

# Hoe kan ik AI gebruiken om anomalieën te detecteren in motorgeluiden?

## Inhoud

Onderzoeksvragen .....	3
Welke soorten anomalieën zijn er? .....	3
Punt anomalieën .....	3
Contextuele anomalieën .....	3
Collectieve anomalieën .....	4
Welk soort model is het meest geschikt om anomalieën op te sporen? .....	5
Isolation forest .....	5
Local outlier factor .....	5
One class SVM .....	6
Resultaten .....	6
Kan een autoencoder beter presteren dan bestaande anomalie algoritmen? .....	8
CNN model .....	8
Undercomplete convolutional autoencoder .....	8
Denoising convolutional autoencoder .....	9
Resultaat undercomplete AE & undercomplete denoising AE .....	9
Undercomplete (denoising) autoencoder & CNN .....	10
Conclusie .....	10
In welke mate zijn false positives of false negatives belangrijk? .....	11
Wat zijn false positives (FP) en false negatives (FN). .....	11
Belang FN en FP .....	11
Welke methoden bestaan er al om anomalieën of falende motoren op te sporen? .....	12
Welke voordelen of nadelen haal je uit het detecteren van anomalieën in motorgeluiden? .....	13
Welke rol spelen achtergrondgeluiden? .....	15
Conclusie .....	16
Bronnen .....	17

## Onderzoeksvragen

### Welke soorten anomalieën zijn er?

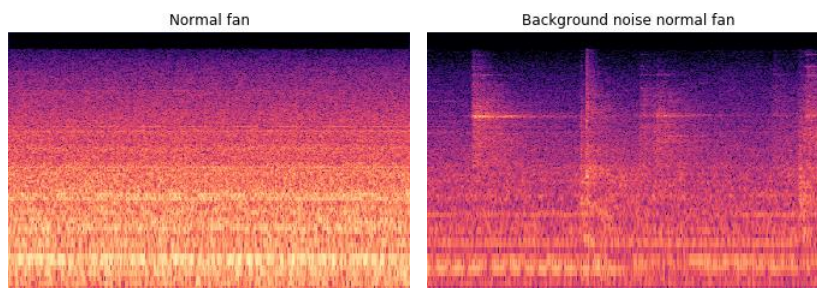
Binnen onderzoek naar anomalieën zijn er 3 grote soorten die zich voordoen, namelijk:

- Punt anomalieën
- Contextuele anomalieën
- Collectieve anomalieën

### Punt anomalieën

Punt anomalieën zijn punten die niet thuishoren in een bepaalde set.

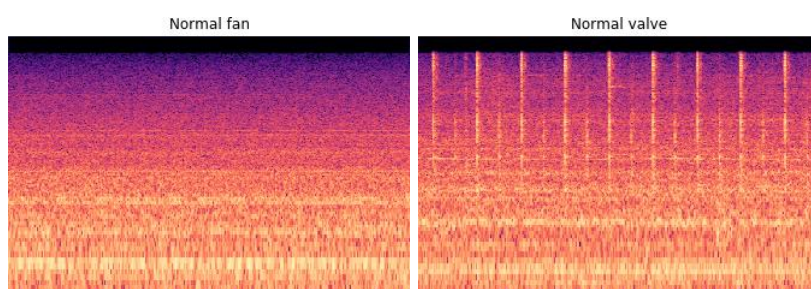
Ook binnen anomalie onderzoek in geluid zijn er punt anomalieën. Dit zijn dan waarnemingen zoals deze te zien op de afbeelding: “Background noise normal fan”. Hier zorgt een achtergrondgeluid bij een toch normaal werkende machine voor een zeer grote piek, op de afbeelding “Normal fan” kunt u een normaal geluid zien waar de piek niet aanwezig is.



### Contextuele anomalieën

Contextuele anomalieën zijn anomalieën die niet thuishoren in een bepaalde set komende door de context, bijvoorbeeld als we -5 graden Celsius als temperatuur in België gaan meten is dit niet noodzakelijk een anomalie, maar het is wel een anomalie als we dit meten in de zomer. Het zijn van een anomalie of niet hangt dus volledig af van de context.

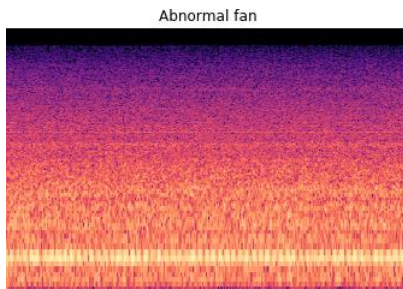
Een voorbeeld hiervan binnen het onderzoek naar falende machines aan de hand van het geluid ervan is bijvoorbeeld het geluid van een compleet andere machine zoals een “valve”, komt terecht bij de geluiden afkomstig van een “fan”. Beide zijn geluiden van een normaal werkende machines en op zich dus geen anomalieën, maar zijn toch anomalieën omdat ze bij elkaar in dezelfde set worden geplaatst.



### Collectieve anomalieën

Collectieve anomalieën zijn anomalieën die bestaan uit meerdere punten. Het is bijvoorbeeld een anomalie om 30 keer hetzelfde cijfer te rollen met een normale dobbelsteen.

