Onderzoeksrapport

Voorspellen van onderhoud aan rollagers met deep learning

Lectoraat Data Science – HZ University of Applied Sciences

Onderzoeksrapport v1.0 van Loek van der Linde
studentnummer 69724
in het kader van de opleiding HBO-ICT aan de
HZ University of Applied Sciences te Vlissingen

Onderzoeksrapport

Voorspellen van onderhoud aan rollagers met deep learning

Lectoraat Data Science – HZ University of Applied Sciences

Samenvatting

Er wordt de laatste jaren steeds vaker gesproken over predictive maintenance, het voorspellen van onderhoud aan apparatuur waarbij die voorspelling is gebaseerd op de staat van dat apparaat. Er wordt continu gemeten en pas ingegrepen als de metingen aangeven dat het fout gaat. Om deze metingen te kunnen analyseren, zijn goede modellen nodig. Apparatuur wordt complexer en meetsituaties chaotischer en de uitdaging om een goed model te maken voor deze complexe meetsystemen wordt dus ook steeds groter. Een manier van modelgeneratie die opgewassen lijkt tegen deze omstandigheden, is deep learning: het toepassen van diepe neurale netwerken.

Bij het lectoraat Data Science aan de HZ University of Applied Sciences wordt recent aandacht besteed aan predictive maintenance. Het lectoraat is daarbij benieuwd naar de toepassingen van deep learning. Men is benieuwd wat de concrete waarde kan zijn van deep learning bij het voorspellen van onderhoud. Om dit te onderzoeken, is de volgende vraag geformuleerd: *"Is het mogelijk om door deep learning toe te passen op vibratiedata beter te voorspellen wanneer een apparaat onderhoud nodig heeft dan met traditionele methoden?"*.

Vervolgens zijn met een dataset en de regels van de PHM IEEE 2012 Data Challenge drie methoden van deep learning onderzocht. De in dit onderzoek voorgestelde methode, getraind op gestandaardiseerde fourier transformaties van vibratiesignalen, scoort even goed als de winnaar van de PHM IEEE 2012 Data Challenge. Deze methode combineerde geavanceerde statistiek en mechanica, terwijl de methode uit dit onderzoek weinig tot geen domeinspecifieke kennis behoeft om toch succesvol gebruikt te worden. Daarmee lijkt de in dit onderzoek ontwikkelde methode een goed antwoord te geven op de hoofdvraag, en concrete kennis en tooling te geven aan het lectoraat Data Science.

Summary

In recent years, the subject of predictive maintenance is brought up with increasing frequency. Predictive maintenance is a maintenance strategy that looks at the current state of machinery to predict when said machinery would need maintenance. It has proven to be reducing both costs and the frequency of failures. To do this, machinery is constantly monitored by measuring all kinds of behavior like vibration or temperature. It's getting harder and harder to analyze these measurements however, since machinery is becoming more complex and the environments more chaotic. One way of analyzing this data, that seems robust even in these situations, is deep learning.

The applied research centre data science at the HZ University of Applied Sciences has recently started to look into the concrete application of predictive maintenance and the role deep learning can play in it. To see what deep learning can mean for predictive maintenance, the following question was formed: "*Based on vibration data, is it possible to predict more accurately when machinery needs maintenance than with traditional prediction methods?*".

For this research, the dataset and rules of the PHM IEEE 2012 Data Challenge were used and three methods of deep learning were researched: LSTM, fully connected and autoencoding. The fully connected network, trained on standardised fourier transforms performed the best and was on par with the original winner of the PHM IEEE 2012 Data Challenge. That method combined advanced statistics and mechanics, while the proposed method in this research required little to no knowledge about both these areas to be used effectively. For these reasons, it is believed the main research question was answered adequately and the applied research centre data science was provided with concrete knowledge and tooling.

Voorwoord

Zonder Mischa was dit onderzoek niet mogelijk geweest, en daar wil ik hem dan ook voor bedanken. Ik was weer eens koppig genoeg om een veel te academisch onderwerp uit te kiezen, en jij gaf me tegelijkertijd de vrijheid en de kaders om dat toch op een enigszins praktijkgerichte manier te doen. Dit was het afstudeertraject waarvan ik had gehoopt dat ik hem zou hebben, met bijna onbegrijpelijke materie en toffe collega's - die ik door een onverwachte samenloop van omstandigheden ook volgend jaar weer mijn collega's mag noemen. Ik ben afgelopen jaar veel te weten gekomen over deep learning en data science, en heb misschien nog wel meer vragen dan ervoor. Gelukkig heb ik een vak gekozen waarin deze onderwerpen een steeds grotere rol gaan spelen, dus waarschijnlijk ga ik ooit nog wel eens antwoord vinden op mijn vragen.

Ik wil ook Gert bedanken, die met zijn bijna bodemloze put aan ervaring en kennis over data science op werkelijk al mijn processen en modellen feedback had. Soms zakte de moed me een beetje in de schoenen. Had ik eindelijk een goed model, kwam jij weer langs met je "Ja. Heb je ook hier aan gedacht..?". Maar uiteindelijk heeft je feedback me geleid tot een serie modellen die stevig in hun schoenen staan. 100% kans dat vakexperts in machine learning nog veel verbeterpunten zien, maar het fundament van mijn modellen klopt, en daar ben ik trots op.

Daarnaast wil ik graag Juul even in het zonnetje zetten. Jij moest je scriptie net wat eerder inleveren dan ik, had overal net even een andere kijk op dan ik, en stiekem heb ik best een hoop goede ideëen van je gepikt. Ook wil ik Louisa en Max; Louisa met je altijd lachwekkende grappen, creatieve mindset en out-of-the-box ideëen die na even nadenken helemaal zo gek nog niet zijn. Max met je kennis over deep learning waarmee je mij een paar keer goed op weg geholpen hebt. Last but not least, Alex. Je altijd broodnuchtere kijk op het leven werkte lekker relativerend als we weer eens samen in onze stamkroeg zaten. Ook was je niet te beroerd om mijn scriptie van voor naar achter door te lezen en op alle verbeterpunten die je zag feedback te geven. Dit rapport is vast niet PhD waardig, maar we kunnen dan ook niet allemaal plasmafysicus worden...

Inhoud

1. Inleiding

1.1. Aanleiding

Op het lectoraat Data Science op de HZ University of Applied Sciences wordt sinds kort flink ingezet op predictive maintenance, een techniek waarmee "just in time" onderhoud aan apparaten is te voorspellen (Mobley, 2004). Predictive maintenance richt zich hierbij op de conditie van het te onderhouden apparaat, die gemeten wordt met verschillende sensoren. Uit de data die deze sensoren genereren, is in theorie te bepalen wat voor gedrag het gemeten apparaat vertoont (Girdhar, 2004). Zolang er geen abnormaal gedrag (cq. falen) plaatsvindt, is onderhoud onnodig en kunnen kosten worden bespaard.

Uit recent onderzoek in Nederland, België en Duitsland onder 280 bedrijven, blijkt dat zij bezig zijn om predictive maintenance te implementeren. In dit onderzoek werden vier niveaus vastgesteld:

1. Visuele inspectie,
2. Instrumentele inspectie,
3. Real-time monitoring,
4. Voorspellen van falen.

Het bleek dat maar 11% van de onderzochte bedrijven het hoogst gestelde niveau te behalen. Tweederde van de onderzochte bedrijven blijft steken op niveau een of twee: het planmatig inspecteren van apparatuur, eventueel aan de hand van instrumenten (Haarman, Mulders, & Vassiliadis, 2017). Dit niveau komt vrijwel overeen met preventive maintenance (Mobley, 2004). 89% van de onderzochte bedrijven zijn nog niet in staat falen aan apparatuur voorspellen en er is dus nog enorme winst te behalen voor deze bedrijven. Er kan bijvoorbeeld worden voorkomen dat een fabriek onverwacht stil moet worden gezet omwille van een falend apparaat in het productieproces. Het herstarten van een fabriek brengt grote kosten met zich mee en is onwenselijk. Als tijdig kan worden voorspeld wanneer een apparaat gaat falen, is onderhoud in te plannen en kunnen kosten worden bespaard.

Bedrijven uit de regio Zeeland zijn zeer geïnteresseerd in wat predictive maintenance voor hun kan betekenen. Het succesvol toepassen van predictive maintenance – en dat uiteindelijk doorvoeren naar een “best in time” versie – kan voor deze bedrijven, als DOW Chemical, enorme besparingen op jaarbasis betekenen.

1.2. Probleemstelling

Het lectoraat Data Science heeft ondervonden dat bedrijven in de regio Zeeland over het algemeen bekend zijn met predictive maintenance en dit tot op zekere hoogte ook geïmplementeerd hebben. Sinds zeer recent maken ontwikkelingen in hard- en software het echter mogelijk om ook technieken als deep learning toe te passen. Deep learning - ook wel bekend als diepe neurale netwerken - leent zich bij uitstek voor patroonherkenning en modelgeneratie (Schmidhuber, 2014) en lijkt daarmee een goed middel om in te zetten voor predictive maintenance. Apparaten in

met name de procesindustrie worden steeds complexer, met als gevolg dat het voorspellen van onderhoud niet goed meer in behapbare modellen te vatten is (Yam, Tse, Li, & Tu, 2001). Deep learning is in staat betrouwbare modellen te genereren uit ontelbare variabelen (LeCun, Bengio, & Hinton, 2015). Er is vraag naar de toepassingen van deep learning op vibratiedata omdat hier nog weinig over bekend is, waardoor de techniek momenteel niet toepasbaar is voor het lectoraat Data Science.

1.2.1. Doelstelling

Het uiteindelijke doel van dit onderzoek is om een neuraal netwerk te ontwerpen en te trainen aan de hand van vibratiedata verkregen uit trilsensoren rondom een apparaat. Dit netwerk moet na training vast kunnen stellen of een apparaat faalt als het nieuwe data gevoed krijgt. Het neurale netwerk stelt het lectoraat Data Science in staat bedrijven te helpen bij het voorspellen van onderhoud aan hun apparatuur; of op zijn minst te helpen bij het ontwikkelen van betere modellen.

1.2.2. Relevantie

Dit onderzoek is relevant voor het lectoraat Data Science omdat het resultaat van dit onderzoek de weg kan openen naar het inzetten van deep learning in haar bezigheden. Tevens worden er stappen gezet in de toepassingen van predictive maintenance; er kunnen immers voorspellingen worden gedaan. Daarnaast kan dit onderzoek van waarde zijn voor bedrijven die geïnteresseerd zijn in predictive maintenance en deep learning. De HZ University of Applied Sciences is een praktijkgerichte school, waarbij studenten samen met docenten en bedrijven kennis vergaren en creëeren die vervolgens in de regio Zeeland toegepast kan worden. Het lectoraat Data Science speelt hierbij een instrumentele rol. Dit onderzoek zorgt voor kennis over de concrete toepassingen van deep learning (bij predictive maintenance) bij het lectoraat Data Science. Bedrijven kunnen samen met het lectoraat, de HZ en haar studenten kijken hoe deze kennis in hun situaties toegepast kan worden. Tevens kan dit onderzoek los als handvat dienen voor bijvoorbeeld bedrijven uit het onderzoek van Haarman et al (2017); bedrijven die op niveau 1, 2 en 3 zitten, kunnen wellicht gebruik maken van (de resultaten van) dit onderzoek om zichzelf naar een hoger niveau van predictive maintenance te tillen.

Ten slotte is dit onderzoek van belang omdat het een schaars onderzoek is. Er is nog weinig onderzoek uitgevoerd op de toepassing van deep learning op vibratiedata; daarnaast geven de wel uitgevoerde onderzoeken een erg theoretisch beeld over de toepassingen. Een praktische oplossing is lastig te vinden. Dit onderzoek tracht in dit gat te springen en te dienen als startpunt voor meer praktijkgeorienteerd onderzoek.

1.3. Vraagstelling

Dit onderzoek wordt benaderd vanuit de hypothese dat het toepassen van deep learning op vibratiedata zorgt voor betere voorspellingsmodellen dan het toepassen van traditionele methoden. Om deze hypothese te onderzoeken en doelstelling te bewijzen, is de volgende onderzoeksvraag opgesteld:

"Is het mogelijk om door deep learning toe te passen op vibratiedata beter te voorspellen wanneer een apparaat onderhoud nodig heeft dan met traditionele methoden?"

