

De Vlaamse Scriptieprijs 2015

# Een monitoring raamwerk voor een soft-sensor

**Seppe DUWÉ**

Promotoren: Jeroen Boydens

Co-promotoren: David Verstraeten  
Jan Verhasselt

Masterproef ingediend tot het behalen van  
de graad van master of Science in de  
industriële wetenschappen:  
elektronica-ICT ICT

Mededeling

Deze eindverhandeling was een examen. De tijdens de verdediging geformuleerde opmerkingen werden niet opgenomen.

## Het woord vooraf

Mijn opleiding Master in de industriële wetenschappen Elektronica-ICT met afstudeerrichting ICT (4MI2EI) vereist voor het behalen van mijn diploma dat ik een stage doorloop met daaraan gekoppeld het volbrengen van een masterproef. Mijn stage duurde 6 weken, nl: van 2 februari tot 20 maart 2015.

Bij het opstellen van deze masterproef, die een leerrijke uitdaging vormde om zelfstandigheid te verwerven bij het uitvoeren en het rapporteren van wetenschappelijk onderzoek, hebben tal van personen me, al dan niet bewust, op de meest diverse wijze ondersteuning geboden. Ik wens hen dan ook een uitdrukkelijke vermelding te geven.

In eerste instantie gaat mijn oprechte dank uit naar Dr. Ir. David Verstraeten, Dr. Ir. Ing. Jan Verhasselt (buitenpromotoren) en Prof. dr. ing. J. Boydens (binnenpromotor) voor hun deskundige begeleiding. Het was een uiterst aangename ervaring om met hen samen te werken. Allen stonden tijdens het afgelopen jaar steeds klaar met heldere en accurate raadgevingen bij de inhoudelijke en vormelijke aspecten van deze masterproef.

Evenmin mag het belang niet onderschat worden van de morele steun die ik mocht ervaren tijdens de gehele duur van mijn opleiding. Vooral mijn ouders, die me de kansen en het vertrouwen hebben gegeven om deze opleiding tot een goed einde te brengen en me op de zware momenten gesterkt hebben met doorzettingsvermogen, ben ik veel dank verschuldigd.

Tot slot hoop ik nog dat u, beste lezer, bij het doornemen van deze masterproef hetzelfde enthousiasme ervaart dat mij al die tijd heeft aangezet om dit document succesvol tot stand te brengen. Hopelijk mag u dit als een verrijking beschouwen.

## Abstract

### Nederlands

Wanneer een productiebedrijf beschikt over een aanzienlijk aantal sensoren, waarvan enkele fysieke sensoren in het productieproces (soms) onbetrouwbaar, duur of tijdrovend zijn, kan er gebruik gemaakt worden van soft-sensors. Deze zijn ook bekend als surrogate, proxy of virtuele sensoren. Soft-sensors zijn computermodellen die de output van een fysieke sensor voorspellen. Dit gebeurt op basis van gemakkelijkere en/of vakere metingen elders in het proces. Deze computermodellen worden opgesteld op basis van historische data met behulp van machine learning technieken.

Een raamwerk voor een virtuele sensor wordt in de Python programmeertaal geïmplementeerd, deze beschikt over een monitoring- en alarmsysteem voor industriële processen en communiceren via “OLE (object linking and embedding) for Process Control” (OPC). Het systeem controleert de toestand en de prestaties van de soft-sensor model(len) en hun ingangen. Bij afwijkingen, gebaseerd op verschillende detectie algoritmen, produceert het configureerbare waarschuwingen zoals SMS, e-mail of HipChat. Dit laat gebruikers (bv. procesingenieurs) toe om verschillende instellingen te beheren (wie welke alarmen krijgt, drempelwaarden, ...) waarna er samenvattingen en gedetailleerde rapporten kunnen geraadpleegd worden.

In een ontwikkelomgeving is er meestal geen toegang tot een OPC-server in productie. Alsook geen mogelijkheid om fouten of fictieve data te injecteren. Daarom is er een virtuele OPC-server mock-up geïmplementeerd. Hiermee worden op deterministische wijze de verschillende detectie algoritmen en functionaliteiten getest.

