

MASTERS IN INNOVATION



**USING ARTIFICIAL INTELLIGENCE
TO CONTROL THE LIGHTING
ON FLEMISH HIGHWAYS**

Predictieve modellen - overzicht

08-01-2018

Jochem Grietens
Verhaert

Alexander Frimout
Verhaert

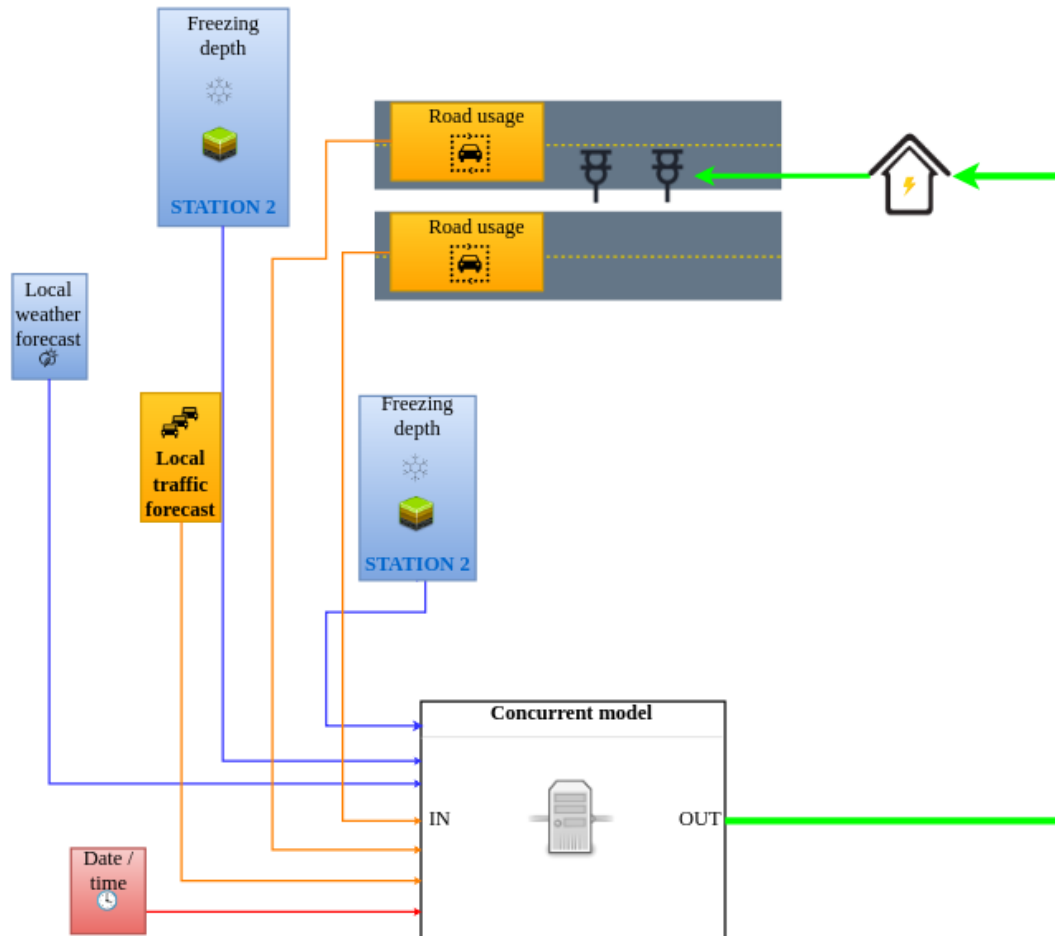
AI voor lichtcontrole

Het doel is om de verlichting van de Vlaamse snelweg slimmer te maken met behulp van A.I. technologie. Concreet hebben we historische en real-time data van verschillende bronnen alsook de logs van de effectieve lichtsturing.