Deze soort anomalieën zullen de belangrijkste zijn binnen het geluid van machines. Indien er een machine defect is zal dit meestal geen plotse piek veroorzaken (tenzij er net iets afbreekt), maar zal dit een reeks punten zijn die een andere waarde hebben. Op de afbeelding “Abnormal fan” is er bijvoorbeeld een steeds terugkomende toon te zien in het geluid dat aangeeft dat de machine defect is..



### Welk soort model is het meest geschikt om anomalieën op te sporen?

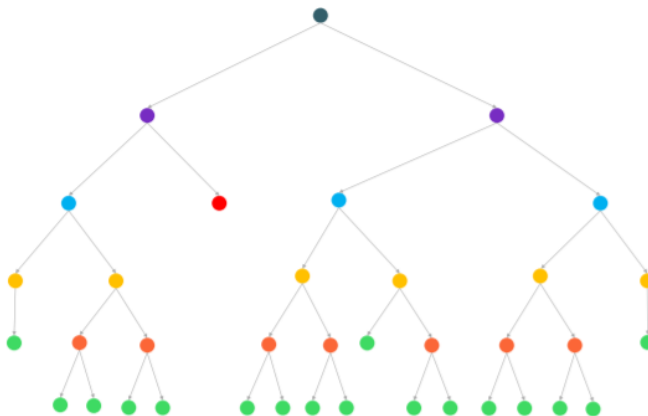
Er zijn tal van verschillende soorten anomalie algoritmes die zouden kunnen gebruikt worden, maar binnen dit onderzoek wordt er echter gefocust op 4 verschillende anomalie algoritmes, namelijk: Isolation forest, Local outlier factor, One class SVM en een autoencoder waarvan er verschillende versies zullen worden getest.

#### Isolation forest

Een isolation forest is een algoritme dat elk datapunt probeert op te splitsen volgens de waarden van de verschillende features. Het algoritme gaat volgens een omgekeerde boom te werk, het gaat namelijk verschillende vertakkingen gaan maken gebaseerd op de waarde van een feature en de threshold waarbij de sample wordt opgesplitst. Deze vertakkingen gaat het algoritme maken tot elke sample in een apart blad terecht komt, met andere woorden tot wanneer er niet meer kan worden opgesplitst.

Doordat er slechts een paar voorwaarden/vertakkingen nodig zijn om de anomalieën te gaan afzonderen zullen ze sneller terecht gaan komen in een afzonderlijk blad, ze komen met andere woorden sneller in isolatie terecht. Normale samples daarentegen zullen veel meer voorwaarden te zien krijgen voor ze in isolatie zijn terechtgekomen en zullen dus meer vertakkingen doormaken.

Classificatie gebeurt dan bij gevolg op basis van het aantal vertakkingen die een sample heeft gezien vooraleer die in isolatie terecht komt, met andere woorden op basis van de "path length". Een anomalie zal een kleine "path length" hebben, terwijl een normale sample een grote "path length" zal hebben.

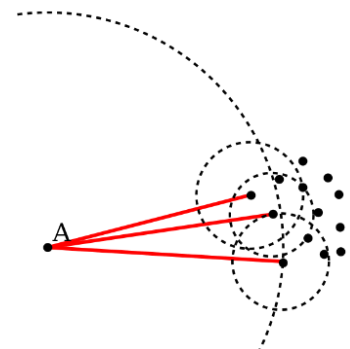


#### Local outlier factor

Local outlier factor is een unsupervised algoritme die voor elke sample de anomalie score berekent zijnde de local outlier factor.

Het algoritme gaat het verschil in dichtheid gaan berekenen van een sample rekening houdend met z'n burens. Een sample wordt als anomalie geclassificeerd indien z'n eigen dichtheid kleiner is dan die van z'n burens.

De parameter **n\_neighbors** gaat bepalen naar de hoeveel dichtste buur er wordt gekeken om de dichtheid te gaan meten. Hierdoor kan ervoor gezorgd worden dat er niet



enkel wordt gekeken naar de dichtste burens maar ook naar burens die verder af liggen.

#### One class SVM

One class SVM is een supervised algoritme, het model wordt getraind met de data van enkel 1 klasse, namelijk de normale samples.

Het algoritme gaat de eigenschappen van de normale samples gaan leren, waardoor het later dan kan bepalen welke samples hiervan afwijken en dus anomalieën zijn.

#### Resultaten

Er werden verschillende variaties van de geluiden gebruikt als input. Er werd gebruik gemaakt van de ruwe geluiden en van afbeeldingen van de geluiden. Binnen de afbeeldingen werden dan nog eens verschillende varianties gebruikt zoals een andere representatie of andere grote van de afbeelding.

Het gebruik van de ruwe geluiden had gelijkaardige resultaten aan die van de afbeeldingen, maar de tijd nodig om het model te gaan trainen en evalueren was veel langer, daarom werd er geen gebruik meer gemaakt van de ruwe geluiden, maar werden de afbeeldingen gebruikt als input.

Er werden ook verschillende groottes van training data gebruikt. De bevinding hierbij was, hoe meer data er gebruikt wordt, hoe minder accuraat het model wordt. Dit komt doordat de geluiden vaak in verschillende omstandigheden werden opgenomen en dus meer varianties aanwezig zijn. Door een grotere trainingsdata te gebruiken werd het model wel robuuster tegen de verschillende omstandigheden.

