Correct Geclassificeerd	Fout Geclassificeerd	Nauwkeurigheid
Neural Network	68	56	55 %
Support Vector Machine	68	56	55 %
Gradient Boosted Tree	79	45	64 %
K Nearest Neighbors	69	55	56 %

TABEL D.1: Nauwkeurigheid resultaten ML-Algoritmes

gekeken naar de verschillende resultaten in hoofdstuk D.4 dan is ook duidelijk te zien dat de algoritmes niet de fout modi D en E kunnen onderscheiden van de andere fout modi.

Hieronder zijn de resultaten van de tweede test te vinden, waarbij dus wordt gekeken naar hoe vaak de monteur op het verkeerde been wordt gezet door het algoritme.

TABEL D.2: Verdeling voorspellingen per zijde bij KNN

	<i>Goed</i>	<i>Fout</i>	<i>Percentage</i>
Fout aan aansturingszijde	49	39	56%
Fout aan de relaiszijde	50	39	56%
Totaal	99	78	56%

Geen Fout	19	3	86%
------------------	----	---	-----

D.3 Conclusie

Er kan worden geconcludeerd dat deze methode niet erg goed werkt. De op 1 na beste algoritme doet het amper beter dan random kans en de beste methode presteert ook heel vaak niet zo goed. Uit verdere analyse van de resultaten blijkt wel dat

TABEL D.3: Verdeling voorspellingen per zijde bij Gradient boosted decision tree

	<i>Goed</i>	<i>Fout</i>	<i>Percentage</i>
Fout aan aansturingszijde	56	29	66%
Fout aan de relaiszijde	52	29	64%
Totaal	108	58	65%

Geen Fout	19	3	86%
------------------	----	---	-----

het model puur op basis van de het inschatten van categorie A wel goede resultaten laat zien.

D.4 Data

Label\Voorspelling	A	B	C	D	E
A	20	1	0	0	1
B	3	22	4	3	2
C	2	6	20	6	0
D	1	3	4	3	6
E	2	2	4	6	3

TABEL D.4: Neural Network

Label\Voorspelling	A	B	C	D	E
A	18	2	2	0	0
B	2	21	5	5	1
C	3	2	28	0	1
D	1	7	3	0	6
E	0	8	4	4	1

TABEL D.5: SVM

Label\Voorspelling	A	B	C	D	E
A	19	2	1	0	0
B	2	24	3	2	3
C	2	1	29	1	1
D	1	3	2	5	6
E	1	4	2	8	2

TABEL D.6: Gradient Boosted Decision Tree

Label\Voorspelling	A	B	C	D	E
A	19	2	1	0	0
B	2	22	4	5	1
C	1	4	25	1	3
D	2	6	3	1	5
E	0	6	2	7	2

TABEL D.7: K Nearest Neighbors

Bijlage E

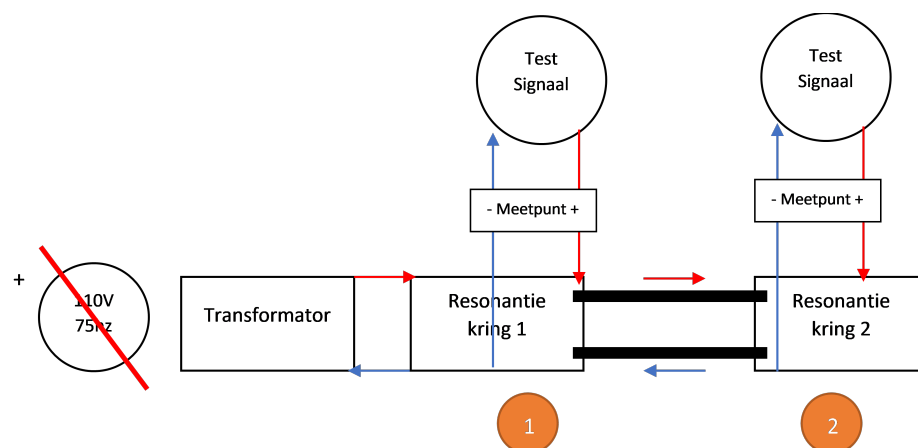
Experiment: Test met getrapte metingen

In dit experiment zal worden geprobeerd of er betere resultaten kunnen worden gehaald door een meer praktische oplossing, omdat voor de oplossing vanaf buiten gemeten kan worden. Echter wordt wel de voedingsbron losgekoppeld in dit experiment omdat dat dan beter de resultaten kunnen worden vergeleken en de prestaties van deze methode beter zichtbaar zijn.