Om deze vraag goed te kunnen beantwoorden, zijn de volgende deelvragen opgesteld:

- Welke deep learning methode is geschikt voor het voorspellen van machinaal falen aan de hand van de beschikbare data?
- Hoe presteert de gekozen methode?
- Hoe kan de methode worden uitgebreid of aangepast zodat er beter voorspeld kan worden?

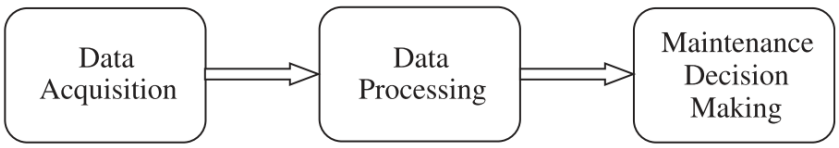
2. Theoretisch kader

In dit hoofdstuk wordt eerder uitgevoerd onderzoek aangehaald dat relevant is voor dit onderzoek. Eerst wordt gekeken naar wat predictive maintenance precies inhoudt, welke stappen nodig zijn en welke problemen daarin naar boven komen. Vervolgens wordt gekeken naar wat deep learning is en wat de mogelijkheden zijn. Tot slot wordt uitgelegd welke mogelijkheden deep learning kan bieden om de problemen die worden ondervonden in predictive maintenance op het gebied van vibratieanalyse op te lossen.

2.1. Predictive maintenance

Predictive maintenance, vaak ook wel *condition based maintenance* genoemd (Swanson, 2001) is zoals eerder genoemd een vorm van onderhoud waarbij specifiek naar de conditie van het te onderhouden apparaat wordt gekeken (Mobley, 2004). Naast predictive maintenance wordt ook vaak gesproken over *preventive maintenance*, gepland onderhoud, en *reactive maintenance* of breakdown maintenance, onderhoud dat plaatsvindt op het moment dat een apparaat gestopt is met werken (Jardine, Lin, & Banjevic, 2006). Het is voor organisaties wenselijk om predictive maintenance op een goede manier te implementeren; dit drukt kosten omdat er alleen mankracht in wordt gezet wanneer dat nodig is. Daarnaast zijn reserveonderdelen tijdig te bestellen en hoeven die niet per se op voorraad te hoeven worden gehouden (Martin, 1994). Om predictive maintenance op een goede manier te implementeren, zijn in ieder geval een drietal stappen nodig – zie ook Figuur 1 – (Jardine e.a., 2006):

1. datavergaring,
2. dataverwerking,
3. beslissing nemen.



Figuur 1: De drie benodigde stappen voor het succesvol implementeren van predictive maintenance

Soms wordt een vierde stap "evaluatie" toegevoegd in dit proces (Foxworth, 2017). Op basis van deze evaluatie-stap, wordt het proces waar nodig aangepast en opnieuw ingegaan. De bovenstaande drie stappen gebeuren achtereenvolgens, maar worden regelmatig doorlopen om altijd op recente resultaten te kunnen handelen (Jardine e.a., 2006).

2.2. Datavergaring

Zoals onderzocht door Haarman e.a. (2017) zijn veel bedrijven bezig om predictive maintenance te implementeren. Dit komt onder andere doordat het steeds makkelijker wordt om data te vergaren en te verwerken. Het zogenaamde Internet of Things maakt het bijvoorbeeld aanzienlijk makkelijker

en goedkoper om apparaten van krachtige sensoren te voorzien (Kiritsis, 2011). Waar vroeger relatief veel mankracht nodig was om apparatuur door te controlen – al dan niet met meetinstrumenten –, is het tegenwoordig mogelijk om betrouwbaar data te vergaren over bijvoorbeeld 4G-verbindingen.

2.3. Dataverwerking

Nadat data vergaard is, kan deze worden verwerkt en geanalyseerd. Hoe de data verwerkt wordt, ligt volledig open (Jardine e.a., 2006). Het is mogelijk relatief eenvoudige analyses uit te voeren door bijvoorbeeld te kijken hoe hard sensoren uitslaan. De resultaten uit deze analyse zijn te vergelijken met tabellen uit bijvoorbeeld ISO 20816 (International Organization for Standardization, 2016).
Vaak wordt een *discrete fourier transformatie* toegepast op vibratiedata, waarna de resultaten vergeleken worden met een baseline (Girdhar, 2004). Een fourier transformatie zet een signaal om in een zogenaamd frequentiespectrum; een grafiek waarin de amplitudes van alle losse frequenties van een signaal zijn uitgezet. Zo is het mogelijk alle frequenties uit een signaal te filteren en los te onderzoeken. Discrete fourier transformaties worden vaak gemaakt middels het *fast fourier transform* (FFT) algoritme. Een FFT, eventueel in combinatie met een *Power Spectral Density Diagram* (PSD), wordt gezien als een van de krachtigste instrumenten om fouten op te sporen in roterende en bewegende apparaten (Bond Randall, 2011). Ook kan een *time waveform analysis*, die specifieker geschikt is voor versnellingen en apparaten die met een zeer lage (rotatie)snelheid worden gebruikt (Girdhar, 2004).

2.4. Problemen met verwerking

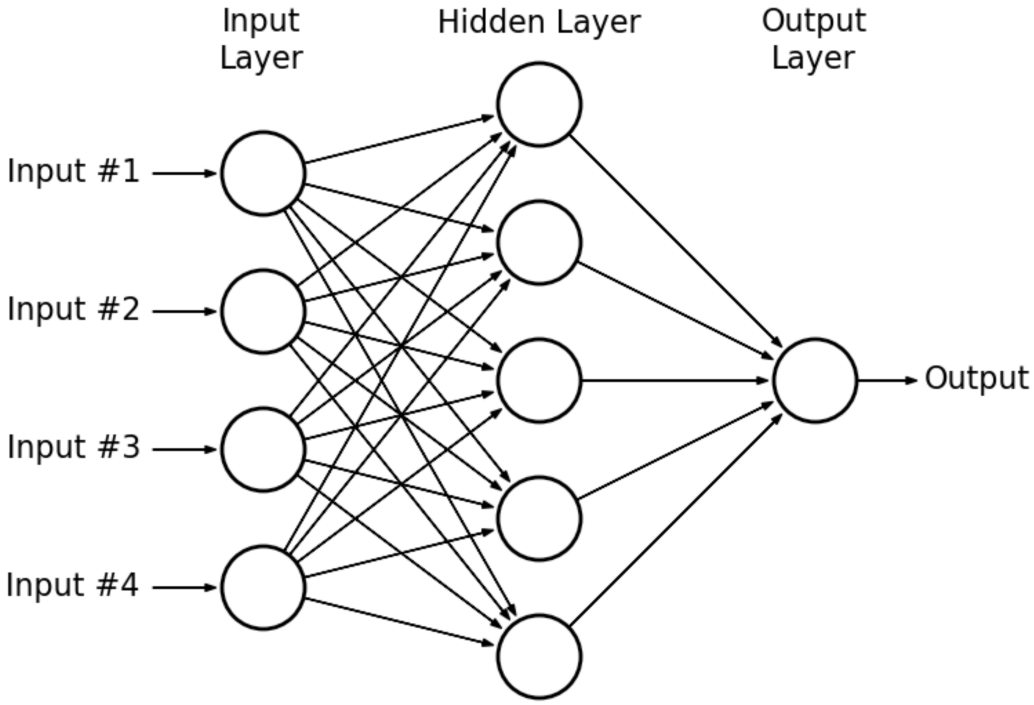
Een groot probleem met het verwerken van de vergaarde data is dat er vaak te veel data is om het proces handmatig te doorlopen. Een relatief eenvoudige machine bestaat al snel uit tientallen draaiende en bewegende onderdelen (Girdhar, 2004). Een grote fabriek omvat meer dan duizend van dit soort machines, met elk meerdere sensoren. Het is onmogelijk om al deze data met de hand te verwerken. In de jaren 1990 en 2000 is in veel industrieën een explosieve groei aan sensoren is ontstaan (Kadlec, Gabrys, & Strandt, 2009). De enorme hoeveelheden data die deze sensoren opleven, wordt ook wel *big data* genoemd. Kenmerkend aan big data is dat er simpelweg te veel data is om efficiënt – en in sommige gevallen überhaupt – te verwerken op "traditionele" manieren zoals baselinevergelijkingen (Hashem e.a., 2015). Bedrijven als Google en Microsoft proberen in dit gat te springen met nieuw ontwikkelde cloudplatformen waarop big data inzichtelijk te maken is.

Uit de genoemde en nog volgende literatuur blijkt dat het vrijwel onmogelijk is om voorspellingen te doen over onderhoud aan apparaten. Traditionele methoden zijn redelijk goed in het classificeren van fouten, maar zijn vaak arbeidsintensief, gevoelig voor ruis en vereisen specialistische kennis over de apparatuur die onderzocht wordt. Hoewel traditionele methoden nog altijd worden gebruikt, gaf Martin in 1994 al aan dat deze methoden niet meer krachtig genoeg waren om volledig op te vertrouwen. Zo waren Combet, Martin, Jaussaud, & Léonard in 2006 in staat om vanaf 160 uur voordat een machine zou falen te voorspellen dat er scheurtjes in de aandrijfas van de machine zaten, maar was het onmogelijk om tot op het moment van falen zelf aan te geven wanneer de machine zou falen. Daarbij moet vermeld worden dat dit onderzoek achteraf plaats vond en dus

bekend was naar welke fout gezocht moest worden.

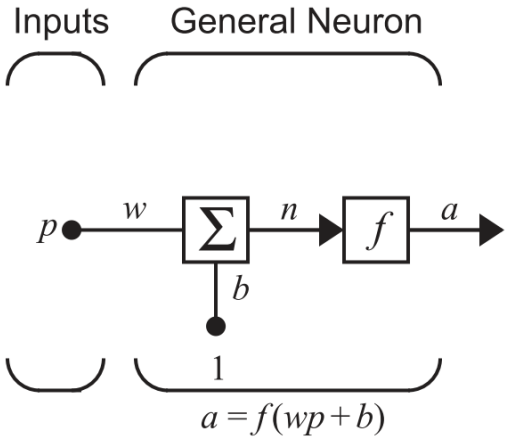
2.5. Neurale netwerken

Een relatief nieuwe manier om (big) data inzichtelijk te maken, is deep learning. Deep learning heeft recentelijk veel aandacht gekregen en is lastig te definiëren. Over het algemeen wordt er een verzameling technieken bedoeld, die het mogelijk maken machine learning te doen zonder dat de interne datastructuren van de ingevoerde data bekend zijn (LeCun e.a., 2015). Deep learning wordt vaak toegepast in samenwerking met neurale netwerken, algoritmes die sterk lijken op hoe neuronen in het menselijk lichaam werken. Biologische neuronen vormen een gigantisch netwerk – het zenuwstelsel – en geven signalen door aan volgende neuronen (Izhikevich, 2003). Als de actie "arm optillen" moet worden uitgevoerd, reist dat signaal door een specifiek pad van neuronen tot het bij de eindbestemming is en de arm daadwerkelijk wordt opgetild. Neurale netwerken volgen hetzelfde principe: de data uit de *input layer* reist door een specifiek pad langs een of meer *hidden layers* tot het de *output layer* heeft bereikt (Demuth, Beale, De Jesus, & Hagan, 2014). Verschillende inputs zorgen op die manier voor verschillende outputs – zie Figuur 2.



Figuur 2: Neuraal netwerk met 4 inputs, 1 hidden layer en 1 output

In Figuur 3 is een schematisch overzicht opgenomen van de werking van een neuron in een neuraal netwerk. Input *p* wordt doorgegeven aan het neuron, waar het allereerst vermenigvuldigd wordt met een zogenaamd *weight (w)*. Deze waarde wordt vervolgens opgeteld bij een statische waarde, de *bias (b)*. De uitkomst van deze som *(n)*, wordt in een zogenaamde transferfunctie *(f)* gestopt, waar uitkomst *a* uit komt. Voor deze transferfunctie wordt vaak een *Rectified Linear Unit* gebruikt (Schmidhuber, 2014). In dit voorbeeld wordt uitgegaan van een neuron met een enkele input. In de praktijk - zoals ook uit Figuur 2 blijkt - hebben neuronen vaak meerdere inputs. De beschreven berekening blijft in dit geval hetzelfde, maar wordt dan uitgevoerd op een matrix van *p* en *w* (Demuth e.a., 2014).