Het Turing project heeft als doel deze data, expertkennis en contextgegevens samen te gebruiken om een model te maken dat de de sturing van de verlichting volledig of gedeeltelijk automatiseert. Zo'n systeem wordt een model genoemd en berekend een gewenste sturing uit de inkomende input gegevens. In dit geval neemt het model de inputs zoals lokale weersinformatie, vriesdiepte-metingen, road-usage data ,... om een storingssignaal naar de licht lussen te sturen.

In onderstaande afbeelding werd een vereenvoudigde voorstelling gemaakt van het soort systeem dat we trachten te bekomen. We zien het "concurrent model" onderaan centraal dat allerhande input data krijgt en als output een signaal naar de sturings cabine stuurt die op haar beurt de verlichtingslus aanstuurt.

In dit project is het belangrijk rekening te houden met een context waar het model niet volledig autonoom kan werken en nog overschreven kan worden in noodgevallen of door een extern opgelegde planning. Deze vereiste moet weerspiegeld worden in de uiteindelijke architectuur.



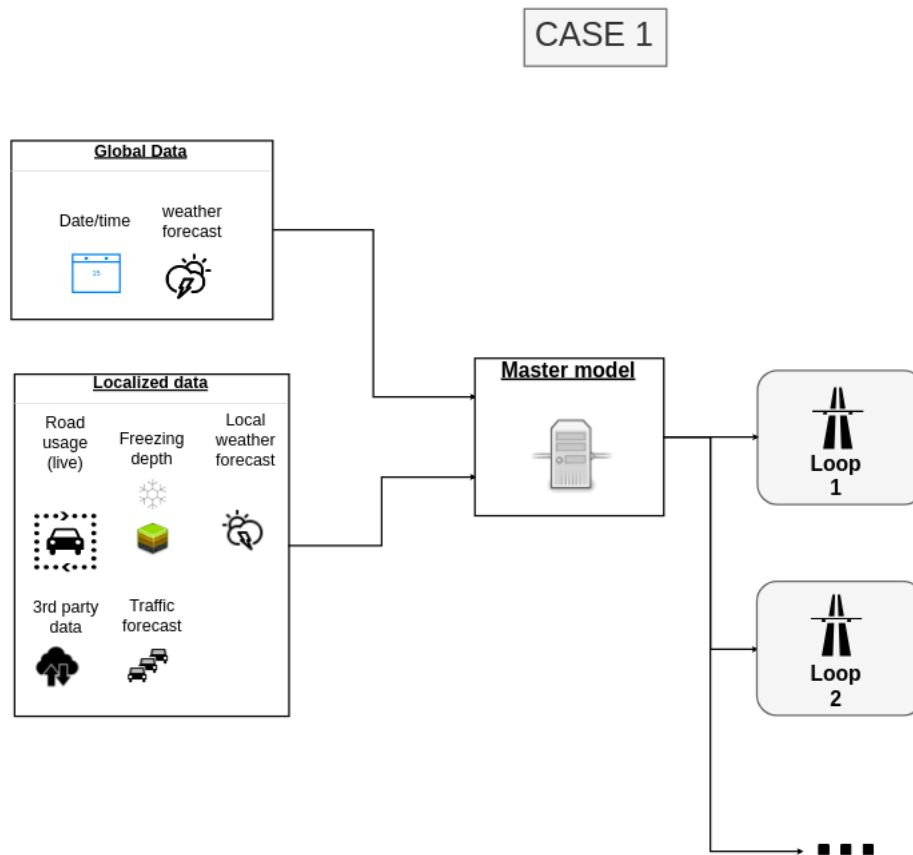
Model distributie

Een belangrijke overweging in de modelkeuze en architectuur design is de distributie van de modellen. Ter illustratie stellen we 3 vereenvoudigde cases voor.

Case 1 : Master model

Eén master model neemt als input de globale data en alle lokale data over heel het land. Als output geeft het model een sturend commando voor elke loop.