E.1 Opzet

Hetzelfde doel wordt aangehouden als in bijlage C en D. De opbouw van dit experiment is te zien in figuur E.1. Deze meting is getrapd omdat er 2 metingen nodig zijn, een aan de aansturing zijde en een aan de relais zijde. Het idee achter deze techniek is dat monteur een meting kan doen op de plek waar hij vermoed dat het probleem zit. En als deze meting geen uitsluitsel geeft kan de monteur naar de andere zijde lopen en proberen te achterhalen of het probleem zich daar bevindt.

Er zal ook worden getest of het mogelijk is om aan een zijde alle fout modi worden herkend, echter is de verwachting dat dit zal zorgen voor slechtere resultaten.



FIGUUR E.1: Schematische weergave meting

E.1.1 Parameters

Dezelfde parameters als in D zullen worden gebruikt.

E.2 Fout modi

Dezelfde fout modi als in C zullen worden gebruikt. Er zal wel op 2 verschillende manieren geprobeerd worden input te leveren voor de ML-modellen. Er zal geen verschil zijn met de metingen, maar wel op de manier hoe data wordt gelabeld. Hieronder zijn de 2 verschillende manieren gepresenteerd.

E.2.1 Volledig methode 3a

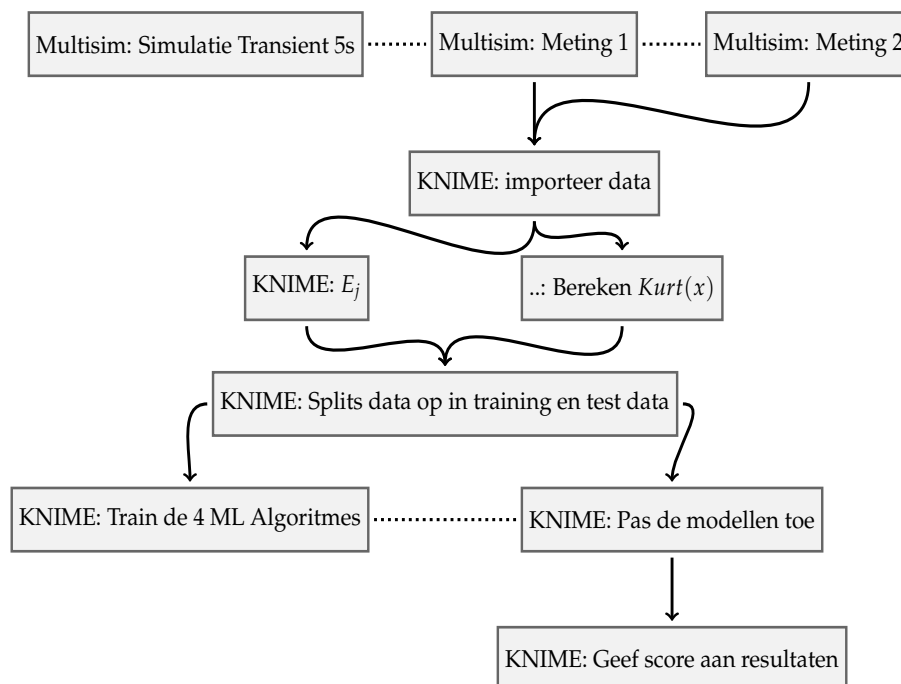
Hierbij worden alle fout modi aangeboden aan het ML-Model, dit betekent dus dat fout modi A, B, C, D en E bij beide zijde aan het ML-algoritme worden aangeboden.

E.2.2 Prioriteit op resonantie kring methode 3b

Bij deze methode wordt prioriteit gegeven aan de resonantie kring waar gemeten wordt. Dus als wordt gemeten aan resonantie kring 1 zal alleen fout modi A, C en D worden aangeboden de overige fout categorieën worden gedefinieerd als A.

E.3 Stroomdiagram

In dit stroomdiagram is te zien hoe het proces verloopt om resultaten te verkrijgen. Het verschil met het vorige experiment is dat in deze meting dus 2 metingen worden gedaan en worden ingeladen.