**Sleutelwoorden:** soft-sensor, virtual-sensor, monitor, raamwerk, notificatie, OPC, afwijkingen, Python

## English

A production company can have a significant number of sensors. However, there is a possibility that some of the physical sensors in the production process are unreliable, expensive or time consuming. Therefore, the soft sensor, also known as surrogate, proxy or virtual sensor, is the right solution. Soft sensors are computer models that calculate estimates of physical sensor measurements or lab measurements, which are based on other variables that can be more easily and/or more frequently measured elsewhere in the process. These computer models are established based on historical data using machine learning techniques.

A framework for a virtual sensor is implemented in the Python programming language, which has a monitoring and alarm system for industrial processes and communicates through “OLE (object linking and embedding) for Process Control” (OPC). This system monitors the state and performance of the soft sensor model(s) and their inputs. If differences exist, based on detection algorithms, the system produces configurable alerts such as SMS, email and Hipchat. This allows users (e.g. Process engineers) to manage different settings (who gets what alarms, thresholds, ...). A summary and detailed reports are available as well.

In a development environment, there is usually no access to an OPC-server into production. Moreover, there is no ability to inject faults or fictitious data. Therefore, a virtual mock-up OPC-server is implemented, which makes it possible to test the various detection algorithms and functionality in a deterministic manner.

**Keywords:** soft sensor, virtual sensor, monitor, framework, notification, OPC, anomaly, Python

## Inhoudsopgave

Het woord vooraf .....	III
Abstract .....	IV
Nederlands .....	IV
English .....	V
Inhoudsopgave .....	VI
Lijst met illustraties .....	VII
Gebruikte afkortingen en symbolen.....	VIII
1 Inleiding .....	1
1.1 Bedrijfsinformatie.....	2
1.2 Methodiek literatuurstudie .....	2
1.3 Soft-sensor.....	3
2 Onderzoeksvraag.....	5
3 Python .....	6
3.1 Tools .....	6
3.2 Ontwikkelomgeving.....	7
4 OPC-server.....	9
4.1 OPC.....	9
4.2 Server mock-up .....	9
5 Monitor.....	11
5.1 OPC-cliënt.....	11
5.2 Anomaliedetectie .....	11
5.3 Notificatie/Alarm.....	19
6 Webapplicatie.....	20
6.1 Raamwerk.....	20
6.2 Lay-out raamwerk .....	24
6.3 I18n en L10n .....	24
6.4 Plot.....	25
6.5 Functies .....	28
7 Databank .....	30
7.1 ORM.....	30
7.2 Structuur.....	31
Besluit.....	34
Bijlagen .....	35
Bibliografie .....	36

## Lijst met illustraties

Figuur 1 Logo Yazzoom.....	2
Figuur 2 Soft-sensor schema .....	3
Figuur 3 Soft-sensor proces .....	4
Figuur 4 Papierproductie.....	4
Figuur 5 Soft-sensor nauwkeurigheid .....	4
Figuur 6 Schema opdracht.....	5
Figuur 7 Pylint.....	7
Figuur 8 PyDev IDE .....	8
Figuur 9 OPC-schema .....	9
Figuur 10 Uitschieter .....	12
Figuur 11 Grafische weergave anomalie .....	12
Figuur 12 Statistische weergave anomalie.....	12
Figuur 13 Overzicht Anomalie Detectie.....	13
Figuur 14 Point Anomalies .....	13
Figuur 15 Contextual Anomalies .....	13
Figuur 16 Collective Anomalies .....	13
Figuur 17 Classificatie anomalie detectie.....	14
Figuur 18 Poedit .....	24
Figuur 19 Webapplicatie Engels .....	25
Figuur 20 Webapplicatie Nederlands.....	25
Figuur 21 Voorbeeld Matplotlib .....	27
Figuur 22 Profile .....	28
Figuur 23 Bevestig e-mail .....	28
Figuur 24 E-mail notificatie .....	28
Figuur 25 Hipchat notificatie .....	28
Figuur 26 Algorithm notificatie .....	28
Figuur 27 Settings algorithm .....	29
Figuur 28 Select tags.....	29
Figuur 29 Statusmonitor.....	29
Figuur 30 Dashboard .....	29
Figuur 31 Structuur databank.....	31