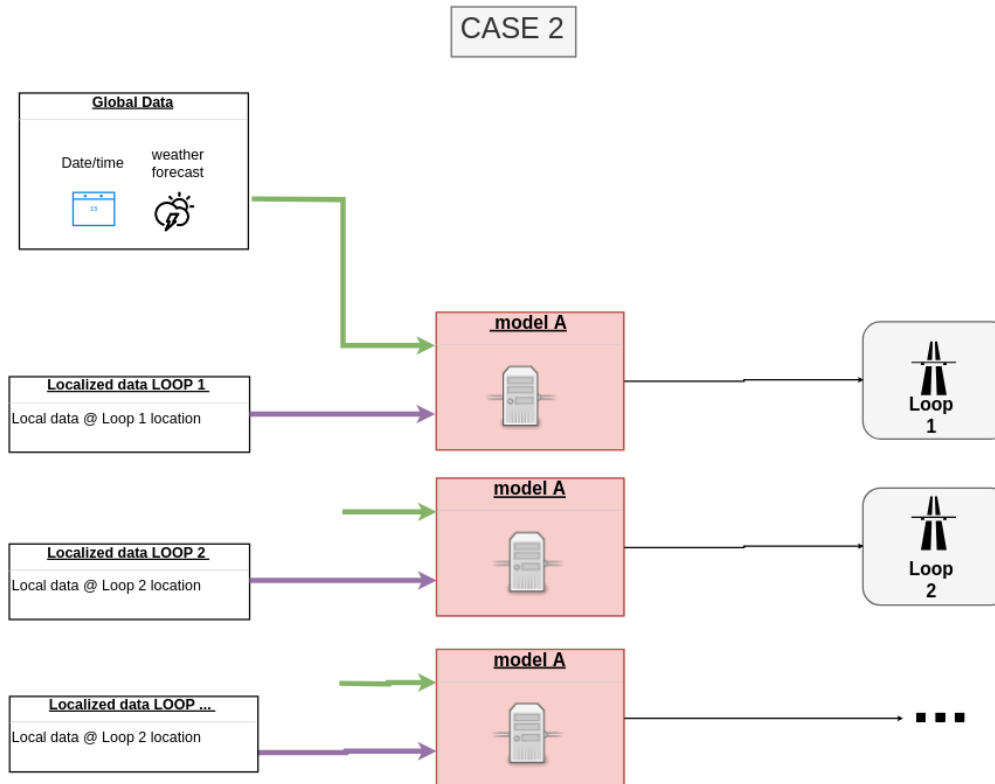
Zulk een model biedt weinig controle en overzicht tijdens ontwikkeling en onderhoud. Wij raden een modulaire approach aan zoals in Case 2 en 3.



Case 2: Lokaal model – niet uniek

Elke lus krijgt een eigen model maar dit model is identiek voor elke lus. Het model is dus hetzelfde voor elke lus, het verschil is dat elke instantie van het model de unieke lokale data krijgt die overeenkomt met de lus die het hoort te controleren.

De voordelen van deze verdeling zijn dat het makkelijker te ontwikkelen en onderhouden is. Er is minder historische data nodig om zelflerende algoritmes te gebruiken om het model te trainen.