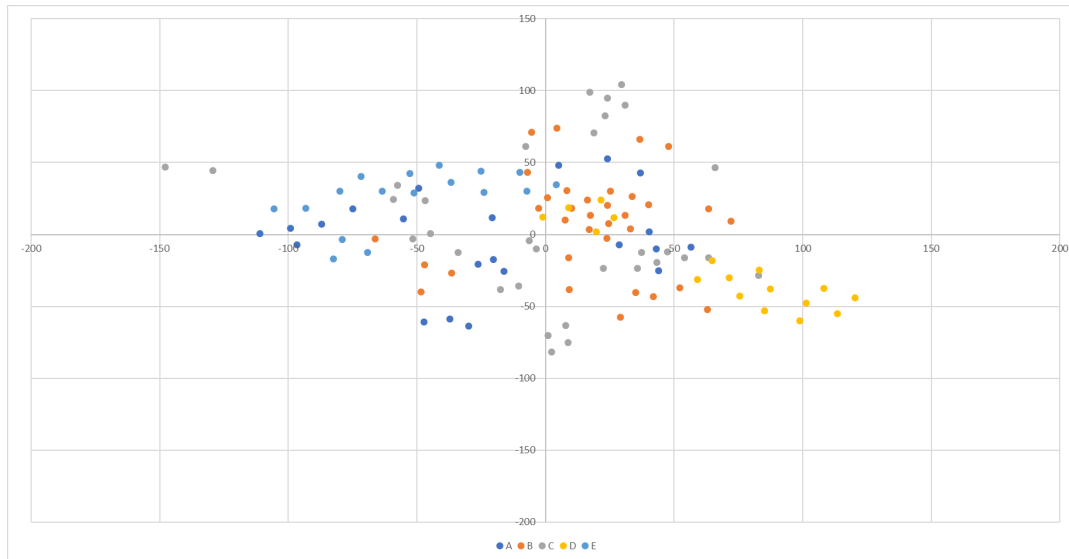
FIGUUR E.2: Stroomdiagram meting zonder toegevoegde bron

E.4 Resultaten

Er zijn 2 soorten resultaten een waarbij de fout modi volledig is gelabeld en de ander waarbij alleen de fout modi is gelabeld die gerelateerd is aan de resonantie kring die dichtbij het meet punt ligt.

E.4.1 Volledig

Hieronder zijn de resultaten van de ML-modellen die als invoer de volledig fout modi lijst.



FIGUUR E.3: t-SNE weergaven van features

Duidelijk te zien is dat er een paar patronen in de dat zitten zo zit bijvoorbeeld categorie E vaak in het 2^{de} kwadrant, en zo een patroon is ook te zien bij de categorie D. Echter is er wel veel overlap tussen de verschillende categorieën, zo liggen veel van de resultaten in het midden van het coördinaten stelsel.

Het is voor een betere besluitvorming om de 2 best presenterende modellen ook in

Model\Resultaat	Correct Geclassificeerd	Fout Geclassificeerd	Nauwkeurigheid
Neural Network	88	36	71 %
Support Vector Machine	92	32	74 %
Gradient Boosted Tree	92	32	74 %
K Nearest Neighbours	94	30	76 %

TABEL E.1: Nauwkeurigheid resultaten ML-Algoritmes

te delen op waar de fout is ontstaan. Hieronder zijn de geclusterd of de voorspelling wel aan de juiste fysieke locatie zat. Dit betekent dat wanneer een voorspelling in B en E ligt dus de relais zijde van het systeem dat deze worden opgeteld als een juiste zijde voorspelling. Hetzelfde geldt voor de C en D die fout modi in de eerste resonantie kring aan de aansturingszijde weergeven.

	<i>Goed</i>	<i>Fout</i>	<i>Percentage</i>
Fout aan aansturingszeide	44	15	75%
Fout aan de relaiszijde	44	13	77%
Totaal	88	28	76%

Geen Fout	12	6	67%
------------------	----	---	-----

TABEL E.2: Verdeling voorspellingen per zijde bij KNN

	<i>Goed</i>	<i>Fout</i>	<i>Percentage</i>
Fout aan aansturingszeide	44	9	83%
Fout aan de relaiszijde	46	11	81%
Totaal	90	20	82%