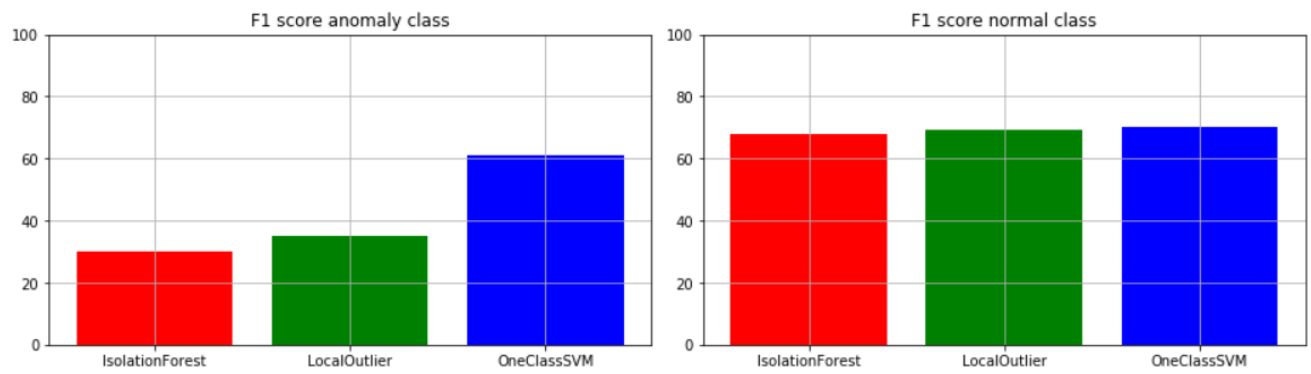
De beste modellen werden gecreëerd door het gebruik van de CQT representaties (constant-Q transform) van de geluiden en zonder de grote van de afbeeldingen te wijzigen. Er werd ook geen normalisatie meer toegepast, omdat de pixelwaarden zich al op een goede schaalverdeling bevonden.

#### Beste model

Het beste resultaat van de verschillende anomalie algoritmes werd verkregen door de One Class SVM. Bij deze beslissing werd er voornamelijk gekeken naar de [FP en FN](#) scores. Er waren namelijk 0 False negatives. Dit wil zeggen dat het model geen enkele anomalie classificeerde als een normale sample, iets wat bij dit soort onderzoek zonder twijfel 1 van de belangrijkste eigenschappen is. Jammer genoeg waren er wel enkele False Positives. Dit wil zeggen dat het model normale samples gaat classificeren als een anomalie. Dit is ten koste van de precisie van de anomalie klasse. Ook de accuraatheid van de One Class SVM was het beste van de verschillende anomalie algoritmes.

Bij de modellen komende van Isolation Forest en Local Outlier Factor waren er telkens teveel False Negatives om het model te gaan overwegen in de praktijk. Amper de helft van de anomalieën werden door de algoritmes ontdekt.

Aan de hand van de F1-scores op de verschillende klassen is er een handig overzicht te zien van de 3 verschillende anomalie algoritmes.



### Kan een autoencoder beter presteren dan bestaande anomalie algoritmen?

Doordat er geen standaard bestaat voor een anomalie binnen het geluid van een machine, kan er geen gebruik gemaakt worden van een classificatie model om de anomalieën te gaan onderscheiden van de normale geluiden. Er bestaan namelijk veel te veel manieren hoe een onderdeel defect kan gaan en bij gevolg dus teveel verschillende geluiden. Hierdoor wordt het ook onmogelijk om voor data van alle verschillende situaties te zorgen laat staan om op alle mogelijkheden te gaan trainen. Een standaard classificatie algoritme zoals een CNN zal hier dus niet volstaan. Daarom zal er dus gebruikt gemaakt worden van een daarvoor bedoeld anomalie algoritme om de anomalieën te gaan opsporen.

### CNN model

Eerst werd er echter toch een CNN gebruikt worden om te bepalen of de data wel geclassificeerd kan worden en welke representatie (afbeelding) van de geluiden hiervoor het best geschikt is.

Het CNN model was in staat om een accuraatheid te verkrijgen van 90% op de grootste dataset mogelijk, dit wel na een relatief lange trainingstijd. Ook de FP en FN scores waren goed genoeg om het model effectief te gaan gebruiken. Indien er dus op alle verschillende soorten anomalieën getraind zou kunnen worden, zou een CNN een goede methode zijn om te gaan gebruiken in de praktijk.

Image augmentation was niet in staat om de score van het model te gaan verbeteren. Ook van transfer learning is er geen gebruik meer gemaakt, dit doordat de afbeelding niet naar de correcte dimensies kon worden omgezet die nodig waren voor transfer learning.

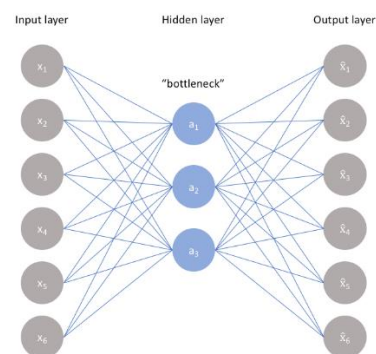
Het beste resultaat afkomstig van het CNN algoritme wordt ook gebruikt in de interface, omdat er aan dit model ook een probabilliteit kan gekoppeld worden aan het al dan niet zijn van een anomalie.