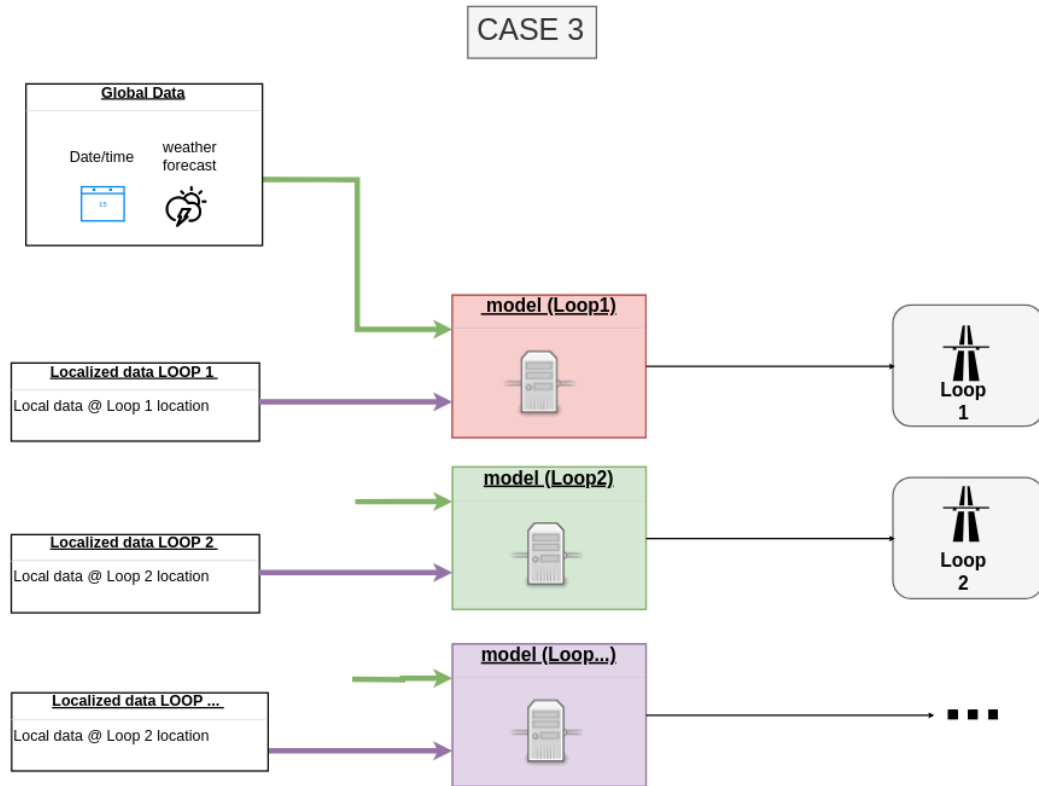


Case 3: Lokaal model – uniek

Case 3 is gelijkaardig aan case 2 met het verschil dat elk model uniek is. Elk model is ontwikkeld voor de specifieke lus en bijhorende locatie.

Het unieke voordeel aan dit systeem is dat de intelligentie uniek is aan de lokatie. Dit kan positief zijn voor performantie. De keerzijde is dat de ontwikkeling en het onderhoud oploopt. Er is ook een hogere vereiste van voldoende representatieve data voor elk

model.




Data

Het model zal verschillende input data bronnen gebruiken tijdens ontwikkeling en operatie. In het voorbeeldschema in bovenstaande afbeelding werden enkele voorbeelden genomen van het type data dat gebruikt kan worden. In de onderstaande paragrafen worden enkele opmerkingen gemaakt over hoe om te gaan met deze data. Deze lijst is niet exhaustief maar enkel illustratief om aan te tonen welke accenten belangrijk zijn bij de interactie tussen model en input data.

Datum/Tijd

Datum en tijd in standaard formaat. Het is belangrijk dat het model de dag van het jaar en de tijd in de dag kent. Het is mogelijk om additionele informatie te voorzien zoals een feestdagen kalender.

Road usage (live)



Data die komt van lussen, camera's en dergelijke en de real-time drukte op de baan aangeeft. Men kan verwachten dat er een variërend aantal van deze inputs zijn per licht-regel-lus. Bij voorkeur heeft een model hetzelfde aantal inputs voor elke licht-regel-lus. Om de input te standaardiseren over alle modellen/regel-lussen zijn er verschillende opties.

2 Voorstellen:

- Data aggregeren tot één "drukke-metric" die samengesteld is uit de verschillende metingen
- De X kortst bijzijnde/toegewezen lussen gebruiken als input van het model. Hier is X een constante voor alle lussen.

Local traffic forecast & reporting

Elke lus is toegewezen aan een specifieke regio. Er kunnen verschillende variabelen gedefinieerd worden voor de relevante metrieken: zichtbaarheid, kans op file, filelengte, gemiddelde snelheid, ...