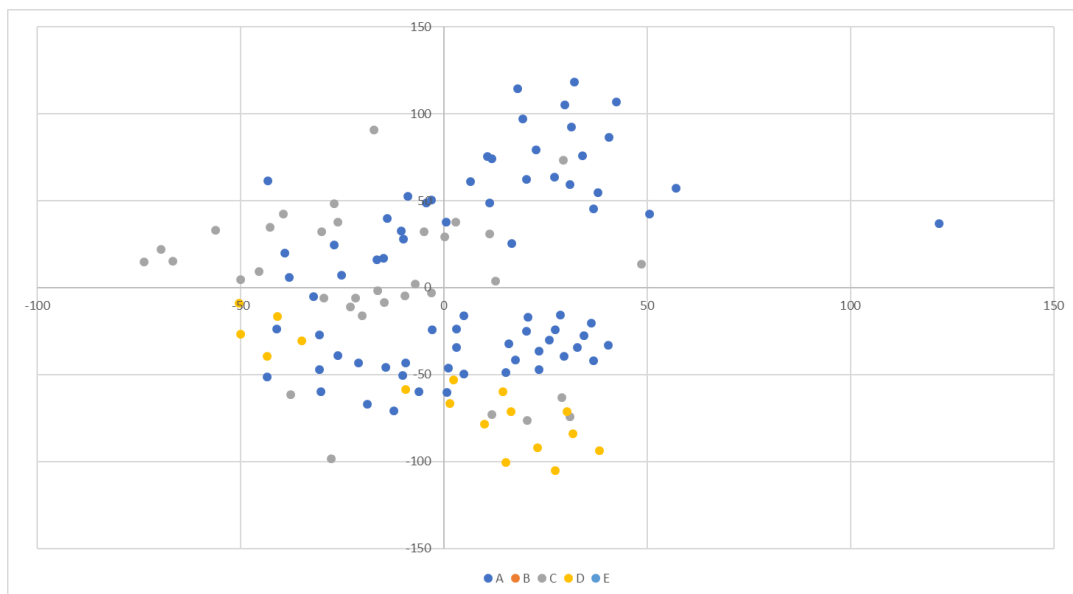
Geen Fout	14	8	64%
------------------	----	---	-----

TABEL E.3: Verdeling voorspellingen per zijde bij c

Bij deze vergelijking is te zien dat de Gradient boosted decision tree beter is in het voorspellen van de zijde dan het KNN model.

E.4.2 Resonantie kring prioriteit

In figuur E.4 zijn de resultaten weergegeven van de ML-modellen die de aangepaste invoer data hebben gehad. Via deze methode is al een veel betere scheiding te zien



FIGUUR E.4: t-SNE weergaven van features

van categorieën. Er is nog steeds een grote overlap met categorie D en A, natuurlijk staat hier wel tegenover de *trade-off* van het mogelijk maken van het herkennen van alle fout-modi.

Model\Resultaat	Correct Geclassificeerd	Fout Geclassificeerd	Nauwkeurigheid
<i>Neural Network</i>	99	25	80 %
<i>Support Vector Machine</i>	99	25	80 %
<i>Gradient Boosted Tree</i>	102	22	82 %
<i>K Nearest Neighbours</i>	103	21	83 %

TABEL E.4: Nauwkeurigheid resultaten ML-Algoritmes

In de resultaten is ook duidelijke verbetering te zien ten op zichten van de methode waarbij geprobeerd wordt om alle fout modi te herkennen.

Netzoals bij de analyse van de volledige dataset is het ook interessant om de data te vertalen in meer vergelijkings parameters. Omdat bij deze metingen alleen maar gezocht wordt naar een fout in de resonantiekring waaraan gemeten wordt, kan alleen maar worden gepresenteerd hoe goed het model is in het voorspellen van een fout in die resonantie kring en het voorspellen of een fout niet bij die resonantie kring ligt.

	<i>Goed</i>	<i>Fout</i>	<i>Percentage</i>
Fout	45	15	75%
Geen fout aan gemeten resonantie kring	64	6	91%

TABEL E.5: Verdeling voorspellingen in de resonantie kring of daarbuiten bij KNN

	<i>Goed</i>	<i>Fout</i>	<i>Percentage</i>
Fout	42	18	70%
Geen fout aan gemeten resonantie kring	64	9	88%

TABEL E.6: Verdeling voorspellingen in de resonantie kring of daarbuiten bij Gradient boosted decision tree

Hier is te zien dat het KNN model beter is in het exact voorspellen van de fout modus, maar dat het ook beter is in het voorspellen of de fout daadwerkelijk in de gemeten resonantiekring ligt.

E.4.3 Conclusie

Het bepalen welke methode beter is lastiger dan de alleen kijken welke methode nauwkeuriger is. Er moet namelijk rekening worden gehouden met die belangrijke *trade-off* tussen nauwkeurigheid en gebruiksgemak.

Maar het volgende kan worden gezegd over de methode waarbij alle fout modi worden meegegeven aan het ML-model. De K Nearest Neighbours(KNN) methode werkt het beste en is het meest exact in het voorspellen van de juiste fout modus. Met de werkelijke situatie in gedachte waarbij het meer preferabel is dat het model wanneer het een foute voorspelling doet wel de zijde van waar de fout zit goed voorspelt. Dat in gedachte meegenomen is het Gradient boosted decision tree model beter in voorspellen.

Bij de andere methode is wel duidelijk te zien dat het KNN model beter geschikt is voor het voorspellen van de fout modi.