Figuur 3: Innerlijke werking van neuron

Neurale netwerken "leren" door eerst trainingsdata tot zich te nemen. Nadat er voldoende getraind is op deze set, kan er nieuwe data in het netwerk worden gestopt om het resultaat te bepalen. Neurale netwerken zijn op verschillende manieren te trainen, de twee bekendste manieren zijn *supervised learning* en *unsupervised learning (Schmidhuber, 2014)*. Tijdens supervised learninig wordt er getraind met een set data waarvan alle waarden bekend zijn. Wordt er bijvoorbeeld getraind op afbeeldingen van getallen herkennen, zoals bij NIST (2016), dan wordt bij elke afbeelding een label met de correcte waarde meegeleverd. Het neuraal netwerk kan zich bij elk nieuw stukje trainingsdata dus corrigeren. Dit gebeurt via een geautomatiseerd proces dat *backpropagation* wordt genoemd (Demuth e.a., 2014). Daarbij wordt vergeleken hoe goed het netwerk heeft gepresteerd ten opzichte van het meegeleverde label. Vervolgens worden de weights en biases van alle neuronen zo aangepast dat het netwerk beter wordt in het matchen van verwachte uitkomst en daadwerkelijke uitkomst. Bij unsupervised learning worden er geen labels met de trainingsdata meegeleverd en is de precisie van het netwerk dus slecht te bepalen. Dit soort netwerken zijn erg goed in het clusteren van data omdat er alleen naar de inputs gehandeld wordt (Demuth e.a., 2014).

Extreem vereenvoudigd bestaat het trainen van een neuraal netwerk uit zes stappen:

1. Presenteren van een meting met bijbehorende uitkomst;
2. Voor elk neuron de activatiefuncties berekenen;
3. De output bepalen uit de uitkomsten van de activatiefuncties;
4. De afwijking van de output met de verwachte output berekenen volgens een cost-functie;

5. De weights en biases van de neuronen aanpassen zodat de afwijking omlaag gaat;
6. Stap 1-5 herhalen totdat een gewenste afwijking is behaald.

Hieruit valt te herleiden dat een neuraal netwerk tijdens training in essentie niets anders doet dan een complexe functie met mogelijk miljoenen parameters benaderen. Het uiteindelijke model is dus een functie die voor een set inputs een bepaalde output genereert.

2.6. Deep learning

Neurale netwerken worden krachtiger naarmate meerdere hidden layers worden toegevoegd, lagen neuronen tussen de input layer en output layer. De term "deep learning" betekent in neurale netwerken dat er meerdere hidden layers worden gebruikt. Elke hidden layer is in staat een klein deelprobleem van het totale probleem op te lossen. Deep learning netwerken zijn hierdoor beter in staat om complexe problemen op te lossen dan conventionele neurale netwerken. Over het algemeen worden neurale netwerken met twee of drie hidden layers gebruikt (Demuth e.a., 2014), maar er bestaan modellen met meer dan honderd hidden layers. Schmidhuber (2015) beschrijft dat er tientallen verschillende deep learning netwerken bestaan, elk geschikt voor een specifieke oplossing. Een *convolutioneel neuraal netwerk* lijkt daarbij een grote gemene deler; dit neuraal netwerk scoort op de meeste taken goed. Het is echter niet vaak de efficiëntste oplossing. Zogenaamde *Long Short Term Memory Networks* (LSTM) zijn erg goed in het voorspellen van data over lange tijdseenheden (Ma, Tao, Wang, Yu, & Wang, 2015).

2.7. Vibratieanalyse met deep learning

Onlangs is begonnen met het onderzoeken van de toepassingen van deep learning op vibratiedata. Opvallend is dat onderzoek in deze richting veel tractie heeft. De onderzoeken die zijn te vinden, zijn vaak theoretisch van aard en zeer recent gepubliceerd. De komende alinea dient daarom vooral ter inspiratie en toont daarnaast de noodzaak aan voor praktijkgericht onderzoek in deze richting.

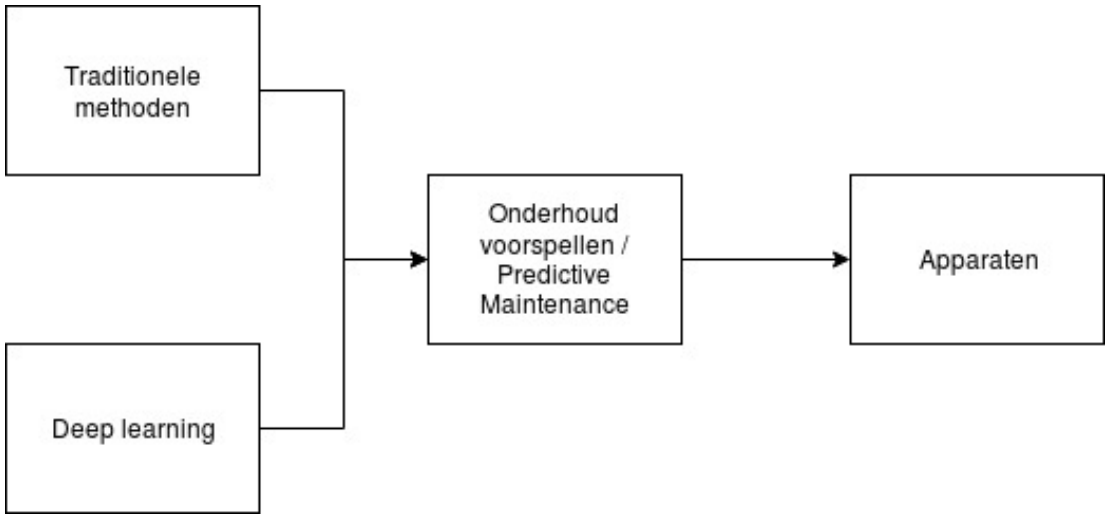
Een onderzoek vergeleek traditionele manieren van foutdetectie met een neuraal netwerk dat classificeerde of een getijdenturbine faalde of niet. Klassieke foutdetectie haalde een precisie van 99.83%, terwijl het neuraal netwerk 100% scoorde. Daarbij was het neuraal netwerk getraind met een dataset van slechts "rauwe" vibratiedata zonder ondersteunende data als rotorsnelheid of waterdruk (Galloway, Catterson, Fay, Robb, & Love, 2015). Uit een ander onderzoek blijkt dat het combineren van verschillende deep learning technieken in een groot netwerk kan leiden tot hoge efficiëntie van foutdetectie in rollagers. Het netwerk was in staat om grote hoeveelheden ruis weg te filteren en was gemiddeld 99% nauwkeurig in het classificeren van de locatie waar een rollager faalde (Gan, Wang, & Zhu, 2016). Tot slot blijkt uit het onderzoek van Li, Liu, Tang, Lu, & Hu (2017) dat convolutionele neurale netwerken wel degelijk in te zetten zijn voor tijdserieanalyses, mits de data goed geprepareerd wordt. Ook blijkt dat verschillende modellen erg wisselend presteren als er getest wordt onder andere omstandigheden dan waarop getraind is. Dit toont de noodzaak van representatieve trainingsdata aan.

2.8. Conclusie

Predictive maintenance en deep learning zijn beide vakgebieden waarin veel onderzoek wordt gedaan en die recentelijk meer en meer aandacht krijgen door verschillende baanbrekende onderzoeken en ontwikkelingen. Predictive maintenance wordt steeds lastiger om uit te voeren door de grote groei in data en steeds complexer wordende machines. Deze twee factoren zorgen ervoor dat traditionele methoden moeilijker inzetbaar zijn en daarbij ook minder accurate resultaten geven. Deep learning lijkt een goed hulpmiddel om in te zetten voor predictive maintenance. Er is nog niet veel bekend over de toepassingen van deep learning op vibratiedata. Er is daarentegen wel veel bekend over foutanalyse op basis van vibratiedata en de werking van deep learning. Deze informatie kan als goede basis dienen voor dit onderzoek, waarin de eerste stappen naar een praktijkoplossing gezet worden.

2.9. Conceptueel model

In Figuur 4 is in schematisch overzicht het conceptuele model van dit onderzoek weergegeven. De focus op dit onderzoek ligt daarbij op kijken of deep learning als betere input geldt voor het bedrijven van predictive maintenance dan huidig gebruikte traditionele methoden.



Figuur 4: Conceptueel model onderzoek

3. Methode

3.1. Onderzoeksstrategie

Een onderzoek kan kwantitatief of kwalitatief van aard zijn (Baarda, 2014). In een kwantitatief onderzoek staat de onderzoeksvraag vast en worden vooraf vastgelegde ideeën getoetst. Uit de vergaarde gegevens worden doorgaans modellen en grafieken als resultaat gepresenteerd. Dit onderzoek is van kwantitatieve aard; het doel is immers om te toetsen of deep learning betere resultaten produceert dan traditionele methoden.

Er is gebruikt van een openbare dataset van het FEMTO-ST instituut in Frankrijk. Deze dataset is opgesteld ten behoeve van de "IEEE PHM 2012 Data Challenge" (Nectoux e.a., 2012). De dataset beschrijft het gedrag van een rollager onder verschillende omwentelsnelheid en torsie. Voor de challenge is een trainingsset van 6 rollagers beschikbaar, elk met metingen van de temperatuur en van de horizontale en verticale uitslag van de lager. Daarnaast is een testset van 11 rollagers beschikbaar, die na een willekeurig aantal metingen zijn afgekapt. Doel van de challenge was om te voorspellen wat de *remaining use of life* voor deze rollagers was: hoe lang duurt het voordat elke rollager faalt (Gouriveau e.a., 2012). Er waren winnaars in twee categorieeën: industrie en academisch. Respectievelijk haalden deze winnaars scores van 0.28 (Porotsky & Bluvband, 2012) en 0.30 (Sutrisno, Oh, Vasan, & Pecht, 2012). In dit onderzoek worden de temperatuurmetingen buiten beschouwing gelaten en alleen gebruik gemaakt van de vibratiemetingen.

Een voordeel van deze dataset is dat de experimenten zijn afgekapt op het moment dat een rollager faalde. Dit fenomeen zal in de praktijk zelden tot nooit voorkomen, omdat het daadwerkelijk laten falen van een rollager een te groot risico is voor het bedrijf in kwestie. Het gaf de te trainen neurale netwerken echter een concreet en onomstotelijk punt om op te voorspellen. Daarnaast is de dataset uitvoerig behandeld in de wetenschappelijke literatuur, en er dus een betrouwbare benchmark is voor hoe goed traditionele methoden presteren in het voorspellen van onderhoud.

3.2. Deelvraag 1 – Welke deep learning methode is geschikt voor het voorspellen van machinaal falen aan de hand van de beschikbare data?

Deze vraag is beantwoord middels een combinatie van literatuuronderzoek en brainstormsessies. Deze combinatie heeft uiteindelijk tot een "pakketselectie" voor een dataverwerkingsomgeving geleid; met de omgeving was deelvraag 2 te beantwoorden.

Om een keuze te kunnen maken voor een geschikte deep learning methode, moest de onderzoeker eerst bekend krijgen aan welke eisen het uiteindelijk op te leveren neuraal netwerk moest voldoen. Een deel van deze eisen (requirements) was te verkrijgen door (al deels beschreven) literatuuronderzoek. Om de eisen sluitend en realistisch te krijgen, werden gesprekken met het lectoraat Data Science, de lector Asset management van de HZ en een docent aan de opleiding HBO-ICT van de HZ gevoerd. Deze gesprekken vonden plaats in de vorm van een brainstormsessie, omdat de onderzoeker vooral benieuwd was naar informatie die hij nog niet had

gehoord of gelezen. Uit eerdere ervaringen van het lectoraat Data Science bleek dat stakeholders uit de regio niet altijd genoeg inzicht of begrip hebben om gestelde vragen te beantwoorden. Om deze reden is gekozen voor een brainstormsessie in plaats van een interview. De verwachting was dat een brainstormsessie meer zou uitnodigen tot meedenken en dus tot nieuwe inzichten zou leiden.