## Gebruikte afkortingen en symbolen

AJAX	
Asynchronous JavaScript And XML .....	22
API	
Application programming interface.....	22, 23, 26
CSS	
Cascading Style Sheet .....	8, 24, 26
DCS	
Distributed control system .....	1
HMI	
Human-Machine Interface.....	9
HTML	
HyperText Markup Language.....	8, 22, 24
HTTP	
Hypertext Transfer Protocol .....	22, 23, 25
IDE	
Integrated development environment .....	8
OLE	
Object Linking and Embedding .....	9
OPC	
OLE for Process Control .....	IV, 1, 5, 9, 11, 28, 31
ORM	
Object relationele mapper.....	20, 21, 22, 30
PLC	
Programmable logic controller .....	1
R&D	
Research and Development.....	2
SMS	
Short message service .....	IV, 1, 28, 32
SQL	
Structured Query Language.....	22, 30
URL	
Uniform Resource Locator.....	20
WSGI	
Web Server Gateway Interface.....	22, 23
XSS	
Cross-site scripting.....	22

## 1 Inleiding

Bij mijn zoektocht naar een geschikte masterproef en daaraan gekoppelde stage werd mijn interesse gewekt door het bedrijf Yazzoom. Zowel door de opdracht alsook in een aantal diensten die Yazzoom aanbiedt, zoals productinnovatie, procesverbetering en data-analyse. Dit door middel van o.a. machinaal leren en computer modellering.

Wanneer een productiebedrijf beschikt over een aanzienlijk aantal sensoren waarvan enkele fysieke sensoren in het productieproces (soms) onbetrouwbaar, duur of tijdrovend zijn kan er gebruik gemaakt worden van een gespecialiseerde tool suite. Yasense is voor soft-sensors ontwikkeld door Yazzoom. Soft-sensors ook bekend als surrogate sensors, proxy sensors of virtuele sensoren zijn computermodellen die de output van een fysieke sensor voorspellen. Het voorspellen gebeurt op basis van gemakkelijkere en/of vakere metingen elders in het proces. Deze computermodellen worden opgesteld met historische data en behulp van machine learning technieken.

Deze tool kan antwoorden bieden op vragen: “Zijn enkele van de fysieke sensoren in uw productieproces (soms) onbetrouwbaar? Moet u vaak dure en tijdrovende labmetingen uitvoeren? Duurt het lang voordat bepaalde kwaliteitsaspecten van geproduceerde producten kan evalueren? Wilt u een doorlopende schatting van een hoeveelheid waarvoor geen (betaalbare) fysieke sensor bestaan? [1] Waarschuwingen ontvangen bij een abnormaliteit? Overzichtelijke samenvattingen en gedetailleerde rapporten bekomen van het productieproces?” De soft-sensor zal hier een goede oplossing bieden samen met de Yasense (monitor) tool.

YASENSE is een productfamilie die bestaat uit drie verschillende types:

Yasense Development: Een tool suite om de gegevens te optimaliseren (importeren, schoonmaken, filteren, annotatie, etc.), soft-sensor testdata (verschillende soorten machine learning algoritmen om op basis van deze data computer modellen te leren, alsook *whitebox* computermodellen) en soft-sensor validatie.

Yasense Run-time: Leest het proces en andere gegevens uit die dienen als input voor de soft-sensormodel (via OPC), berekent de soft-sensor uitgang(en) en publiceert deze op een computernetwerk (gewoonlijk naar een DCS / PLC).

Yasense Monitor: Controleert de toestand en de prestaties van de Yasense run-time, de soft-sensor model(s) en hun ingangen. Het produceert configureerbare waarschuwingen (SMS of e-mail), samenvattingen en gedetailleerde rapporten.

De masterproef vormt de basis voor de Yasense Monitor tool.