### Undercomplete convolutional autoencoder

Een autoencoder is een neurale netwerk dat data afkomstig van 1 klasse (supervised) zo goed mogelijk gaat reconstrueren.

Het neurale netwerk krijgt als input data te zien afkomstig van 1 klasse, vervolgens gaat het neurale netwerk de data gaan representeren in een kleinere dimensie om dan vervolgens weer zo goed mogelijk te gaan reconstrueren in de originele dimensie.

Een autoencoder kan enkel de data gaan reconstrueren gelijkaardig aan de input. Dit wil zeggen dat het data die niet gelijkaardig is aan de input (anomalieën) niet zal kunnen reconstrueren en dus ook een grote loss zal hebben op deze data. Op basis van deze loss en een threshold zal er dan worden geclassificeerd.



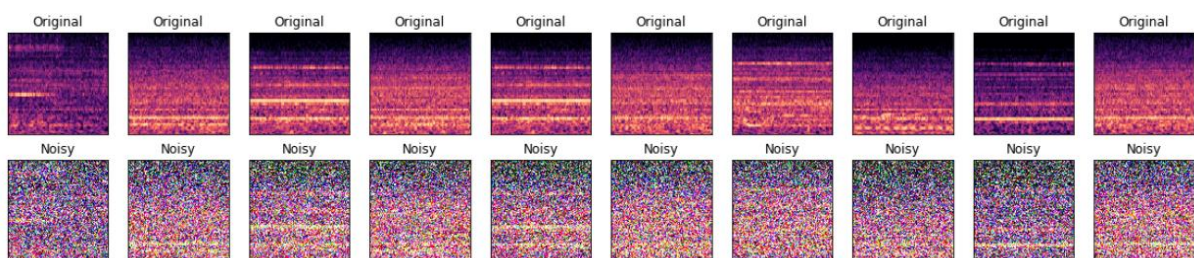


De loss is het verschil tussen de input en de (gereconstrueerde) output. In dit geval zal er gebruik gemaakt worden van de MSE om de loss te gaan berekenen.

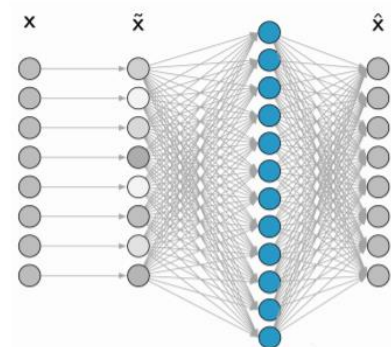
Het convolutional deel van de autoencoder wil zeggen dat er gebruikt gemaakt zal worden van convolution en pooling lagen in het model. Als input zal het model een afbeeldingen te zien krijgen die het netwerk dan zo goed mogelijk gaat reconstrueren in de output.

#### Denoising convolutional autoencoder

Een denoising convolutional autoencoder verschilt qua structuur niet veel van een standaard autoencoder, het enige verschil zit in de input. Alvorens de data door het netwerk te sturen wordt er ruis toegevoegd aan de input. Het is dan de bedoeling dat het netwerk herkent dat het over ruis gaat en deze weg filtert. In de output moet het netwerk de originele data zonder ruis gaan reconstrueren.

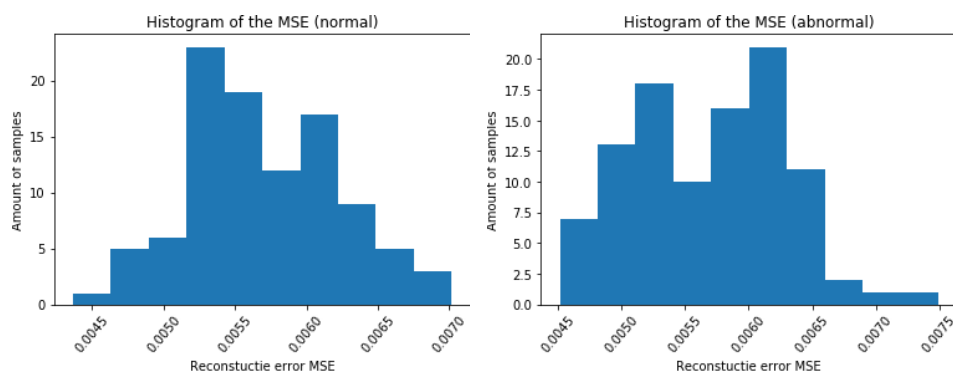


Dit soort strategie wordt meestal toegepast bij een overcomplete autoencoder als regularisatie zodat de input niet naar de output kan worden gekopieerd. Bij een overcomplete autoencoder heeft de bottleneck een grotere dimensie dan de input en output lagen. Bij dit onderzoek zal het echter worden toegepast bij een undercomplete autoencoder.



#### Resultaat undercomplete AE & undercomplete denoising AE

Het resultaat afkomstig van een standaard undercomplete autoencoder is niet in staat om de anomalieën te onderscheiden van de normale data. Dit komt doordat de AE de 2 klassen telkens even goed/slecht reconstrueert, de AE is met andere woorden niet in staat om de verschillen van de data bloot te leggen. Dit is ook zeer goed te zien op de histogram van de MSE van de verschillende klassen.