Aan de hand van de verkregen requirements *is* een zoekplan opgesteld. Met dit zoekplan werd gezocht naar geschikte deep learning methoden. Uit de gevonden long list van methoden is met vastgestelde knock-out criteria voor een definitieve methode gekozen,

Er is gekozen voor de combinatie literatuuronderzoek en brainstormsessie omdat op deze manier op gestructureerde wijze bronnen worden doorzocht en ideeën worden gegenereerd. Wat was daadwerkelijk mogelijk en onmogelijk op het gebied van deep learning? De brainstormsessie met het lectoraat Data Science en andere stakeholders gaven vervolgens de aanknooppunten waarop een methode gekozen kan gaan worden. Bijvoorbeeld hoe ver van tevoren er voorspeld moet worden dat een apparaat kapot gaat. Hierdoor kon een gefundeerde keuze voor methode gemaakt worden.

3.3. Deelvraag 2 – Hoe presteert de gekozen methode?

Deze vraag is beantwoord middels een experiment, namelijk het uitvoeren van de gekozen voorspellingsmethode op de beschikbare data.

De training- en testdata is uitgezet in frequentiespectra met behulp van fast fourier transformaties en gestandaardiseerd voor een gemiddelde van 0 en standaarddeviatie van 1.

Elke losse meting in de dataset is omgezet tot een frequentiespectrum. Zo zijn bijvoorbeeld voor rollager 1_1 2803 frequentiespectra voor de horizontale uitslag en 2803 frequentiespectra voor de verticale uitslag gemaakt. Deze frequentiespectra zijn per meting samengevoegd. Een losse meting bevatte uiteindelijk 5120 datapunten om op te trainen.

Vervolgens zijn de verwerkte metingen opgedeeld in reeksen van 150 metingen, met een overlap van 149 metingen. Elk stukje trainingsdata schoof dus een meting op, totdat de laatste meting werd bereikt. De biasen van de LSTM neuronen werden geinitialiseerd op 1, zoals beschreven door Jozefowicz, Zaremba, & Sutskever (2015). Verder zijn de weights willekeurig geïnitialiseerd.

Hoe goed het netwerk trainde, werd berekend met een cost-functie. Hiervoor is de mean squared error gebruikt; de gemiddelde gekwadrateerde afwijking van de voorspelling ten opzichte van de waarheid. De uitkomt van deze cost-functie is de waarde waarop daadwerkelijk getraind gaat worden. Hoe dichter bij nul, hoe beter het netwerk presteert; de voorspelling klopt dan namelijk steeds beter. De uitkomst van de cost-functie is ook wel bekend als *loss.*

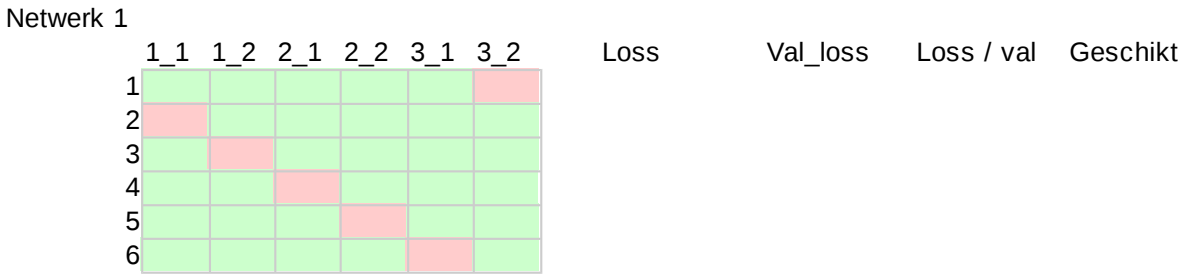
Naast het aanpassen van de weights en biases van alle neuronen, heeft een neuraal netwerk ook globale parameters. Deze parameters bepalen de trade-off tussen snelheid en precisie tijdens training. De globale parameters van het netwerk worden tijdens training aangepast door een zogenaamde *optimizer*. Het aanpassen van de globale parameters in combinatie met de cost-functie zorgt voor optimale training. In dit onderzoek is de optimizer *Adam* gebruikt. Alle netwerken zijn voor 500 *epochs* getraind, waarbij early stopping werd toegepast. Dit houdt in dat de loss voor de validatiedata binnen 3 epochs moet dalen, anders werd de training stopgezet. Tijdens een epoch ziet het netwerk alle trainingsdata precies een keer.

Lager	Metingen	FFTs
1_1	2803	5606
1_2	870	1740
2_1	911	1822
2_2	797	1594
3_1	515	1030
3_2	1637	3274

Netwerk	Layers	Neuronen
1	1	250
2	1	500
3	1	1000
4	2	250
5	2	500
6	2	1000

Figuur 5: Meta-informatie data en cross-validatie voor FEMTO set

Om tot de beste oplossing te komen, is aan cross-validatie gedaan met 6 verschillende netwerkarchitecturen, beschreven in Figuur 5 en Figuur 6. Elk netwerk is getraind op 5 lagers (groen) en gevalideerd op 1 lager (rood). De validatiestap wordt door de optimizer gebruikt om de juiste globale parameters voor het gekozen netwerk uit te zoeken.



Figuur 6: Template voor resultaten cross-validatie voor FEMTO set

Na cross-validatie is per netwerk gekeken wat de verhouding loss : validatieloss is. Deze moest tussen 0.8 en 1.2 liggen, anders was het netwerk *overfit*. Het netwerk was dan zo goed getraind op het voorspellen van de trainingsdata, dat het geen goede voorspelling gaf voor nieuwe data. De oplossing van het netwerk was dan heel specifiek en niet generiek bruikbaar.

Als de berekende loss en validatieloss allebei minder dan 20.000 waren, werd het netwerk getest door een voorspelling uit te voeren op de niet-geziene testdata. Deze voorspelling bestond uit het gemiddelde van de voorspelling die op basis van de laatste 10 metingen gedaan worden. Deze voorspellingen zeiden iets over de daadwerkelijke voorspellingskracht van de netwerken. De netwerken hebben de testdata immers nooit gezien. De afwijking van deze voorspelling (en daarmee de prestaties van de netwerken) werd bepaald volgens scoreformules gegeven in de PHM Challenge (Gouriveau e.a., 2012). Zie tevens Formule 1, Formule 2 en Formule 3. De eindscore was een getal tussen 0 en 1, waarbij 1 de maximale score is en alle voorspellingen op de testdata precies goed waren.

$$\%Er_i = \frac{100 * ActRUL_i - PredRUL_i}{ActRUL_i}$$

Formule 1: Bepalen procent-afwijking van voorspelling.

$$if\ Er_i > 0 : A_i = e^{-\ln(0.5)*(Er_i/5)}$$
$$if\ Er_i \leq 0 : A_i = e^{-\ln(0.5)*(Er_i/20)}$$

Formule 2: Scorefunctie per lager aan de hand van eerder bepaalde afwijking. Late voorspellingen scoren slechter dan vroege voorspellingen. Zie tevens Bijlage A - Scorefunctie

$$Score = \frac{1}{11} \sum_{i=1}^{11} A_i$$

Formule 3: Bepalen score netwerk: gemiddelde van de 11 eerder bepaalde scores per lager.

3.4. Deelvraag 3 – Hoe kan de methode worden uitgebreid of aangepast zodat er beter voorspeld kan worden?

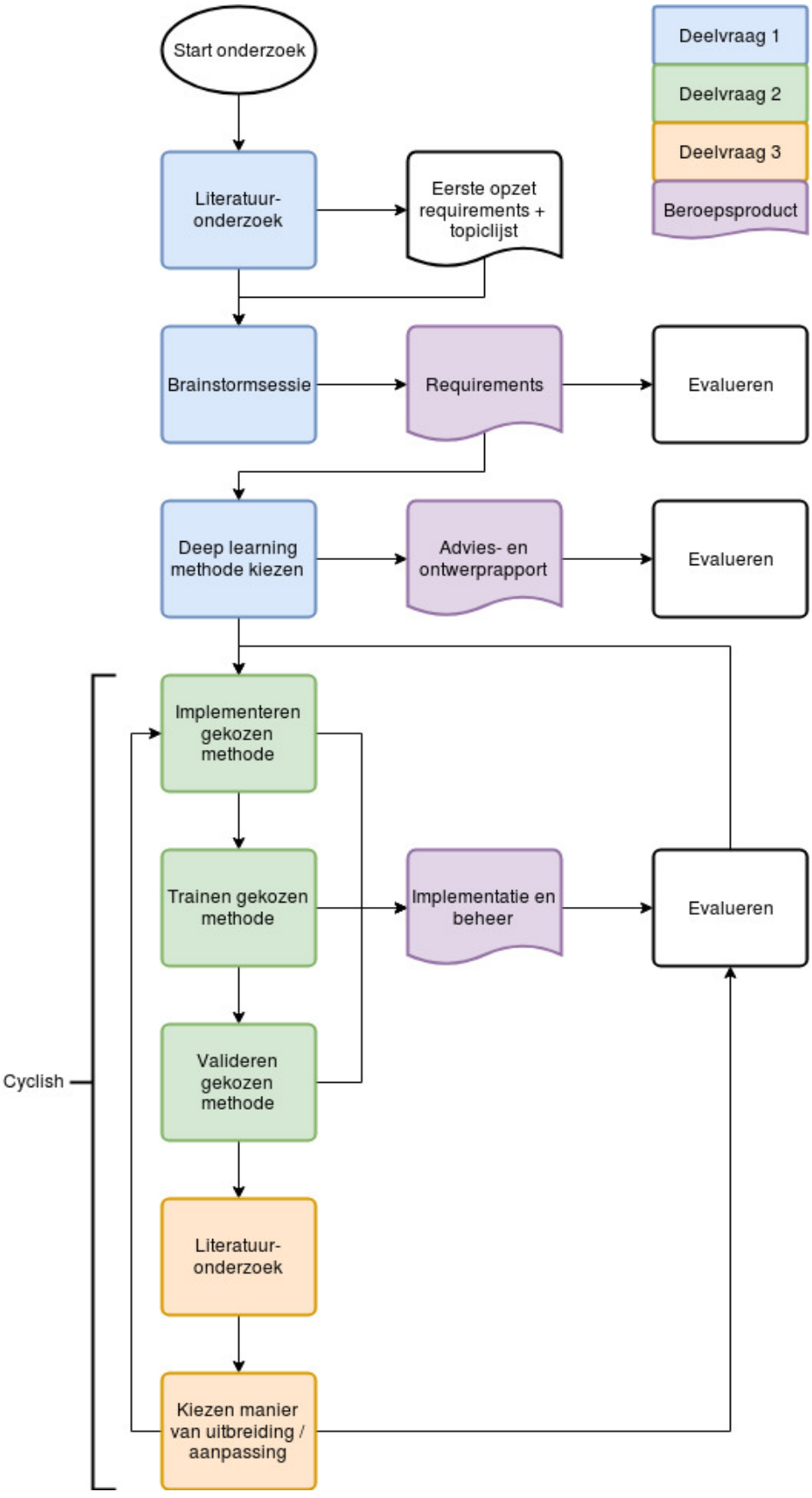
Deze vraag is beantwoord middels een combinatie van literatuuronderzoek en experimentatie.

Deze deelvraag probeerde te beantwoorden of het mogelijk was de opgezette deep learning methode zo aan te passen is, dat de resultaten beter werden.

Om deelvraag 3 te kunnen beantwoorden, was een zoekplan opgesteld. Met dit zoekplan werd gezocht worden in literatuur hoe de gekozen methode was uit te breiden of aan te passen zodat deze deelvraag te beantwoorden was. Uit de gevonden manieren is wederom een long list opgesteld. Met nader te bepalen knock-out criteria werd gekozen voor een definitieve methode. Vanaf dit moment werd het onderzoek cyclisch vanaf deelvraag 2 opnieuw doorlopen, zie Figuur 7. Dit om een zo goed mogelijk resultaat te kunnen behalen. Het was daarbij mogelijk dat eerder gevonden antwoorden op deze deelvraag slecht (of niet) bleken te scoren, waardoor voor een andere methode van aanpassing gekozen moest worden.

Er is gekozen voor literatuuronderzoek omdat op die manier op gestructureerde wijze bronnen doorzocht worden. Wat zijn hier de beste manieren voor en hoe zijn die te implementeren?

3.5. Totaalbeeld



Figuur 7: Totaalbeeld van onderzoeksmethoden en -strategie

4. Resultaten

4.1. Deelvraag 1 - Welke deep learning methode is geschikt voor het voorspellen van machinaal falen aan de hand van de beschikbare data?