E.5 Data (Wordt nog uitgebreid met CT info)

	<i>Volledig</i>						<i>Resonantie Kring Prioriteit</i>				
label\voorspelling	A	B	C	D	E		A	B	C	D	E
A	11	2	3	4	2		62	0	7	4	0
B	4	28	2	0	0		0	0	0	0	0
C	4	0	27	3	0		7	0	25	2	0
D	4	1	2	8	2		2	0	3	12	0
E	0	0	1	2	14		0	0	0	0	0

TABEL E.7: Resultaten Neural Network

	<i>Volledig</i>						<i>Resonantie Kring Prioriteit</i>				
label\voorspelling	A	B	C	D	E		A	B	C	D	E
A	10	4	4	2	2		61	0	9	3	0
B	0	33	1	0	0		0	0	0	0	0
C	6	0	26	2	0		4	0	29	1	0
D	2	0	5	9	1		3	0	5	9	0
E	0	0	2	1	14		0	0	0	0	0

TABEL E.8: Resultaten SVM

	<i>Volledig</i>						<i>Resonantie Kring Prioriteit</i>				
label\voorspelling	A	B	C	D	E		A	B	C	D	E
A	14	3	4	1	0		64	0	5	4	0
B	0	33	1	0	0		0	0	0	0	0
C	3	0	29	2	0		6	0	26	2	0
D	3	0	4	9	1		3	0	2	12	0
E	2	0	0	2	13		0	0	0	0	0

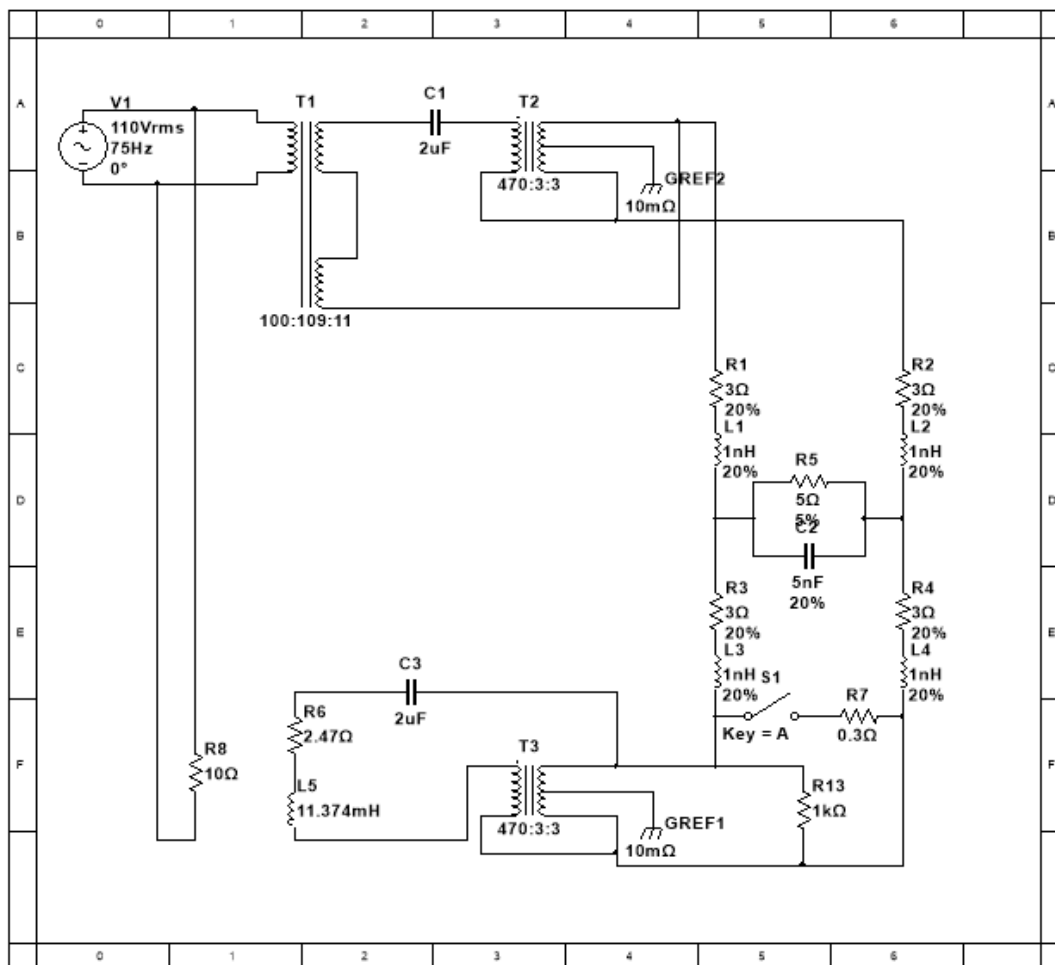
TABEL E.9: Resultaten Gradient Boosted Decision Tree