Ook het resultaat afkomstig van een undercomplete denoising autoencoder was niet in staat de anomalieën te onderscheiden van de normale data.

#### Undercomplete (denoising) autoencoder & CNN

Een 3<sup>de</sup> variant die geprobeerd werd was het resultaat afkomstig van een undercomplete autoencoder combineren met een CNN algoritme. Hier werd er geclassificeerd aan de hand van de latent representaties komende van de autoencoder. De latent representatie van een autoencoder is de kleinste representatie van de data gecreëerd in de bottleneck van die autoencoder. Deze data werd dan gebruikt als input voor het CNN algoritme.

Het resultaat komende van deze methode presteerde zeer goed op de train data en de test data vergeleken met de resultaten komende van enkel een autoencoder. Het model was in staat om de samples voor 88% juist te classificeren met een F1-score op de abnormale klasse van 0.79 en een F1-score van 0.92 op de normale klasse.

Wanneer het model echter werd getest op compleet nieuwe data, was het resultaat zeer slecht. Het model kon geen anomalieën meer onderscheiden van normale samples. Dit komt doordat de samples waarop getraind werd en geëvalueerd werd zeer goed op elkaar lijken, ze werden namelijk binnen dezelfde omstandigheden opgenomen. Het model kon hierdoor een zeer goed resultaat bekomen, maar wanneer er getest werd op andere samples uit andere omstandigheden faalde het model compleet.

#### Conclusie

Een autoencoder kon bij deze data niet beter presteren dan het beste resultaat van een daarvoor bestemd anomalie algoritme, zeker wanneer er gekeken wordt naar het belang van [FP en FN](#) scores en naar de accuraatheid van het model.

In welke mate zijn false positives of false negatives belangrijk?

Wat zijn false positives (FP) en false negatives (FN).

Indien het verkregen model een geluid toegestuurd krijgt zal het model 1 van de 2 volgende zaken terugsturen: een correcte of foutieve voorspelling. Indien de voorspelling foutief is, zijn er weer 2 mogelijkheden, namelijk: een false positive (FP) of false negative (FN).

Een FP betekent dat het model voorspelt dat de sample een anomalie was, terwijl de sample effectief een normale sample was. Een FN betekent dat het model voorspelt dat de sample normaal was terwijl de sample effectief een anomalie was.

	Abnormal Prediction	Normal Prediction
Abnormal Actual	True positive	False Negative
Normal Actual	False Positive	True Negative

Belang FN en FP.

Bij dit onderzoek naar anomalie detectie gaat de voorkeur uit om het aantal false negatives zo laag mogelijk te houden, terwijl toch nog zo weinig mogelijk false positives te hebben.

De FP en FN waarden zijn veel belangrijker dan de accuraatheid van het model. De accuraatheid is namelijk een zeer slechte factor om een model te evalueren bij anomalie detectie. Dit komt doordat de data meestal zeer ongebalanceerd is en een model bijvoorbeeld een accuraatheid kan hebben van 90% terwijl alle samples in dezelfde klasse worden geclassificeerd.

Bij de autoencoder zal er hiervoor nood zijn om de optimale threshold te vinden waarbij alle samples zo optimaal mogelijk tot de verschillende klassen worden geclassificeerd, om te zorgen voor zo weinig mogelijk false negatives.

De reden waarom er zo weinig mogelijk false negatives mogen zijn is om zo weinig mogelijk anomalieën onopgemerkt voorbij te laten gaan. Telkens er een FN is wil dit zeggen dat er een geluid was die eigenlijk een anomalie was, maar verkeerd werd geclassificeerd door het model en bijgevolg dus voor ernstige problemen kan zorgen in de praktijk.

### Welke methoden bestaan er al om anomalieën of falende motoren op te sporen?

Er bestaan al verschillende methoden die op dit moment gebruikt worden om falende motoren te gaan opsporen. Enkele van deze methoden zijn de volgende:

Een 1<sup>ste</sup> methode is om een werknemer te gaan rondsturen bij de verschillende machines en deze 1 voor 1 te laten controleren op defecten. Dit is voor een bedrijf een zeer dure en tijd innemende methode. Dit omdat ze iemand moeten betalen om dit werk te gaan uitvoeren en doordat de machine een tijd onnodig zal stil liggen. Bijgevolg kan er geen gebruik gemaakt worden van de machine waardoor de kosten al zeer snel oplopen.

Een 2<sup>de</sup> methode is om de machine preventief te gaan vervangen. Dit soort methode wordt bijvoorbeeld zeer veel toegepast bij auto's. Na een bepaald aantal kilometers wordt bijvoorbeeld de waterpomp al vervangen ook al kon het voertuig nog enige tijd verder zonder vervanging. Dit is momenteel de veiligste optie waar er voor kan gekozen worden.

Een 3<sup>de</sup> methode is om de machine te gaan vervangen wanneer deze effectief defect is. Voor deze optie wordt er nagenoeg nooit gekozen, omdat deze vaak zeer veel extra kosten met zich meebrengt die konden vermeden worden.