Het eerste dat opviel in gelezen literatuur, is dat eigenlijk elk onderzoek in ieder geval gebruik maakt van een klassiek fully connected neural network. Vaak is dit om te benchmarken hoe goed een andere methode werkt, echter worden soms de beste resultaten ook gehaald met een fully connected network (Mahamad, Saon, & Hiyama, 2010). Fully connected networks zijn al beschreven in de jaren 50 en vormen de de facto standaard waar andere vormen van neurale netwerken tegen worden getest.

Een zo'n vorm is een convolutioneel neuraal netwerk, een methode van deep learning die speciaal is ontwikkeld voor het classificeren van afbeeldingen (Schmidhuber, 2014). Convolutionele netwerken schuiven als het ware een vergrootglas over een afbeelding, waarbij de afbeelding dus wordt opgedeeld in tientallen, honderden of zelfs duizenden kleine afbeeldingen. Vervolgens kunnen uit deze kleine afbeeldingen features worden geleerd, die later in het netwerk weer samengevoegd worden zodat de afbeelding betrouwbaar geclassificeerd kan worden. Het lijkt mogelijk om convolutionele netwerken in te zetten bij vibratiemetingen, door gebruik te maken van spectogrammen (Shaheryar, Yin, & Ramay, 2017). Een spectogram is in essentie namelijk een pixelmap, en dus bruikbaar in een convolutioneel netwerk.

Daarnaast bestaan er nog speciale netwerken ten behoeve van tijdserieanalyse, deze vallen in de categorie recurrent neuraal netwerk (Schmidhuber, 2014). De eenvoudigste vorm hiervan is een fully connected recurrent netwerk, wat in essentie een fully connected netwerk is waarbij de neuronen in hidden en output layers hun output terug sturen als input naar eerder layers. Een groot nadeel van fully connected recurrent netwerken is dat ze steeds slechter leren naarmate de netwerken meer layers krijgen, ook wel het *vanishing gradient problem* genoemd (Gamboa, 2017). Om dit probleem op te lossen, is het Long Short-Term Memory (LSTM) netwerk ontwikkeld (Gamboa, 2017). In deze netwerken worden de neuronen vervangen door cellen met een aantal in- en outputs die ook onderling verbonden zijn. Met een iets gewijzigde formule voor het berekenen van weights en biases, wordt gezorgd dat de vanishing gradient vastgezet wordt in een enkele cel en zich dus niet door het netwerk kan verspreiden. Eerder is aangetoond dat LSTMs succesvol gebruikt kunnen worden voor het voorspellen van het tijdstip van falen van complexe straalmotoren (Wu, Yuan, Dong, Lin, & Liu, 2018).

Uit de brainstormsessies met drie stakeholders bleek dat het vrijwel onmogelijk is om concreet eisen te stellen aan een voorspellingsalgoritme. Elk apparaat is verschillend en heeft een verschillende functie, wat het per definitie onmogelijk maakt om aan te geven wat de laatst acceptable voorspelling is en hoe accuraat die moet zijn. Daarnaast heeft elke manier van falen van elk apparaat ook weer zijn eigen parameters. Ook spelen bedrijfsprocessen een grote rol. Lector Asset Management noemde als voorbeeld Rijkswaterstaat. Zij wil 18 maanden van tevoren weten of asfalt gerepareerd moet worden, omdat het simpelweg 18 maanden duurt voordat alle vergunningen

en contracten ten behoeve van de reparatie zijn geregeld.

Meerdere respondenten noemden daarnaast de PF-curve. Dit is een eenvoudig model waarin een gezondheidsindicatie en een voorspelling van tijdstip van functioneel falen van een apparaat worden gegeven. Uit het literatuuronderzoek en de brainstormsessies zijn een aantal eisen aan het model - en zelfs praktische softwaretoepassing - naar voren gekomen, zie Bijlage B - Requirements.

Om tot een methode van deep learning te komen, zijn op basis van de eigenschappen van de data en technische eisen aan het model knock-out criteria vastgesteld zoals weergegeven in Figuur 8. Hiernaast zijn per uigelichte methode van deep learning de scores aangegeven. FCN = fully connected netwerk, CNN = convolutioneel neuraal netwerk, FCRN = fully connected recurrent netwerk, LSTM = Long Short-Term Memory netwerk.

	FCN	CNN	FCRN	LSTM
In staat fourier-transformatie als input te gebruiken				
In staat zelf features te herkennen				
In staat 5120 inputs te verwerken				
In staat 1 output te genereren				
In staat om te gaan met signaalruis				
Voorspellen op basis van tijdserie mogelijk				
Verliest geen herkende features bij gebruik meerdere layers				
In staat afhankelijkheden over lange termijn te volgen				
Kan inputs van <0 en >1 verwerken				
Compatibel met Keras				
Compatibel met Tensorflow				
Bruikbaar met Mean Square Error als costfunctie				
Bruikbaar met Adam als optimizer				

Mogelijk
Niet mogelijk
Strict gezien mogelijk, maar vereist heftige ingreep op data of model

Figuur 8: Knock-out criteria en scores per deep learning methode

Op basis van de scores op deze knock-out criteria is een short list ontstaan van:

- Fully connected netwerk
- Fully connected recurrent netwerk
- LSTM

4.2. Deelvraag 2 - Hoe presteert de gekozen methode?

De resultaten van de LSTM cross-validatie zijn te zien in Figuur 10.

Geen enkele LSTM configuratie haalt een loss en validatieloss van minder dan 20000. De beste kandidaat lijkt netwerk 5.6. Dit netwerk heeft een loss : validatieloss ratio van 1 en heeft de laagste gecombineerde loss en validatieloss.

Trachten te voorspellen met dit netwerk geeft altijd de waarde 47.96572, welke zorgt voor een totaalscore van 0.036. Zie Figuur 9, Formule 1, Formule 2 en Formule 3.

Layers: 2
Neurons: 500

Lager	Set	Voorspelling	Verwacht	% Eri	Ai	Score
1	3	47.96572	5730	99.16	0.03	0.036
1	4	47.96572	339	85.85	0.05	
1	5	47.96572	1610	97.02	0.03	
1	6	47.96572	1460	96.71	0.04	
1	7	47.96572	7570	99.37	0.03	
2	3	47.96572	7530	99.36	0.03	
2	4	47.96572	1390	96.55	0.04	
2	5	47.96572	3090	98.45	0.03	
2	6	47.96572	1290	96.28	0.04	
2	7	47.96572	580	91.73	0.04	
3	3	47.96572	820	94.15	0.04	
		`	Avg % Eri	95.8762		

Figuur 9: Score beste LSTM-netwerk

Ophogen van het aantal neuronen of aantal layers zorgt ervoor dat de computer waarmee getest wordt uit geheugen loopt. Hierdoor was het onmogelijk om binnen de gestelde tijd onderzoek te doen naar grotere netwerkarchitecturen. Huidige training van een netwerk kost ongeveer drie uur op een computer met een Nvidia GTX 1050 Ti met 4gb VRAM. Een ronde cross-validatie duurt dus bijna drie werkdagen.

Netwerk 1

	1_1	1_2	2_1	2_2	3_1	3_2	Loss	Val_loss	Loss / val	Geschikt
1							547,844	341,415	1.60	nee
2							178,425	909,956	0.20	nee
3							206,252	113,403	1.82	nee
4							205,763	130,589	1.58	nee
5							254,407	45,237	5.62	nee
6							192,072	141,075	1.36	nee

Netwerk 2

	1_1	1_2	2_1	2_2	3_1	3_2	Loss	Val_loss		
1							610,129	391,373	1.56	nee
2							189,543	978,640	0.19	nee
3							207,528	118,579	1.75	nee
4							208,973	127,672	1.64	nee
5							213,513	106,444	2.01	nee
6							179,993	201,334	0.89	ja

Netwerk 3

	1_1	1_2	2_1	2_2	3_1	3_2	Loss	Val_loss		
1							602,266	405,311	1.49	nee
2							190,746	900,169	0.21	nee
3							208,026	128,681	1.62	nee
4							209,212	133,351	1.57	nee
5							213,335	118,675	1.80	nee
6							182,893	197,086	0.93	ja

Netwerk 4

	1_1	1_2	2_1	2_2	3_1	3_2	Loss	Val_loss		
1							665,951	250,857	2.65	nee
2							189,941	1,076,895	0.18	nee
3							206,307	103,974	1.98	nee
4							208,233	122,279	1.70	nee
5							253,986	46,376	5.48	nee
6							191,935	128,891	1.49	nee

Netwerk 5

	1_1	1_2	2_1	2_2	3_1	3_2	Loss	Val_loss		
1							614,235	383,775	1.60	nee
2							189,720	979,762	0.19	nee
3							207,483	119,147	1.74	nee
4							208,976	127,528	1.64	nee
5							213,501	106,492	2.00	nee
6							180,026	180,914	1.00	ja

Netwerk 6

	1_1	1_2	2_1	2_2	3_1	3_2	Loss	Val_loss		
1							604,219	400,269	1.51	nee
2							191,145	904,267	0.21	nee
3							207,978	128,642	1.62	nee
4							209,160	132,575	1.58	nee
5							213,590	116,420	1.83	nee
6							182,828	197,169	0.93	ja

Figuur 10: Resultaten cross-validatie LSTM

4.3. Deelvraag 3 – Hoe kan de methode worden uitgebreid of aangepast zodat er beter voorspeld kan worden?

Uit de onderzochte literatuur bleek dat er ontzettend veel opties zijn om de voorspellingen van het netwerk te verbeteren. Er zijn twee veelbelovende opties gekozen om uit te werken: een klassiek fully connected netwerk en het gebruik van autoencoding.

4.3.1. Fully Connected Netwerk

In bijna alle gelezen literatuur is het gangbaar dat een complexe netwerkarchitectuur zoals eerder uitgewerkte LSTM-netwerk wordt vergeleken met een *fully connected* netwerk (FCN). Bij het beantwoorden van deelvraag 1 bleek een FCN goed geschikt te zijn, ondanks dat er op losse metingen getraind en voorspeld zou moeten worden

Er is gekozen om 6 netwerken te trainen, zie Figuur 11. De geleverde trainingsset van FEMTO is random gesplitst in trainingsdata en validatiedata, volgens een ratio van 80 : 20. Om een FCN te trainen, is het niet nodig de data om te zetten in sequences; er wordt op elk los frequentiespectrum getraind. Hierna zijn volgens de LSTM-methode alle losse metingen wederom omgezet in frequentiespectra en gestandaardiseerd voor een gemiddelde van 0 en standaarddeviatie van 1.

De netwerken zijn voor 2500 epochs getraind, met dezelfde early stopping regels als bij de LSTM-methode. Wederom werd mean square error gebruikt als cost-functie, en Adam als optimizer.

Netwerk	Layers	Neuronen
1	8	500
2	16	500
3	8	1250
4	16	1250
5	8	2500
6	16	2500

Figuur 11: Verschillende getrainde fully connected netwerken

Hieruit volgden de resultaten in Figuur 12. De netwerken hebben allemaal een loss van vele malen lagen dan de validatieloss. Alle netwerken zijn echter gebruikt om op de testdata te voorspellen, omdat de validatieloss over kleiner dan 20.000 was. Nadat alle netwerken zijn geëvalueerd op de testdata, bleek dat netwerk 2 het best presteert met een score van 0.30, zie Figuur 13. Training duurde gemiddeld een uur.

Netwerk	Loss	Val_loss	score
1	25	13610	0.13
2	144	11277	0.3
3	65	12138	0.14
4	192	9899	0.2
5	176	13978	0.15
6	4828	19628	0.13

Figuur 12: Resultaten na training fully connected netwerken

Lager	Set	Voorspelling	Verwacht	% Eri	Ai	Score
1	3	15471	5730	-170.00	0.00	0.3004
1	4	155	339	54.28	0.15	
1	5	1542	1610	4.22	0.86	
1	6	1365	1460	6.51	0.80	
1	7	869	7570	88.52	0.05	
2	3	5379	7530	28.57	0.37	
2	4	3508	1390	-152.37	0.00	
2	5	2046	3090	33.79	0.31	
2	6	1189	1290	7.83	0.76	
2	7	4040	580	-596.55	0.00	
3	3	1388	820	-69.27	0.00	
			Avg % Eri	-69.4986		

Figuur 13: Voorspellingen fully connected netwerk met 16 hidden layers van 500 neuronen.