Local weather forecast/station

Elke lus is toegewezen aan een specifieke regio. Er kunnen verschillende variabelen gedefinieerd worden voor de relevante metrieken: zichtbaarheid, schemer, bewolking, mist, kans op neerslag, ...

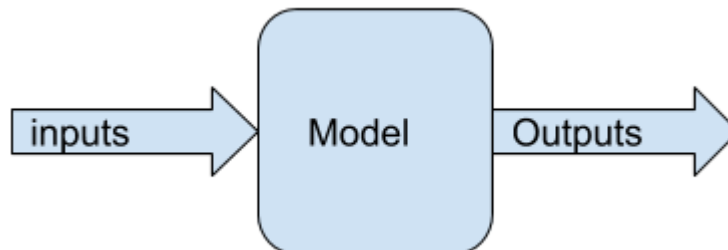
Vries diepte data

De bodemtemperatuur en vries-diepte wordt gemeten. Zoals bij de road-usage-data kan men verwachten dat er een variërend aantal van deze inputs zijn per regel-lus. Dezelfde maatregelen kunnen hier getroffen worden.

A.I. model types

Er zijn verschillende soorten modellen die verschillen in performantie, data requirements, transparantie, vereiste rekenkracht, 'intelligentie' en graad van autonomie. De uiteindelijke selectie van het juiste model kan pas gemaakt worden wanneer de use case en technical requirements volledig duidelijk zijn. Daarnaast is de aard en kwaliteit van de beschikbare data van zeer groot belang voor de keuze van het model en kan enkel een vergelijkende studie op echte data gebruikt worden om een finale selectie te maken op basis van performantie. In deze tekst wordt een aanzet gegeven om a-priori inzicht te krijgen in de kanshebbende modellen.

We maken een onderscheid tussen **2 soorten modellen**. **Expert-systems** zijn modellen gebaseerd op kennis van deskundigen en domeinkennis. **Machine learning-modellen** nemen een data-gedreven aanpak waarbij gegevens worden gebruikt om optimale modellen te leren.



In wat volgt worden enkele modeltypes voorgesteld. Deze lijst is niet exhaustief en dient aangevuld te worden. De voorgestelde modellen zijn gelijk in dat ze gestandaardiseerde inputgegevens aannemen, hier bewerkingen op uitvoeren en als uitgang de lussen sturen. De modellen verschillen op verschillende manieren: Expert systemen worden opgesteld op basis van kennis die bij experts aanwezig is en kan vertaald worden naar regels in de vorm van if-the rules. Zelflerende algoritmes, leren op basis van historische data het gewenste gedrag.

Verder verschillen de algoritmes in:

- Training data requirements: De hoeveelheid/kwaliteit en aard van de data die nodig is om het algoritme te ontwikkelen.
- #input parameters: Modellen verschillen in de hoeveelheid input data die ze kunnen verwerken.
- Noise gevoeligheid: De algoritmes verschillen in de hoeveelheid die ze kunnen verwerken én de hoeveelheid ruis die op de data aanwezig mag zijn.
- Transparantie/interpreteerbaarheid: een transparant algoritme laat zich door mensen lezen, zo kan de mens begrijpen hoe een bepaalde output berekend werd. Daar tegenover staan black-box algoritmes die vaak goede resultaten geven maar hun werking laat zich moeilijk interpreteren.
- Rekenlast inference: De rekenkracht die nodig is om de voorspellingen in real time te maken.
- Rekenlast trainen: De rekenkracht die nodig is om het model te trainen tijdens ontwikkeling.

Expert systemen

In expertsystemen gebruiken we domeinkennis en expertkennis om een model te bouwen dat de gemeten invoergegevens omzet in gewenste outputs. We gaan ervan uit dat we enig begrip hebben van de relaties tussen inputs en outputs en dat we ze kunnen coderen in regels of wiskundige functies. Expert systemen niet erg gesofisticeerd maar kunnen goede resultaten leveren. Expert systemen kunnen ook data driven zijn wanneer tijdens ontwikkeling een data analyse gebeurt op de historische data waarvan de inzichten omgezet worden in regels en functies in het model.