Een 4<sup>de</sup> methode is predicitive maintenance. Hier worden verschillende sensoren bevestigd aan de machine die na een bepaald tijds interval de gemeten waarden gaan opslaan. Na verloop van tijd worden de waarden van de verschillende sensoren doorgestuurd naar een AI model, die dan in het beste geval kan voorspellen wanneer de machine in kwestie defect zal zijn of hoeveel cyclussen er nog doorlopen kunnen worden vooraleer de machine defect zal zijn, in de volgende tabel wordt dit aangegeven door de kolom "tff".

engine_id	setting1	setting2	setting3	sensor4	sensor5	sensor6	sensor7	tff
1	-0.0007	-0.0004	100.0	1400.60	14.62	21.61	554.36	191
1	0.0019	-0.0003	100.0	1403.14	14.62	21.61	553.75	190
1	-0.0043	0.0003	100.0	1404.20	14.62	21.61	554.26	189
1	0.0007	0.0000	100.0	1401.87	14.62	21.61	554.45	188
1	-0.0019	-0.0002	100.0	1406.22	14.62	21.61	554.00	187
1	-0.0043	-0.0001	100.0	1398.37	14.62	21.61	554.67	186
1	0.0010	0.0001	100.0	1397.77	14.62	21.61	554.34	185
1	-0.0034	0.0003	100.0	1400.97	14.62	21.61	553.85	184
1	0.0008	0.0001	100.0	1394.80	14.62	21.61	553.69	183
1	-0.0033	0.0001	100.0	1400.46	14.62	21.61	553.59	182
1	0.0018	-0.0003	100.0	1400.64	14.62	21.61	554.54	181

### Welke voordelen of nadelen haal je uit het detecteren van anomalieën in motorgeluiden?

Tegenwoordig wordt er binnen de industrie zeer veel geautomatiseerd, kijk maar naar de automatisatie van bijvoorbeeld de productielijnen van auto's tot die van zeep producten. Bij deze automatisering komen er zeer veel machines bij kijken, deze machines zullen hoe dan ook na verloop van tijd defect zijn. Om te voorkomen dat er lange wachttijden ontstaan om de machine te herstellen of om ervoor te zorgen dat de machine kan gerepareerd worden voordat hij effectief ernstig beschadigd wordt kan er gebruik gemaakt worden van AI.

Binnen dit onderzoek wordt er 1 methode onderzocht om AI te gaan toepassen. Bij de machines kan namelijk een microfoon geplaatst worden die het geluid van de machine en die van de omgeving gaat opnemen. Dit geluid wordt dan door een AI gestuurd die dan bepaald of er een anomalie zit in het geluid of niet, er wordt m.a.w. bepaald of de machine in kwestie een defect heeft of niet.

Doordat dit proces volledig automatisch gebeurt zijn er tal van voordelen om dit te gaan implementeren ten opzicht van de [bestaande methoden](#) om een falende motor op te sporen. Jammer genoeg komen er ook enkele nadelen bij kijken ten opzichte van de andere methoden.

#### *predictive maintenance methode*

Een methode die bij dit soort onderzoek vast en zeker in gedachten springt is predictive maintenance. Bij predictive maintenance kan er in het beste geval exact voorspeld worden wanneer een machine defect zal zijn. Het voorspellen wanneer een machine precies defect zal zijn is dan ook één van de voordelen die predictive maintenance heeft ten opzichte van detectie met geluiden dat binnen dit onderzoek onderzocht wordt. Bij de detectie met geluiden die binnen dit onderzoek onderzocht wordt kan er niet aangegeven worden wanneer een machine precies een defect zal hebben, er kan enkel aangegeven worden of een machine al dan niet defect is. Het niet kunnen voorspellen van wanneer een machine precies defect zal zijn komt doordat er bij de gebruikte dataset geen tijdsindicatie of voorafgaande cyclussen aanwezig zijn. Een mogelijke verbetering van de dataset om dit wel mogelijk te maken zou de volgende kunnen zijn: Eén lange opname van het geluid van de machine over de gehele levensduur ervan, die later kan worden opgesplitst in kleinere opnames, waardoor er ook nog steeds een gelabelde dataset wordt overhouden, maar met dit keer als extra de data van de voorafgaande cyclussen.

De detectie aan de hand van geluiden heeft ook z'n voordelen tegenover die van predictive maintenance waar sensoren gebruikt worden. Sensoren moeten namelijk vaak zeer dicht geplaatst worden bij het onderdeel zelf om de waarden te kunnen opmeten. In veel gevallen is er echter geen plaats voorzien om dit te doen waardoor de sensoren ergens anders of gewoon niet geplaatst kunnen worden. Een microfoon daarentegen kan het geluid ook opmeten wanneer deze op een andere plaats bevestigd wordt. Hierdoor wordt de detectie aan de hand van geluid nagenoeg in elk geval mogelijk, in tegenstelling tot de detectie aan de hand van sensoren.