	<i>Volledig</i>						<i>Resonantie Kring Prioriteit</i>				
label\voorspelling	A	B	C	D	E		A	B	C	D	E
A	12	2	5	1	2		64	0	6	3	0
B	3	30	1	0	0		0	0	0	0	0
C	1	2	29	2	0		2	0	31	1	0
D	1	1	3	10	2		4	0	5	8	0
E	1	1	1	1	13		0	0	0	0	0

TABEL E.10: Resultaten K Nearest Neighbors

Bijlage F

Gebruikte model voor simulaties



FIGUUR F.1: Vervangingschema gebruikt voor de simulaties van de SSL

Bijlage G

Gebruikte Code - Omzetten Features

```

library(wavelets)
library(e1071)
Y <- knime.in$"Y"
X <- knime.in$"X"
begRow <- 0
numRow <- 1200
Y <- Y[begRow:numRow]
X <- X[begRow:numRow]

df.wt <- dwt(Y, n.levels=4, boundary="periodic", fast=FALSE)

noiseVariance <- function(data.wt){
  return((median(abs(data.wt)))/0.6745)
}
noiseTresholding <- function(noiseVar, decompositionLevel){
  return(noiseVar*sqrt(2*log(2^(decompositionLevel))))
}

# df = data.frame
# wt = Wavelet Transformed
# nv = Noise Variance
# nt = Noise Treshold
# E = Energy level of coefficient level
# kurt = Kurtosis

calculateEnergyAtWTDecompositionLevel <- \
\ function(data, decompostionlayer){
  data.r <- data
  df.wt.nv.f <- noiseVariance(data.r)
  df.wt.nt.f <- noiseTresholding(df.wt.nv.f, decompostionlayer)
  data.r[!(abs(data.r) > df.wt.nt.f)] = 0
  data.r <- sign(data.r)*(abs(data.r)-df.wt.nt.f)
  return(sum(abs(data.r)^2))
}

#compute Shannon entropy
entropy <- function(target) {

```

```
freq <- table(target)/length(target)
# vectorize
vec <- as.data.frame(freq)[,2]
#drop 0 to avoid NaN resulting from log2
vec<-vec[vec>0]
#compute entropy
return(-sum(vec * log2(vec)))
}

df.wt.1.E <- calculateEnergyAtWTDecompositionLevel(df.wt@V$V1, 1)
df.wt.2.E <- calculateEnergyAtWTDecompositionLevel(df.wt@V$V2, 2)
df.wt.3.E <- calculateEnergyAtWTDecompositionLevel(df.wt@V$V3, 3)
df.wt.4.E <- calculateEnergyAtWTDecompositionLevel(df.wt@V$V4, 4)

df.kurt <- kurtosis(Y)

df.ent <- entropy(Y)

result <- data.frame('EnergyWtDecompositionLevel' = \
\ data.frame('dl_1' = df.wt.1.E, 'dl_2' = df.wt.2.E, 'dl_3' = df.wt.3.E,
\
\ }'dl_4' = df.wt.4.E), 'Kurtosis' = df.kurt, 'Entropy' = df.ent)

knime.out <- result
```

Bijlage H

Competenties

In deze bijlage zal ik toelichten waarom ik heb voldaan aan de vereiste competenties van het afstuderen bij de opleiding Elektrotechniek.

H.1 Analyseren

Ik ben van mening dat in deze scriptie duidelijk te lezen is dat deze competenties op niveau drie is behaald. Ik kreeg mijn afstudeer vraagstuk niet op een serveerblaadje toegereikt. De vraag die mij namelijk gesteld werd was "Is het mogelijk door twee kabels aan het spoor te bevestigen te identificeren welke component er kapot is gegaan. Ik heb hier zelf de keuze gemaakt om aan de slag te gaan met het ontwikkelen van een methode op basis van machine learning. Dit was voor mij nieuw terein en vereiste een multidisciplinaire aanpak. Aangezien ik mijn domein kennis van elektrotechniek moest linken aan de complexe mogelijkheden van ML-modellen.

In deze scriptie neem ik daarom ook een grote diepgang op de ML technieken die ik gebruikt heb. Vanuit de opgedane kennis uit die diepgang kon toewerken naar een methode waarbij ik de link kon leggen tussen de signaal metingen en verwerking en de ML-modellen. Daardoor had ik een duidelijke richting hoe dit onderzoek moest verlopen.