4.3.2. Autoencoding

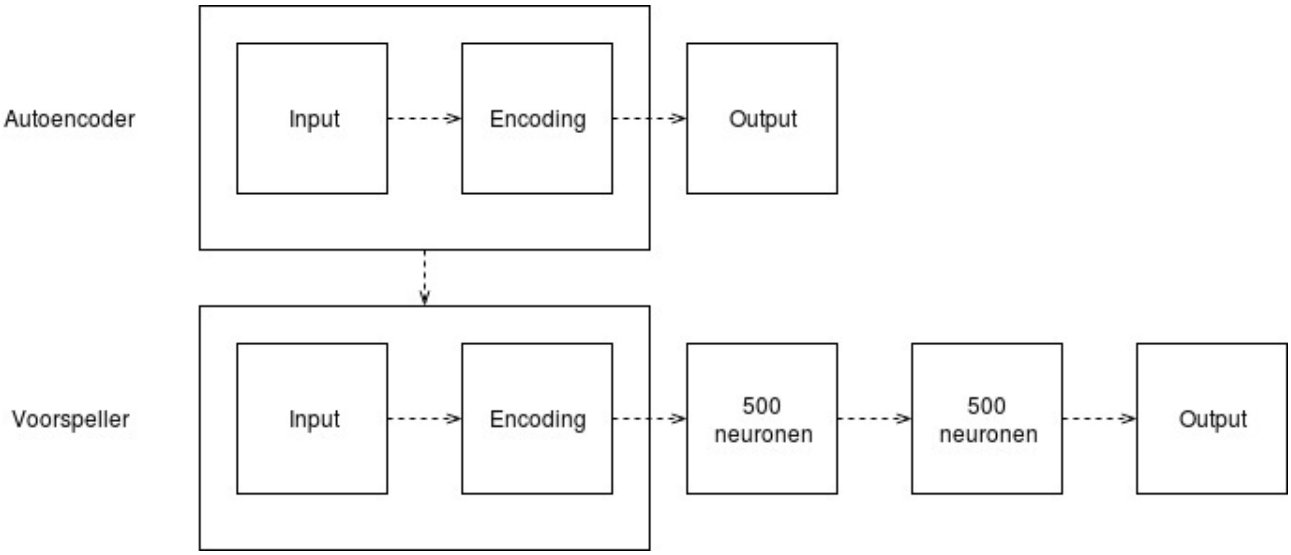
Een andere manier van verbetering, is het inzetten van zogenaamde *autoencoders* (Hinton & Salakhutdinov, 2006). Autoencoders zijn neuronen die een gecomprimeerde versie van de input leren. Autoencoding is een vorm van unsupervised learning, waarbij de input van het netwerk gelijk is aan de output. Men is daarbij vooral geïnteresseerd in de hidden layers, in essentie de compressielagen. Als het aantal hidden layer en de globale parameters van het netwerk goed worden gekozen, leren de autoencoders precies die features die belangrijk zijn voor het reconstrueren van het originele signaal. Daarmee is autoencoding dus een vorm van *Principal Component Analysis,* en zijn de gekozen features in theorie te gebruiken als foutindicatoren.

Lv, Duan, Kang, Li, & Wang toonden in 2015 aan dat een autoencoder in combinatie met een regressielaag zeer accuraat kan voorspellen op welke tijden er veel verkeer langs onderzochte straten rijdt. Eergenoemde Sun e.a. (2016), Liu e.a. (2016) en Galloway e.a. (2015) pasten autoencoders toe op vibratiedata en waren in staat met hoge precisie foutdiagnoses te stellen. Ook Lu, Wang, Qin, & Ma, (2017) was hiertoe in staat, hoewel in dat onderzoek gebruik werd gemaakt van een stacked autoencoder.

Er is gekozen om een autoencoder van 500 neuronen te trainen, gezien dit aantal goed presteerde bij de FCN-methode. De geleverde trainingsset van FEMTO is random gesplitst in trainingsdata en validatiedata, volgens een ratio van 80 : 20. Het was het niet nodig de data om te zetten in sequences; er wordt op elk los frequentiespectrum getraind. Hierna zijn volgens de LSTM-methode alle losse metingen wederom omgezet in frequentiespectra en gestandaardiseerd voor een gemiddelde van 0 en standaarddeviatie van 1. Als laatste stap zijn alle frequentiespectra genormaliseerd voor waarden tussen 0 en 1. Dit maakte het gebruik van de efficiënte *binary crossentropy* cost-functie mogelijk. Ook voor deze functie geldt dat hoe dichter bij nul, hoe beter het resultaat. Als optimizer is wederom Adam gebruikt. Na training van 100 epochs convergeerde de autoencoder met een loss van 0.1606 en validatieloss van 0.1616. Omdat het in unsupervised learning moeilijk is het succes van een getraind netwerk te valideren, zijn drie willekeurige frequentiespectra handmatig gevalideerd, zie Bijlage C - Autoencoding. Uit deze validatie bleek dat

de autoencoder behoorlijk in staat was het originele frequentiespectrum te reconstrueren.

Vervolgens zijn de eerste twee lagen, weights en biases van de getrainde autoencoder - de input- en encodinglaag - overgenomen in een neuraal netwerk met twee hidden layers van 500 neuronen; zie Figuur 14. Het volledige netwerk is hierna op een supervised manier getraind, ook wel *fine tuning* genoem.



Figuur 14: Gebruik van getrainde autoencoder in fully connected netwerk

Er is geëxperimenteerd met het aantal hidden layers achter de encodinglaag, waarbij twee layers het beste resultaat gaven met een score van 0.22, zie Figuur 15. Training van de autoencoder duurde ongeveer tien minuten, het trainen van het uiteindelijke netwerk ongeveer vijftien minuten.

Lager	Set	Voorspelling	Verwacht	% Eri	Ai	Score
1	3	10710	5730	-86.91	0.00	0.2173
1	4	1836	339	-441.59	0.00	
1	5	2364	1610	-46.83	0.00	
1	6	1482	1460	-1.51	0.81	
1	7	902	7570	88.08	0.05	
2	3	4370	7530	41.97	0.23	
2	4	3170	1390	-128.06	0.00	
2	5	2011	3090	34.92	0.30	
2	6	1558	1290	-20.78	0.06	
2	7	4066	580	-601.03	0.00	
3	3	806	820	1.71	0.94	
			Avg % Eri	-105.458		

Figuur 15: Voorspelligen beste autoencoding netwerk met autoencoder van 500 neuronen en twee hidden layers met tevens 500 neuronen.

5. Conclusies

In dit hoofdstuk worden de resultaten uit vorig hoofdstuk in context gezet en geïnterpreteerd. Wat betekenen deze resultaten precies? Eerst wordt op elke deelvraag een antwoord gegeven, waarna er afgesloten wordt met een antwoord op de hoofdvraag.

5.1. Deelvraag 1 - Welke deep learning methode is geschikt voor het voorspellen van machinaal falen aan de hand van de beschikbare data?

De LSTM-methode is de best geschikte methode, blijkt uit de vastgestelde knock-out criteria. Opvallend is dat het convolutioneel neuraal netwerk aan weinig van de knock-out criteria voldoet, terwijl deze methode bij het literatuuronderzoek in het theoretisch kader naar boven kwam als een over het algemeen goede methode voor welke taak dan ook. Het fully connected recurrent netwerk scoort goed, maar heeft twee grote gebreken. Dit zijn niet toevallig precies de gebreken die een LSTM oplost. Het klassieke fully connected netwerk scoort ook goed, maar kan slechts op enkele metingen voorspellingen doen. Omdat de verwachting is dat er afhankelijkheden op de lange termijn in de data zitten, is gekozen deze methode niet te gebruiken. Als laatst moet ook genoemd worden dat het bijna onmogelijk is om concrete en sluitende eisen aan een voorspellingsmethode te stellen. Apparaten en hun gebruikssituaties zijn simpelweg te verschillend om dit te kunnen doen. Het is wel mogelijk voorspellingen in modellen als een PF-curve te hangen, welke onderhoudsexperts vervolgens kunnen raadplegen om gegronde beslissingen te maken.

5.2. Deelvraag 2 - Hoe presteert de gekozen methode?

Uit de score volgens de regels van de PHM IEEE Data Challenge blijkt dat de gekozen LSTM-methode slecht presteert: de score is vrijwel nul. Dit blijkt uit het feit dat er constant een zelfde waarde wordt voorspeld, los van welke meting er gebruikt wordt. Daarnaast scoort de methode niet alleen slecht volgens de PHM IEEE Data Challenge, de methode is ook onbruikbaar in de praktijk. Als voor elke meting dezelfde waarde wordt voorspeld, is deze waarde niet te vertrouwen. De rollager kan immers over een minuut kapot gaan, of misschien nog drie maanden zonder problemen werken. Waarschijnlijk voorspelt de methode altijd dezelfde waarde omdat tijdens de trainingsfase een gemiddelde is aangeleerd, waarvoor de cost-functie zo laag mogelijk is.

5.3. Deelvraag 3 - – Hoe kan de methode worden uitgebreid of aangepast zodat er beter voorspeld kan worden?

Uit de twee gekozen opties voor verbetering, blijkt dat het LSTM-netwerk toch het best vervangen kon worden door een fully connected netwerk met 16 hidden layers van 500 neuronen. Met dit netwerk werd een eindscore van 0.3 behaald. Daarnaast werd het tijdstip van falen voor drie van de elf lagers binnen tien procent afwijking voorspeld.

Aan dit fully connected netwerk is een aanpassing gedaan door de eerste layer van het netwerk te vervangen door een autoencoder. Hierop konden vervolgens 13 layers weggehaald worden, zonder

dat er een groot verschil in prestatie te merken was. Het best getrainde autoencodingnetwerk scoorde over de gehele set iets minder met 0.22, maar haalde wel twee voorspellingen met minder dan twee procent afwijking.

Beide netwerken trainen daarnaast aanzienlijk sneller dan eerder uitgewerkte LSTM-methode. Het autoencodingnetwerk kost het minste tijd; het is mogelijk binnen een half uur een volledige training te doorlopen van zowel autoencoder als fine tuning van het fully connected netwerk.

Op dit moment lijkt het fully connected netwerk de beste keuze, gezien de bijna 50% betere score waardevoller is dan de snellere trainingstijd en lagere complexiteit van het autoencodingnetwerk.

5.4. Conclusie

De hoofdvraag van dit onderzoek luidde:

"Is het mogelijk om door deep learning toe te passen op vibratiedata beter te voorspellen wanneer een apparaat onderhoud nodig heeft dan met traditionele methoden?".

Een in dit onderzoek ontwikkelde deep learning methode scoort even goed als de winnaars van de PHM IEEE 2012 Data Challenge. Het idee achter deze competitie was dat het met traditionele methoden als baseline- en spectogramanalyse vrijwel onmogelijk was om te voorspellen op de geleverde dataset.

Deelnemers aan de PHM IEEE 2012 Data Challenge werden dus geacht innovatieve en/of nieuwe methoden verzinnen om toch tot acceptabele voorspellingen te komen. De winnaars van de competitie behaalden scores van 0.30 en 0.28 uit een maximumscore van 1.

Een in dit onderzoek ontworpen diep neuraal netwerk komt tot een score van 0.30. Hiermee scoort dit netwerk even goed als de eerste winnaar van de competitie, die een drietal nieuwe methodes ontwikkelden waarbij geavanceerde statistiek met mechanica werd gecombineerd.

De PHM IEEE 2012 Data Challenge is inmiddels zes jaar geleden uitgeschreven. Toch lijkt de in dit onderzoek ontwikkelde methode goed antwoord te geven op de hoofdvraag. Zonder enige kennis van de werking van een rollager en met minimale voorbewerking van de metingen, is het mogelijk om even goed te scoren in de competitie als experts op het vakgebied. Dit toont aan dat deep learning het niveau van vakexperts benaderd zonder enige informatie over de context van de data waaraan het voorspelt.

6. Aanbevelingen

In dit hoofdstuk zullen voor- en nadelen van dit onderzoek worden besproken. Er zullen aanbevelingen worden gedaan waar nodig.