*Implementatie van web front-end voor een monitoring- en alarmsysteem voor een zogenaamde virtuele sensor voor industriële processen.*

## 1.1 Bedrijfsinformatie

Het bedrijf Yazzoom, actief in EMEA (Europa, Midden-Oosten en Afrika), met hoofdzetel in Gent, levert software en R&D-diensten voor procesverbetering. Ze bezitten een uitgebreide



Figuur 1 Logo Yazzoom

expertise op het gebied van digitale signaalverwerking, machinaal leren, computer modellering, software ontwikkeling, data analyse en regeltechniek. Deze zijn nuttig voor het aanpakken van heel verschillende uitdagingen in totaal verschillende toepassingsdomeinen. Een overzicht hieronder weergegeven:

Proces Verbetering: Software, regeltechniek en computermodellen voor:

- Verlagen van inspanning, energie en/of materialen
- Verkorten doorlooptijd
- Creëren van nieuwe mogelijkheden
- Oplossen van kwaliteitsproblemen
- Verhogen van proces stabiliteit
- Verhogen van de productie hoeveelheid

Product Innovatie: Softwareontwikkeling en computersimulatie voor:

- Ontwerp en optimalisatie van producten
- Verwerken van sensordata
- Verhogen van de "intelligentie" van toestellen en machines
- Creëren van nieuwe mogelijkheden
- Automatisatie van taken

Analytics: Statistiek, data analyse en machinaal leren voor:

- Extractie van nuttige informatie uit massa's data
- Verbeteren van beslissingsprocessen
- Ontwerp van efficiënte experimenten
- Opsporen van fraude en inefficiënties
- Evalueren van de kwaliteit van toestellen en methodes
- Voorspellen van verkoop
- Efficiëntie verhogen van marketing campagnes

De partners zijn Jan Verhasselt (Zaakvoerder) en David Verstraeten (Engineering). Jan Verhasselt een Industrieel Ingenieur Automatisering, Burgerlijk Elektrotechnisch Ingenieur Natuurkunde en Doctor in de Toegepaste Wetenschappen met specialisatie automatische spraakherkenning. David Verstraeten is Burgerlijk Ingenieur Computerwetenschappen en Doctor in de Toegepaste Wetenschappen met specialisatie machinelere en i.h.b. reservoir computing en neurale netwerken.

## 1.2 Methodiek literatuurstudie

Voor het verwerven van de nodige informatie werden relevante *papers* en boeken gezocht in de bibliotheek en de aangeboden databanken via Limo. Tevens is handmatig gericht gezocht via Google (Scholar) naar ontbrekende literatuur. Alsook werden door mijn buitenpromotor enkele literatuurverwijzingen gegeven.

De lectuur werd geselecteerd op de aanwezigheid van soft-sensor, anomaliedetectie of Python in de titel of het abstract. De trefwoorden werden afzonderlijk en gecombineerd gebruikt. Ook werd gebruik gemaakt van truncate technieken om variaties op de trefwoorden te gebruiken. Daarvoor werd de meest relevante lectuur op basis van het abstracte gekozen.