H.2 Ontwerpen

In deze scriptie heb ik een nieuwe methode ontworpen die nog onbekend is bij ProRail. Daarvoor heb ik een complexe en multidisciplinaire aanpak gebruikt, door een koppeling te maken met machine learning. Ik heb daarvoor een literatuurstudie gedaan om te achterhalen hoe zo een methode eruit moet komen te zien. Zo heb ik vanuit verschillende bronnen de koppeling gemaakt tussen de beschreven technieken en de toepasbaarheid bij de spoorstroomloop. Daarvoor moest ik een de door mij opgestelde eisen aan het product vertalen naar een mogelijk oplossing voor dit systeem

Ik heb daarvoor zelf keuzes gemaakt hoe deze methode getest moest worden. Zo heb ik bijvoorbeeld multisim gebruikt om van een abstracte opbouw van de spoorstroomloop een model te maken die te simuleren was. Verder heb ik ook KNIME gebruikt om de processen omtrent de ML te modelleren en toe te passen voor de SSL.

Verder heb ik een reeks duidelijke onderzoeksvragen opgesteld die als kader voor dit verslag dienen. De antwoorden op deze vragen konden de verschillende elementen in deze scriptie worden vergeleken en getoetst.

H.3 Realiseren

Voor het realiseren van de methode heb ik zoals in de vorige paragraaf toegelicht verschillende hulpmiddelen gebruikt. De resultaten van deze verschillende methoden heb ik gebruikt om de methode die ik ontworpen hebt te verifiëren.

H.4 Managen

Ik was zelf verantwoordelijk voor het verloop van dit proces en de indeling van mijn tijd. Ik heb van ProRail veel vrijheid gekregen om zelf te bepalen wanneer ik wat ging doen. Met deze vrijheid ben ik verstandig omgegaan en zo heb ik toch de doelen behaald die ik wilde behalen.

Verder heb ik door het Covid-19 virus laten zien dat ik capabel genoeg ben om snel te schakelen en de opdracht waar ik aan werk op een dusdanige manier aan te passen dat hij nog steeds van niveau is voor het afstuderen. Verder heb ik zelf verantwoordelijkheid genomen om contact met contact met collega's om de ontwerpen die zeer gericht waren op het spoor beter te begrijpen.

H.5 Onderzoeken

Mijn scriptie is zeer zwaar beladen met onderzoek, omdat de gepresenteerde strategie en methodes nieuw zijn voor ProRail. Ik heb in dit document laten zien dat ik bij een vraagstuk de juiste literatuur kan vinden, en deze kan omzetten naar een bruikbare ontwerp en realisatie. Hiervoor heb ik wetenschappelijke bronnen gelezen en geselecteerd op bruikbaarheid.

Verder heb ik laten zien dat ik de behaalde resultaten kan samenvatten en interpreteren aan de hand van statistiek, en vanuit die resultaten conclusies kan trekken voor de gekozen onderzoeksvragen. Daar heb ik ook duidelijk laten zien dat ik duidelijk aangeef wat ik onderzoek en waar de grenzen liggen van mijn conclusies.