6.1. Vergelijken traditionele methoden

Allereerst wordt aanbevolen om dit onderzoek te vergelijken met scores verkregen uit traditionele methodes van voorspelling. Het is onbekend of dit zal leiden tot bruikbare resultaten, omdat de dataset specifiek is gemaakt om het onmogelijk te maken met traditionele methoden. Daarnaast, zoals aangegeven in het theoretisch kaders van dit onderzoek, is het per definitie al bijna onmogelijk goed te voorspellen met traditionele methoden als baseline- en spectogramanalyse.

6.2. Beter voorspellen

Hoewel de hoofdvraag positief is beantwoord, is er veel ruimte voor betere voorspellingen. Zo zit het beste netwerk er een keer 600% naast (en is deze nog te laat ook), twee keer ruim 150% (wederom te laat) en twee bijna 100%. Dit zijn voorspellingen waar weinig waarde aan gehecht kan worden, ze wijken simpelweg te veel af. In een praktijksituatie zou een model met zoveel variantie waarschijnlijk niet gebruikt worden. Het is beter om altijd 20% te vroeg te voorspellen, dan de ene keer perfect en de andere keer met 600% afwijking. Als de afwijking consistent was geweest, was daarvoor te corrigeren. Dat is nu niet het geval. Hoewel er dus goed wordt gescoord in de competitie, blijft te betwisten hoe goed het ontwikkelde model daadwerkelijk presteert in de werkelijkheid.

De makkelijkste manier voor verbetering van de voorspellingen is om meer vibratiedata te gebruiken voor de training. Dit onderzoek heeft zich aan de regels van de PHM IEEE Data Challenge gehouden, waarbij op zes lagers is getraind en op elf is getest. Er zijn echter datasets van zeventien lagers beschikbaar. Een aanbeveling is om te kijken hoe goed er gepresteerd wordt als er een aantal testsets worden toegevoegd aan de trainingset. Zo zou er op bijvoorbeeld twaalf lagers getraind en op vijf lagers getest kunnen worden. Ook zou de methode op andere datasets of zelfs een combinatie van getest kunnen worden.

Daarnaast zijn er in de gelezen literatuur een aantal manieren gevonden om tot betere resultaten te komen. Om deze manieren goed in te zetten, is meer domeinspecifieke kennis over zowel deep learning als mechanica een vereiste.

De meest voor de hand liggende manier is handmatige feature extraction, zoals beschreven in onderzoeken van Guo, Li, Jia, Lei, & Lin (2017), Ben Ali, Fnaiech, Saidi, Chebel-Morello, & Fnaiech (2015) en Lei e.a. (2018). De dataset van de PHM IEEE Data Challenge is hiervoor wellicht niet geschikt (Lei e.a., 2018), omdat de resolutie van de fourier transformaties te laag is. Voor goede feature extraction bij rollagers is een resolutie van 1 of 2 Hz nodig, in de huidige dataset is deze 10 Hz. Er kunnen wel andere features gebruikt worden, zoals de root mean square van het signaal, de kurtosis of bijvoorbeeld Weibull parameters zoals beschreven door Mahamad e.a. (2010).

Er zijn ook nog manieren die specifiek voor deep learning gelden, zoals dropout. Bij dropout worden tijdens de trainingsfase van een netwerk willekeurige neuronen uit gezet. Hierdoor zou het netwerk een betere generieke oplossing moeten leren (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Zoals te zien in de resultaten van deelvraag 3, vindt er enorme overfitting plaats. Het netwerk presteert bijna perfect op de trainingsdata, maar heeft duidelijk meer moeite met de validatie- en testdata. Een generiekere oplossing zou mogelijk tot betere resultaten kunnen leiden.

Daarnaast zijn er binnen autoencoding nog een aantal slimme manieren om tot betere resultaten te komen. Er kan gebruik gemaakt worden van zogenaamde *sparsity constraints*, waarbij de neuronen van een netwerk na training gemiddeld een activatie hebben van bijna 0. Dit forceert neuronen alleen te vuren bij inputs die echt belangrijk zijn, en is dus een vorm van automatische feature extraction (Ng, 2011). Liu e.a. (2016 en Sun e.a. (2016) waren met deze vorm van autoencoding in staat zeer accurate foutdiagnoses te geven op verschillende vibratiesignalen.

Daarnaast kan ook gebruik gemaakt worden van een stacked autoencoder, een autoencoder met meerdere encodinglagen. Hier geldt hetzelfde principe als bij deep learning, meer lagen zorgen in theorie voor betere patroonherkenning en dus betere resultaten. Lu e.a. (2017) maakten gebruik van een stacked autoencoder en waren daarbij zelfs goed in staat om ruis uit het signaal te filteren. Daarnaast kan er ook een autoencoder ingezet worden bij een LSTM-netwerk.

Het wordt aanbevolen om deze vier manieren van verbetering - en combinates daarvan - verder te onderzoeken om te kijken of deze tot verbetering van resultaten leiden. De verwachting op basis van gelezen literatuur is dat een combinatie van sparse autoencoding en dropout tot goede resultaten kan leiden.

6.3. Voorspellen onderhoud

Er wordt sterk aanbevolen om onderzoek te doen naar mogelijkheden tot foutclassificatie in combinatie met faalvoorspelling. Het meerendeel van de beschreven literatuur in dit onderzoek, heeft onderzoek gedaan naar mogelijkheden voor betrouwbare foutdetectie en -classificatie. Het fundament is aanwezig, maar moet zeker onderzocht en uitgewerkt worden. Een combinatie van deze twee methoden kan voor gerichter onderhoud zorgen.

6.4. Realistische data

Als laatst wordt de noodzaak voor het toepassen op realistische data aangekaart. Hoezeer de data voor de PHM IEEE Data Challenge ook is opgezet om voorspellen lastig te maken, zit hem dat vooral in de technische details. Er is te weinig data, de resolutie is te laag en de metingen verschillen flink in duur, maar het is relatief gezien nog altijd een zeer bruikbare dataset. Het moment van falen is vastgezet op een duidelijk moment, er is geen enkele missende meting en de omstandigheden van meten zijn bijna perfect. Er is heel specifiek met twee sensoren gemeten op alleen de rollagers. In de "echte wereld" is dit niet per se het geval en kan dit voor een serieus probleem in voorspellingen zorgen. Het kan best mogelijk zijn dat er apparatuur in de omgeving staat die voor zoveel ruis zorgt, dat de metingen van het te voorspellen apparaat onbruikbaar zijn. Daarnaast staat apparatuur niet altijd aan, wordt het in de tussentijd alsnog vervangen of

onderhouden en kunnen sensoren kapot gaan. Dit zorgt allemaal voor data die representatief is voor de werkelijkheid, maar tegelijk moeilijk te gebruiken is voor voorspellingen. Een van de belangrijkste vragen bij het beantwoorden van deelvraag 1 van dit onderzoek, was wanneer een apparaat nu eigenlijk faalt. In de gebruikte dataset is dit makkelijk, namelijk als de metingen ophouden. In de realiteit is dit echter een lastig vast te stellen moment waar veel domeinspecifieke kennis bij komt kijken.

Het wordt dus sterk aanbevolen om te onderzoeken hoe de ontwikkelde methode presteert op data en eisen verkregen uit de "echte wereld".

7. Bibliografie

Baarda, B. (Dirk B. (2014). *Dit is onderzoek! : handleiding voor kwantitatief en kwalitatief onderzoek* (Tweede). Noordhoff Uitgevers.

Ben Ali, J., Fnaiech, N., Saidi, L., Chebel-Morello, B., & Fnaiech, F. (2015). Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics*, *89*, 16–27. https://doi.org/10.1016/J.APACOUST.2014.08.016

Bond Randall, R. (2011). *Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications* (First). John Wiley & Sons, Ltd.

Combet, F., Martin, N., Jaussaud, P., & Léonard, F. (2006). Detection of gear coupling misalignment in a gear testing device. In *Eleventh International Congress on Sound and Vibration, ICSV11*. St Petersburg, Russia.

Demuth, H. B., Beale, M. H., De Jesus, O., & Hagan, M. T. (2014). *Neural Network Design* (Second). Martin Hagan.

Foxworth, T. (2017). Using IoT and Machine Learning for Industrial Predictive Maintenance | Losant Enterprise IoT Platform. Geraadpleegd 23 januari 2018, van https://www.losant.com/blog/using-iot-and-machine-learning-for-industrial-predictive-maintenance

Galloway, G. S., Catterson, V. M., Fay, T., Robb, A., & Love, C. (2015). Diagnosis of Tidal Turbine Vibration Data through Deep Neural Networks, 1–9.

Gamboa, J. C. B. (2017). Deep Learning for Time-Series Analysis. In *Seminar on Collaborative Intelligence*.

Gan, M., Wang, C., & Zhu, C. (2016). Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mechanical Systems and Signal Processing*, *72–73*, 92–104. https://doi.org/10.1016/J.YMSSP.2015.11.014

Girdhar, P. (2004). *Machinery Vibration Analysis & Predictive Maintenance*. (C. Scheffer & S. Mackay, Red.) (First). Oxford: Elsevier.

Gouriveau, R., Javed, K., Medjaher, K., Mosallam, A., Nectoux, P., Ramasso, E., & Zerhouni, N. (2012). IEEE PHM 2012 Data challenge. IEEE.

Guo, L., Li, N., Jia, F., Lei, Y., & Lin, J. (2017). A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, *240*, 98–109. https://doi.org/10.1016/j.neucom.2017.02.045

Haarman, M., Mulders, M., & Vassiliadis, C. (2017). *Predictive Maintenance 4.0: Predict the*

unpredictable.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Ullah Khan, S. (2015). The rise of “big data” on cloud computing: Review and open research issues. *Information Systems*, *47*, 98–115. https://doi.org/10.1016/J.IS.2014.07.006

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, *313*(5786), 504–7. https://doi.org/10.1126/science.1127647

International Organization for Standardization. (2016). ISO 20816-1:2016(en), Mechanical vibration — Measurement and evaluation of machine vibration — Part 1: General guidelines. Geraadpleegd 24 januari 2018, van https://www.iso.org/obp/ui/#iso:std:iso:20816:-1:ed-1:v1:en

Izhikevich, E. M. (2003). Simple Model of Spiking Neurons. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, *14*(6). https://doi.org/10.1109/TNN.2003.820440

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. https://doi.org/10.1016/J.YMSSP.2005.09.012

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. *Journal of Machine Learning Research*, *2015*.

Kadlec, P., Gabrys, B., & Strandt, S. (2009). Data-driven Soft Sensors in the process industry. *Computers & Chemical Engineering*, *33*(4), 795–814. https://doi.org/10.1016/J.COMPCHEMENG.2008.12.012

Kiritsis, D. (2011). Closed-loop PLM for intelligent products in the era of the Internet of things. *Computer-Aided Design*, *43*(5), 479–501. https://doi.org/10.1016/J.CAD.2010.03.002

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834. https://doi.org/10.1016/j.ymssp.2017.11.016

Li, S., Liu, G., Tang, X., Lu, J., & Hu, J. (2017). An Ensemble Deep Convolutional Neural Network Model with Improved D-S Evidence Fusion for Bearing Fault Diagnosis. *Sensors*, *17*(8), 1729. https://doi.org/10.3390/s17081729

Liu, H., Li, L., & Ma, J. (2016). Rolling Bearing Fault Diagnosis Based on STFT-Deep Learning and Sound Signals. *Shock and Vibration*, *2016*, 1–12. https://doi.org/10.1155/2016/6127479

Lu, C., Wang, Z.-Y., Qin, W.-L., & Ma, J. (2017). Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Processing*, *130*, 377–388. https://doi.org/10.1016/J.SIGPRO.2016.07.028

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, *16*(2). https://doi.org/10.1109/TITS.2014.2345663

Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, *54*, 187–197. https://doi.org/10.1016/J.TRC.2015.03.014

Mahamad, A. K., Saon, S., & Hiyama, T. (2010). Predicting remaining useful life of rotating machinery based artificial neural network. *Computers & Mathematics with Applications*, *60*(4), 1078–1087. https://doi.org/10.1016/J.CAMWA.2010.03.065

Martin, K. F. (1994). A review by discussion of condition monitoring and fault diagnosis in machine tools. *International Journal of Machine Tools and Manufacture*, *34*(4), 527–551. https://doi.org/10.1016/0890-6955(94)90083-3

Mobley, R. K. (2004). *An introduction to predictive maintenance* (Second). Woburn: Elsevier.