If-then rules

Regels worden geïmplementeerd op basis van expert kennis.

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenklast inference	Rekenlast trainen
If-then rules	expert kennis + validatie data	beperkt tot het menselijk overzicht over de input paramters	Noise ok in het geval dat men dit voorziet in tijdens implementatie	Hoog	n.a.	Laag

Belief networks

Belief networks kunnen opgesteld worden op basis van expert kennis of zelflerend zijn of een combinatie van beide voorstellen. Een verdere beschrijving wordt gegeven onder de machine learning modellen.

Machine learning models

Met machine learning methodes worden optimale relaties tussen input en gewenste output geleerd op basis van historische data. Een machine learning algoritme wordt tijdens de ontwikkeling 'getrained' op de historische data. Het wordt daarna in het veld gebruikt door real time data te verwerken en een voorspelling te maken van de gewenste data.

Zelflerende algoritmes kunnen slechts leren van de beschikbare data. Dit wil zeggen dat het belangrijk is dat het gewenste gedrag reeds in de historische data terug te vinden is of door een

expert kan aangeleverd worden. Indien dit niet het geval is het aangeraden expert systemen te ontwikkelen of alternatieven te onderzoeken.

Rule learning systems

Regels worden geleerd op basis van voorbeeld data waar gewenst gedrag vertoond wordt.

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenklast inference	Rekenlast trainen
Rule learning systems	minimaal	lage input dimensie toegelaten, hoge input dimensie toegelaten	Noise avers	Hoog	Laag	Laag

Decision trees

Beslissingsbomen worden geleerd op basis van voorbeeld data waar gewenst gedrag vertoond wordt.

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenklast inference	Rekenlast trainen
Decision trees	minimaal	lage input dimensie toegelaten, hoge input dimensie toegelaten	Noise avers	Hoog	Laag	Laag

Random forests

Verschillende beslissingsbomen worden geleerd. Bij inferentie hebben alle bomen een stem om te bepalen welk het uiteindelijke antwoord wordt(ensemble models). Forests zijne en

verbetering omdat ze minder label-noise gevoelig zijn dan enkelvoudige decision trees. Het leren gebeurt op basis van voorbeeld data waar gewenst gedrag vertoond wordt.

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenlast inference	Rekenlast trainen
Random forests	minimaal	lage input dimensie toegelaten, hoge input dimensie toegelaten	Beperkte noise ok	Matig	Laag	Laag

Support vector machines

Alle klassen/acties/output-labels worden gescheiden door lineaire of niet-lineaire scheidingsvlakken op basis van support vectors, dit zijn vectors bestaande uit de input variabelen die het meest bepalend zijn voor de scheiding tussen verschillende klassen. Het leren gebeurt op basis van voorbeeld data waar gewenst gedrag vertoond wordt.

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenlast inference	Rekenlast trainen
Support vector machines	extensieve data set	lage input dimensie toegelaten, hoge input dimensie toegelaten	Beperkte noise ok - Noise resistent in sommige gevallen.	Matig	Medium - Hoog	Medium - Hoog

Neurale netwerken

Er wordt een verband gezocht tussen de gewenste ingang en uitgang door een aaneenschakeling van bewerkingen die "neuronen" genoemd worden. Het leren gebeurt op basis van voorbeeld data waar gewenst gedrag vertoond wordt.

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenklaster inference	Rekenlast trainen
(shallow) neurale netwerken	extensieve data set	lage input dimensie toegelaten, hoge input dimensie toegelaten	Beperkte noise ok	Laag	medium	medium

(diepe) Neurale netwerken

feed-forward neurale netwerken en convolutionele neurale netwerken

Er wordt een verband gezocht tussen de gewenste ingang en uitgang door een aaneenschakeling van bewerkingen die "neuronen" genoemd worden. Diepe neurale netwerken vormen vele lagen van deze neuronen waardoor ze zeer complexe verbanden kunnen aanleren. Het leren gebeurt op basis van voorbeeld data waar gewenst gedrag vertoond wordt.