#### *Methode met menselijke input*

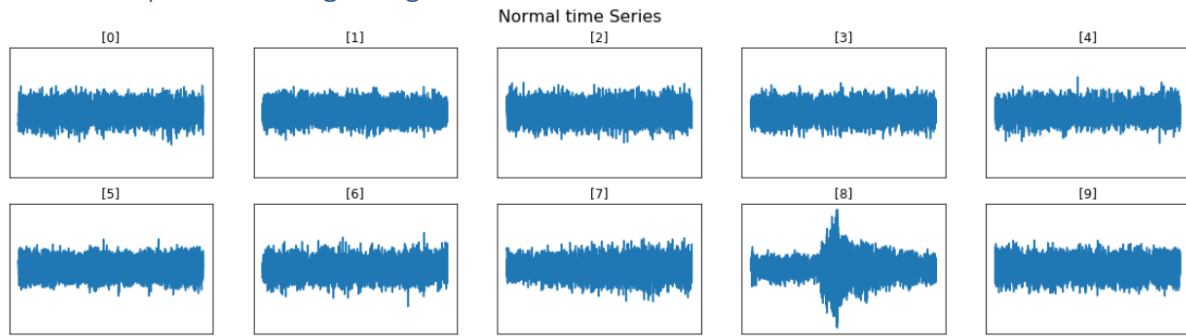
De detectie aan de hand van geluid en AI heeft zeer veel voordelen tegenover die waar menselijke input voor nodig is, enkele van deze voordelen zijn:

- Eenmaal aangekocht is het goedkoper dan het betalen van werknemers.
- Er is geen limiet op hoeveel machines er worden gemonitord.
- Alle machines kunnen tegelijk gemonitord worden.
- Er zijn geen onnodige kosten meer door machines te vroeg te vervangen.
- Ecologisch beter verantwoord doordat er minder verspilling is
- ...

Jammer genoeg zijn er ook nadelen verbonden aan het gebruik van deze oplossing. Het model kan namelijk last hebben van [false negatives](#) en/of [false positives](#). Bij het gebruik van geluid kunnen [achtergrondgeluiden](#) ook een zeer grote invloed hebben op het model, waardoor een sample zou kunnen worden geclassificeerd als een anomalie terwijl dit het niet is. De achtergrondgeluiden zullen echter niet zorgen voor false negatives, dit omdat een achtergrondgeluid er niet voor zal zorgen dat een defecte machine zal klinken als een goed werkende machine. Er zullen dus geen anomalieën geclassificeerd worden als normaal door de achtergrondgeluiden die zich kunnen voordoen.

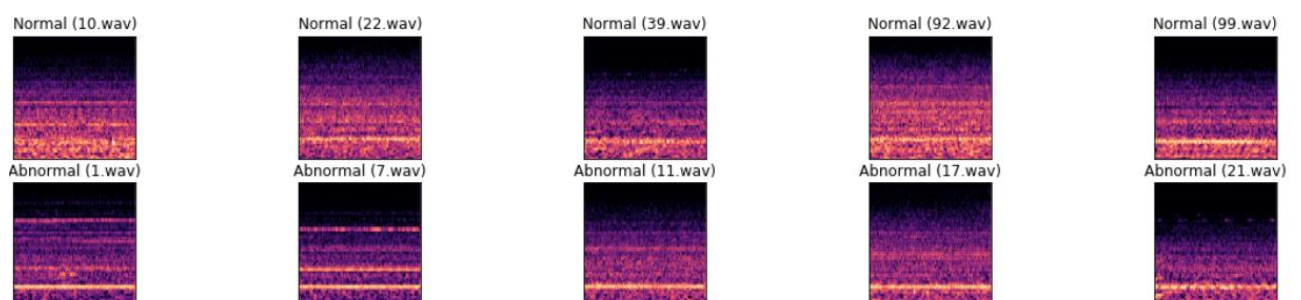


## Welke rol spelen achtergrondgeluiden?



Dit zijn allemaal geluiden afkomstig uit de klasse van normaal werkende machines, maar zoals hier te zien bij geluid '[8]' komt er een grote piek voor. Deze piek is niet afkomstig door het falen van het onderdeel, maar is afkomstig door een achtergrondgeluid. Dit geluid is door het achtergrondgeluid zeer verschillende van de andere geluiden in de klasse en zou door de mens bij het zien van enkel de afbeelding meteen worden geassocieerd als een anomalie.

Achtergrondgeluiden spelen echter geen grote rol bij het classificeren van geluidsamples, bij het nakijken van de fouten die het model maakt, wordt zo goed als geen enkele sample slecht geassocieerd door de achtergrondgeluiden. Dit komt doordat er bij de trainingsdata ook al zeer veel achtergrondgeluiden aanwezig zijn in beide klasse. Het model is in staat de achtergrondgeluiden er zo goed als allemaal uit te filteren en ze te gaan negeren. Ook door te zorgen voor zo veel mogelijk data om op te gaan trainen, blijkt uit observatie dat er minder tot geen misclassificaties zijn komende door de achtergrondgeluiden. Op de volgende afbeelding zijn enkele misclassificatie te zien van beide klassen. De titel geeft aan wat de eigenlijke klasse is, het tegenovergestelde van de titel is wat het model voorspelde.



Er is wel een mogelijkheid om voor minder achtergrondgeluiden te gaan zorgen in de data. Namelijk door de microfoon zo dicht mogelijk te bevestigen bij de machine in kwestie en zonder fysiek contact zodat trillingen en dergelijke kunnen vermeden worden. Ook het zorgen voor zo weinig mogelijk geluid van buitenaf door bijvoorbeeld isolatie kan zorgen voor minder achtergrondgeluiden.