Bibliografie

- [1] Ke Huang, Haralampos-G. Stratigopoulos en Salvador Mir, *Fault diagnosis of analog circuits based on machine learning*. IEEE, 2010. DOI: [10.1109/DATE.2010.5457099](https://doi.org/10.1109/DATE.2010.5457099).
- [2] Y. Deng', Y. He' en Y. Sun, „Fault Diagnosis of Analog Circuits with Tolerances Using Artificial Neural Networks”, tech. rap.
- [3] T. De Bruin, K. Verbert en R. Babuska, „Railway Track Circuit Fault Diagnosis Using Recurrent Neural Networks”, *IEEE Transactions on Neural Networks and Learning Systems*, jrg. 28, nr. 3, p. 523–533, mrt 2017, ISSN: 21622388. DOI: [10.1109/TNNLS.2016.2551940](https://doi.org/10.1109/TNNLS.2016.2551940).
- [4] J. Chen, C. Roberts en P. Weston, „Fault detection and diagnosis for railway track circuits using neuro-fuzzy systems”, *Control Engineering Practice*, jrg. 16, nr. 5, p. 585–596, mei 2008, ISSN: 09670661. DOI: [10.1016/j.conengprac.2007.06.007](https://doi.org/10.1016/j.conengprac.2007.06.007).
- [5] K. Verbert, B. De Schutter en R. Babuška, „Exploiting spatial and temporal dependencies to enhance fault diagnosis: Application to railway track circuits * Exploiting Spatial and Temporal Dependencies to Enhance Fault Diagnosis: Application to Railway Track Circuits”, tech. rap., 2015, p. 3052–3057.
- [6] S. H. Visser A.J., *Spoorstroomlopen Opleiding en Vorming*, 1ste ed. Nederlandse Spoorwegen, 1981, p. 1–154. adres: <https://www.irse.nl/resources/Spoorstroomlopen.pdf>.
- [7] Spoors R, *General Information on Track Circuits*. London: Railroad PLC, 1998.
- [8] AM Treinbeveiliging, *Ontwerpvoorschrift: Laagfrequent spoorstroomlopen 75 Hz secties in ET-gebied*. Utrecht: ProRail, 2013.
- [9] F. Aminian, M. Aminian en H.W. Collins, „Analog fault diagnosis of actual circuits using neural networks”, *IEEE Transactions on Instrumentation and Measurement*, jrg. 51, nr. 3, p. 544–550, 2002. DOI: [10.1109/TIM.2002.1017726](https://doi.org/10.1109/TIM.2002.1017726). adres: <https://ieeexplore.ieee.org/document/1017726>.
- [10] Keith McNulty, *What is Machine Learning?*, 2018. adres: <https://towardsdatascience.com/what-is-machine-learning-891f23e848da>.
- [11] A. Walker, W.E. Alexander en P.K. Lala, „Fault diagnosis in analog circuits using element modulation”, IEEE, 1992, p. 19–29. DOI: [10.1109/54.124514](https://doi.org/10.1109/54.124514).
- [12] A. Masri, *How Do Machines Learn?*, 2019. adres: <https://towardsdatascience.com/how-do-machines-learn-561181ed209a>.
- [13] A. S. S. Vasan, B. Long en M. Pecht, „Diagnostics and prognostics method for analog electronic circuits”, *IEEE Transactions on Industrial Electronics*, jrg. 60, nr. 11, p. 5277–5291, 2013, ISSN: 02780046. DOI: [10.1109/TIE.2012.2224074](https://doi.org/10.1109/TIE.2012.2224074).
- [14] M. Aminian en F. Aminian, „A modular fault-diagnostic system for analog electronic circuits using neural networks with wavelet transform as a preprocessor”, *IEEE Transactions on Instrumentation and Measurement*, jrg. 56, nr. 5, p. 1546–1554, okt 2007, ISSN: 00189456. DOI: [10.1109/TIM.2007.904549](https://doi.org/10.1109/TIM.2007.904549).

- [15] R. Spina en S. Upadhyaya, „Linear Circuit Fault Diagnosis Using Neuromorphic Analyzers”, tech. rap. 3, 1997.
- [16] Glosser.ca, *Artificial neural network.svg*. adres: <https://commons.wikimedia.org/w/index.php?curid=24913461>.
- [17] D. Wilimitis, *The Kernel Trick in Support Vector Classification*, 2018. adres: <https://towardsdatascience.com/the-kernel-trick-c98cdbcae3f>.
- [18] R. Gandhi, *Support Vector Machine — Introduction to Machine Learning Algorithms*, 2018. adres: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.
- [19] AM Treinbeveiligingssystemen, *Onderhoudsdocument: Laagfrequente spoorstroomloopen 75Hz secties in ET-gebied*. ProRail, 2020.
- [20] Erdogan S, Ozev S en Cauvet P, „Diagnosis of Assembly Failures for System-in-Package RF Tuners”, *2008 IEEE International Symposium on Circuits and Systems*, p. 2286–2289, 2008.
- [21] É. F. Cota, M. Negreiros, L. Carro en M. Lubaszewski, „A New Adaptive Analog Test and Diagnosis System”, tech. rap. 2, 2000, p. 223.
- [22] E. Simeu en S. Mir, „Parameter Identification Based Diagnosis in Linear and Nonlinear Mixed-Signal Systems”, tech. rap.
- [23] A. A. K. Mohsen en M. F. Abu El-Yazeed, „Selection of Input Stimulus for Fault Diagnosis of Analog Circuits Using ARMA Model”, tech. rap., 2004, p. 212–217. adres: <http://www.elsevier.de/aeue>.
- [24] JonMcLoone, *morlet.png* - *This W3C-unspecified diagram was created with Mathematica*, CC BY-SA 3.0. adres: <https://commons.wikimedia.org/w/index.php?curid=18678721>.
- [25] —, *mexicanhat.png* - *Own work This W3C-unspecified diagram was created with Mathematica*, CC BY-SA 3.0. adres: <https://commons.wikimedia.org/w/index.php?curid=18678721>.
- [26] ataspinar.com, *A guide for using the Wavelet Transform in Machine Learning*. adres: <http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/>.
- [27] Georg-Johann, *Linear-chirp.svg* - *This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported, 2.5 Generic, 2.0 Generic and 1.0 Generic license*.