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenklaster inference	Rekenlast trainen
(deep) Neurale netwerken	big data	ideaal voor hoge input dimensie	Noise resistent	Black-box	Hoog	Hoog
feedforward neurale netwerken	big data	ideaal voor hoge input dimensie	Noise resistent	Black-box	Hoog	Hoog
recurrente neurale netwerken	big data	ideaal voor hoge input dimensie	Noise resistent	Black-box	Hoog	Hoog

Deze netwerken hebben een geheugen component in de neuronen waardoor makkelijk gewerkt kan worden met data die varieert in de tijd. Belangrijke gebeurtenissen in een vroeger tijdstip worden namelijk "onthouden in neuronen" om te gebruiken in de huidige output berekening.

Belief networks

Belief networks leren kansverdelingen op grafische weergaven van kansverdelingen. In deze grafische weergaven kunnen veronderstellingen over causaliteit vervat worden. Belief networks laten toe de kans te berekenen dat een bepaald scenario zich zal voordoen onder de kennis van andere variabelen. Het leren gebeurt op basis van voorbeeld data waar gewenst gedrag vertoond wordt. Expert kennis kan ook toegevoegd worden aan deze algoritmen.

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenlast inference	Rekenlast trainen
Belief networks	big data en/of expert kennis en/of statistische proceskennis	lage input dimensie toegelaten, hoge input dimensie toegelaten	Hangt af van implementatie	Hoog	Laag - Medium - Hoog	Medium - Hoog

Overzicht

Naam	Training data requirements	# input paramters	Noise gevoeligheid	Transparantie	Rekenlast inference	Rekenlast trainen
If-then rules	expert kennis + validatie data	bepert tot het menselijk overzicht over de input paramters	Noise ok in het geval dat men dit voorziet tijdens implementatie	Hoog	n.a.	Laag
Zelflerende algoritmes						

Rule learning systems	minimaal	lage input dimensie toegelaten, hoge input dimensie toegelaten	Noise avers	Hoog	Laag	Laag
Decision trees	minimaal	lage input dimensie toegelaten, hoge input dimensie toegelaten	Noise avers	Hoog	Laag	Laag
Random forests	minimaal	lage input dimensie toegelaten, hoge input dimensie toegelaten	Beperkte noise ok	Matig	Laag	Laag
Support vector machines	extensieve data set	lage input dimensie toegelaten, hoge input dimensie toegelaten	Beperkte noise ok - Noise resistent in sommige gevallen.	Matig	Medium - Hoog	Medium - Hoog
(shallow) neurale netwerken	extensieve data set	lage input dimensie toegelaten, hoge input dimensie toegelaten	Beperkte noise ok	Laag	medium	medium
(deep) Neurale netwerken	big data	ideaal voor hoge input dimensie	Noise resistent	Black-box	Hoog	Hoog
feedforward neurale netwerken	big data	ideaal voor hoge input dimensie	Noise resistent	Black-box	Hoog	Hoog
recurrente neurale netwerken	big data	ideaal voor hoge input dimensie	Noise resistent	Black-box	Hoog	Hoog

reinforcement learning	big data - leert al doende	lage input dimensie toegelaten, hoge input dimensie toegelaten		Black-box	Hoog	Hoog
Q - learning	big data - leert al doende	lage input dimensie toegelaten, hoge input dimensie toegelaten	Beperkte noise ok	Black-box	Hoog	Hoog
deep reinforcement learning	big data - leert al doende	lage input dimensie toegelaten, hoge input dimensie toegelaten	Noise resistant	Black-box	Hoog	Hoog
Belief networks	big data en/of expert kennis en/of statistische proceskennis	lage input dimensie toegelaten, hoge input dimensie toegelaten	Depends on implementation	Hoog	Laag - Medium - Hoog	Medium - Hoog