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., … Varnier, C. (2012). PRONOSTIA : An experimental platform for bearings accelerated degradation tests. In *IEEE International Conference on Prognostics and Health Management* (pp. 1–8). Denver.

Ng, A. (2011). CS294A Lecture notes Sparse autoencoder. Stanford: Stanford University.

NIST. (2016). NIST Special Database 19 | NIST. Geraadpleegd 25 januari 2018, van https://www.nist.gov/srd/nist-special-database-19

Porotsky, S., & Bluvband, Z. (2012). Remaining useful life estimation for systems with non-trendability behaviour. *2012 IEEE Conference on Prognostics and Health Management*, *2*. https://doi.org/10.1109/ICPHM.2012.6299544

Schmidhuber, J. (2014). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Shaheryar, A., Yin, X.-C., & Ramay, W. Y. (2017). Robust Feature Extraction on Vibration Data under Deep-Learning Framework: An Application for Fault Identification in Rotary Machines. *International Journal of Computer Applications*, *167*(4), 975–8887.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Sun, W., Shao, S., Zhao, R., Yan, R., Zhang, X., & Chen, X. (2016). A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement*, *89*, 171–178. https://doi.org/10.1016/J.MEASUREMENT.2016.04.007

Sutrisno, E., Oh, H., Vasan, A. S. S., & Pecht, M. (2012). Estimation of remaining useful life of ball bearings using data driven methodologies. *2012 IEEE Conference on Prognostics and Health Management*, *2*, 1–7. https://doi.org/10.1109/ICPHM.2012.6299548

Swanson, L. (2001). Linking maintenance strategies to performance. *International Journal of Production Economics*, *70*(3), 237–244. https://doi.org/10.1016/S0925-5273(00)00067-0

Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, *275*, 167–179. https://doi.org/10.1016/J.NEUCOM.2017.05.063

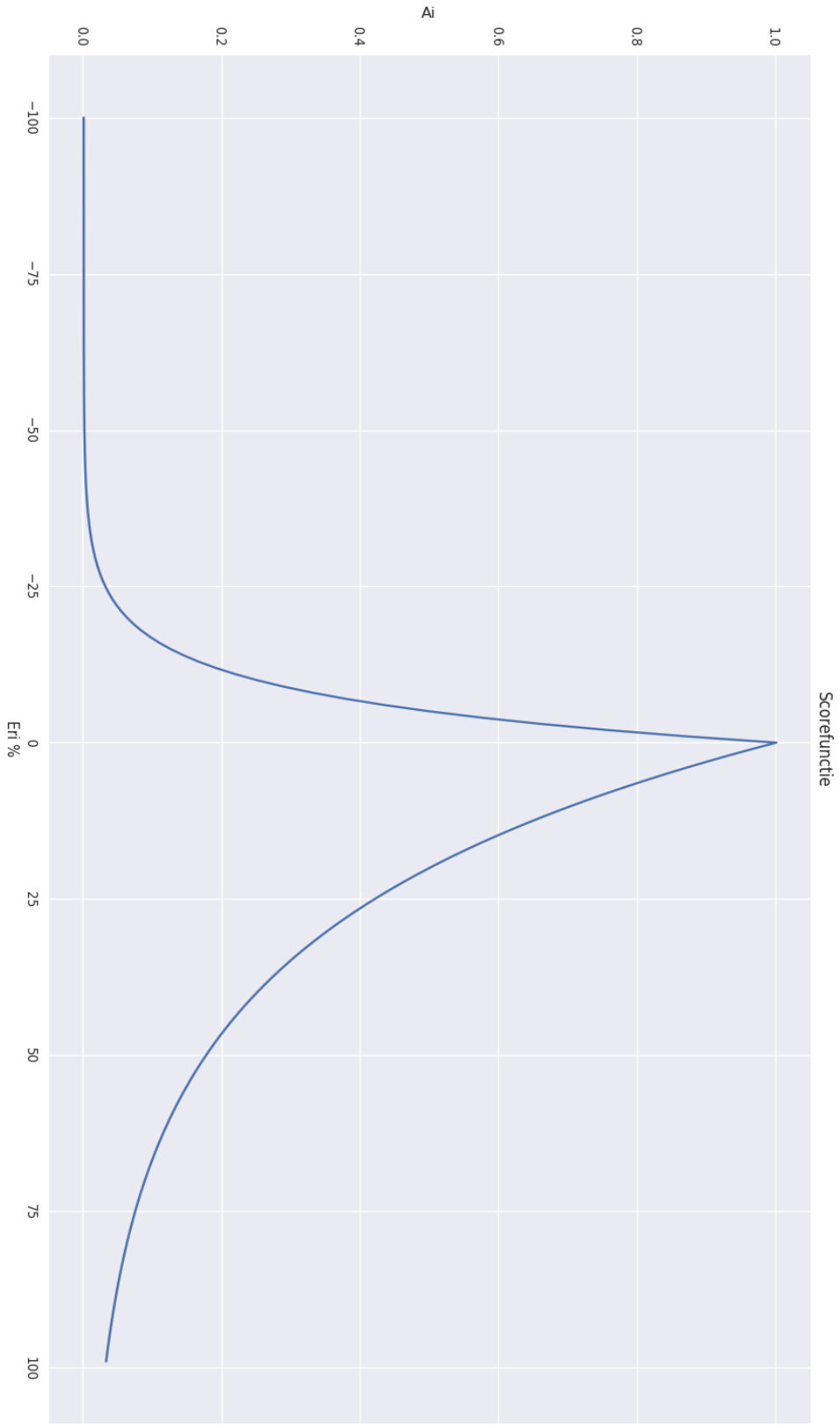
Yam, R. C. M., Tse, P. W., Li, L., & Tu, P. (2001). Intelligent Predictive Decision Support System for Condition-Based Maintenance. *Int J Adv Manuf Technol*, *17*, 383–391.

8. Figurentabel

Figuur	Beschrijving	Bron
1	De drie benodigde stappen voor het succesvol implementeren van predictive maintenance	Jardine e.a., 2006, pp. 1484
2	Innerlijke werking van neuron	Demuth e.a., 2014, pp. 38 NB: paginanummering in boek komt overeen met pp. 2-3
3	Neuraal netwerk met 6 inputs, 1 hidden layer en 4 outputs	http://neuralnetworksanddeeplearning.com/chap5.html
4	Conceptueel model onderzoek	L.W. van der Linde
5	Meta-informatie data en cross-validatie voor FEMTO set	L.W. van der Linde
6	Template voor resultaten cross-validatie voor FEMTO set	L.W. van der Linde
7	Totaalbeeld van onderzoeksmethoden en -strategie	L.W. van der Linde
8	Knock-out criteria en scores per deep learning methode	L.W. van der Linde
9	Score beste LSTM-netwerk	L.W. van der Linde
10	Resultaten cross-validatie LSTM	L.W. van der Linde
11	Verschillende getrainde fully connected netwerken	L.W. van der Linde
12	Resultaten na training fully connected netwerken	L.W. van der Linde
13	Voorspellingen fully connected netwerk met 16 hidden layers van 500 neuronen.	L.W. van der Linde
14	Gebruik van getrainde autoencoder in fully connected netwerk	L.W. van der Linde
15	Voorspellingen beste autoencoding netwerk met autoencoder van 500 neuronen en twee hidden layers met tevens 500 neuronen.	L.W. van der Linde

9. Bijlagen

10. Bijlage A - Scorefunctie



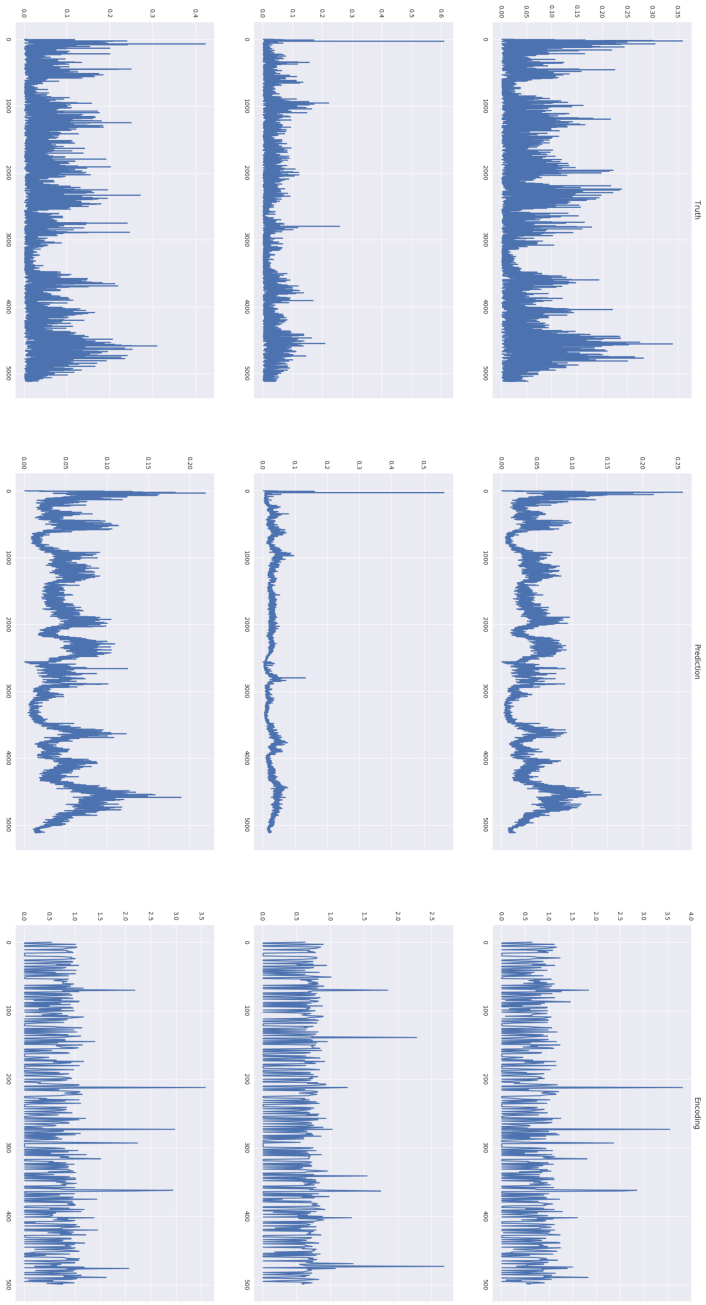
11. Bijlage B - Requirements

Legenda:

- URF: User Requirement Functional
- TRF: Technical Requirement Functional
- NFR: Non Functional Requirement

Referentie	Prioriteit	Requirement
URF 1	Must	De gebruiker moet via een portal een .csv bestand met vibratiedata over een rollager kunnen uploaden.
URF 2	Must	Het systeem moet aan de hand van de geuploade data voorspellen hoe lang het duurt voordat de rollager faalt.
URF 3	Should	Het systeem moet aan de hand van de geuploade data een gezondheidsindicatie van de rollager kunnen bepalen.
URF 4	Must	Het systeem moet de voorspelling teruggeven aan de gebruiker.
URF 5	Must	Het systeem moet aangeven wat de afwijking van de voorspelling is.
URF 6	Should	Het systeem moet een gezondheidsindicatie van de rollager teruggeven aan de gebruiker.
URF 7	Should	Het systeem moet de gezondheidsindicatie en de voorspelling verwerken in een PF-curve.
URF 8	Could	Het systeem moet een foutdiagnose stellen als er een fout gedetecteerd is.
URF 9	Could	Het systeem moet de foutdiagnose terug geven aan de gebruiker.
TRF 1	Must	Het systeem moet data uit .csv bestanden kunnen importeren. Dat wil zeggen losse rijen en losse velden.
TRF 2	Must	Het systeem moet data uit .csv bestanden kunnen verrijken met informatie over de tijd tot falen.
NFR 1	Must	Het systeem moet in zijn volledigheid op de meestgebruikte besturingssystemen kunnen draaien.
NFR 2	Must	Het systeem moet te allen tijde binnen 10 seconden van upload feedback geven aan de gebruiker, zij het met een voorspelling of een foutmelding.
NFR 3	Should	Het systeem moet ook zonder internetverbinding te gebruiken zijn.

12. Bijlage C - Autoencoding



Figuur 16: Kantel naar links. Van links naar rechts: FFT van signaal, reconstructie FFT, aangeleerde encoding