## Conclusie

In de praktijk biedt deze methode tal van voordelen ten opzichte van andere methoden om falende machines te gaan opsporen. Bij een volledig uitgewerkt, compleet en geïnstalleerd product is dit soort onderzoek zeker in de praktijk een mogelijke oplossing.

De huidige interface die werd gecreëerd is echter nog niet gebruik klaar en dient enkel als demo. Om het een gebruik klaar product te maken zou het een autonoom product moeten worden.

Om deze oplossing te gaan evalueren volgens de ROI of return on investment zouden de effectieve prijzen moeten gekend zijn van hoe er nu wordt gecontroleerd binnen een bedrijf. Maar doordat dit een oplossing is die zo goed als geen verdere ontwikkelingen en extra kosten nodig heeft en doordat dit ook een lang termijn oplossing is, is deze oplossing het hoe dan ook waard op lange termijn volgens de ROI. Er zal geen winst gegenereerd worden uit deze oplossing, maar er zal wel een besparing zijn op de huidige kosten, wat de ROI ook ten goede komt.



## Bronnen

- [1]  
C. Dawson, "Outlier Detection with One-Class SVMs", *Medium*, 28-aug-2019.  
[Online]. Beschikbaar op: <https://towardsdatascience.com/outlier-detection-with-one-class-svms-5403a1a1878c>. [Geraadpleegd: 14-jan-2020].
- [2]  
Synced, "What Does 'Broken' Sound Like? First-Ever Audio Dataset of Malfunctioning Industrial Machines", *Medium*, 01-okt-2019. [Online]. Beschikbaar op: <https://medium.com/syncedreview/what-does-broken-sound-like-first-ever-audio-dataset-of-malfunctioning-industrial-machines-b4f8f6d81dd7>. [Geraadpleegd: 06-jan-2020].
- [3]  
P. Wenig, "Local Outlier Factor for Anomaly Detection", *Medium*, 01-aug-2019.  
[Online]. Beschikbaar op: <https://towardsdatascience.com/local-outlier-factor-for-anomaly-detection-cc0c770d2ebe>. [Geraadpleegd: 11-jan-2020].
- [4]  
P. Wenig, "Local Outlier Factor for Anomaly Detection", *Medium*, 01-aug-2019.  
[Online]. Beschikbaar op: <https://towardsdatascience.com/local-outlier-factor-for-anomaly-detection-cc0c770d2ebe>. [Geraadpleegd: 14-jan-2020].
- [5]  
"Predictive maintenance with airborne sound analysis", *Processing Magazine*, sep-2017. [Online]. Beschikbaar op: <https://www.processingmagazine.com/process-control-automation/article/15587140/predictive-maintenance-with-airborne-sound-analysis>. [Geraadpleegd: 14-jan-2020].
- [6]  
"(26) DSP Background - Deep Learning for Audio Classification p.1 - YouTube".  
[Online]. Beschikbaar op: [https://www.youtube.com/watch?v=Z7YM-HAz-IY&list=PLhA3b2k8R3t2Ng1WW\\_7MiXeh1pfQJQi\\_P](https://www.youtube.com/watch?v=Z7YM-HAz-IY&list=PLhA3b2k8R3t2Ng1WW_7MiXeh1pfQJQi_P). [Geraadpleegd: 06-jan-2020].
- [7]  
*1- Deep Learning (for Audio) with Python: Course Overview. .*
- [8]  
"Anomaly Detection Technique in Sound to Detect Faulty Equipment | NTT Technical Review". [Online]. Beschikbaar op: <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201708fa5.html>. [Geraadpleegd: 06-jan-2020].
- [9]  
*Librosa Audio and Music Signal Analysis in Python | SciPy 2015 | Brian McFee. .*
- [10]  
"Package - justgage". [Online]. Beschikbaar op: <https://npm.taobao.org/package/justgage>. [Geraadpleegd: 17-jan-2020].
- [11]  
"r/Python - I'm starting a free YouTube course called 'Deep Learning (for Audio) with Python'", *reddit*. [Online]. Beschikbaar op: [https://www.reddit.com/r/Python/comments/elb099/im\\_starting\\_a\\_free\\_youtube\\_course\\_called\\_deep/](https://www.reddit.com/r/Python/comments/elb099/im_starting_a_free_youtube_course_called_deep/). [Geraadpleegd: 21-jan-2020].
- [12]  
"SECOM Detecting Defected Items". [Online]. Beschikbaar op: <https://kaggle.com/snjumaheshwari/secom-detecting-defected-items>. [Geraadpleegd: 10-jan-2020].
- [13]

“Semi Supervised Classification using AutoEncoders”. [Online]. Beschikbaar op: <https://kaggle.com/shivamb/semi-supervised-classification-using-autoencoders>. [Geraadpleegd: 15-jan-2020].

[14]

“sklearn.neighbors.LocalOutlierFactor — scikit-learn 0.22.1 documentation”. [Online]. Beschikbaar op: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html>. [Geraadpleegd: 11-jan-2020].

[15]

“sklearn.neighbors.LocalOutlierFactor — scikit-learn 0.22.1 documentation”. [Online]. Beschikbaar op: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html>. [Geraadpleegd: 14-jan-2020].

[16]

Gevaert W. Cursus MCT Deep learning 2019

[17]

Dewitte M. Sessie over Anomaly\_Detection 2019