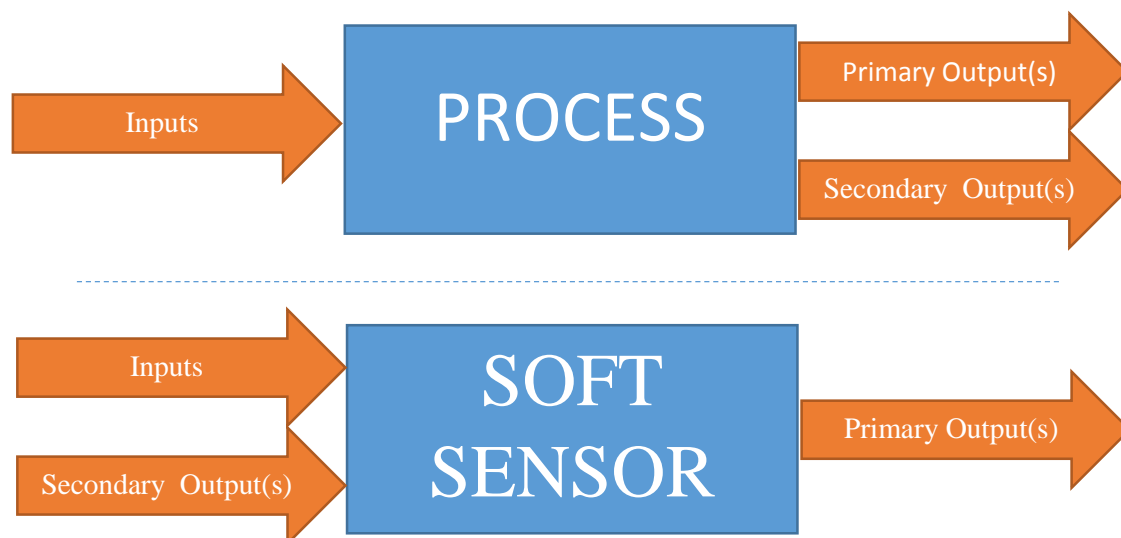
### 1.3 Soft-sensor

Soft-sensors zijn een waardevolle oplossing voor de nadelen die klassieke meetinstrumenten bezitten, zoals het gebrek aan geschikte, betrouwbare, betaalbare en vaak on-line instrumentatie. Laboratoriummetingen gebeuren zelden en onregelmatig, wat vertraging veroorzaakt door de analyse die moet gebeuren (bv. labmeting van papier, vocht- en asgehalte zie 1.3.3). Hier is ook dure apparatuur voor nodig wat een hoge operationele kost met zich meebrengt. On-line sensoren kunnen last hebben van lange meetvertragingen (bv. gaschromatograaf) en zijn niet altijd betrouwbaar. Ze kunnen verkeerde metingen in de eerste minuten produceren na het opstarten of last hebben van *drifts*, condensatie en vervuiling. Fysieke sensoren en proces analyzers hebben meer onderhoud nodig en kunnen van tijd tot tijd foutieve informatie geven.

Door gebruik te maken van de soft-sensor is er een snellere beschikbaarheid van de informatie over het product en/of proceskwaliteit, waardoor er sneller gereageerd kan worden. Er wordt een betere procesbeheersing verkregen, dat als resultaat besparing op energie, grondstof en een verhoogde veiligheid geeft. Een betere kwaliteitscontrole met een betere consistente productie als resultaat.

#### 1.3.1 Definitie

Soft-sensoren zijn computermodellen. Ze berekenen één of meer outputvariabelen op basis van interne modelparameters en inputvariabelen. Deze inputvariabelen zijn waarden die gemakkelijker worden gemeten dan de gewenste uitvoerwaarde(n). Bij soft-sensoren voor productieprocessen, kunnen de soft-sensor ingangsvariabelen bestaan uit ten eerste proces-ingangen waar de invoer stroomt naar het productieproces als ruwe data eigenschappen en waarden. Maar ook als proces kenmerken zoals druk, snelheid, temperatuur of andere metingen. Ten tweede uit proces-uitgangen die gemakkelijker worden gemeten zoals bepaalde producteigenschappen (Figuur 2).



Figuur 2 Soft-sensor schema

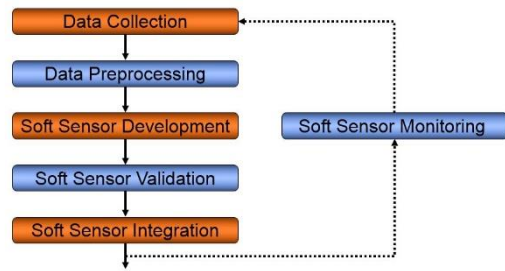
De proces-uitgangen zijn gemakkelijker on-line te meten en worden vaak secundaire proces-uitgangen genoemd. Degene die moeilijker te meten zijn, zijn net die procesuitgangen waarvoor interesse is. Het zijn de zogenaamde primaire procesuitgangen.

In een soft-sensor ontwikkelingsproject kunnen zogenaamde "*white box*" modellen gebouwd worden, evenals "*black box*" modellen of een samenvoeging van de twee ("*gray box*" model). In *white box* modellen wordt het model gemaakt op basis van een fysisch/chemisch begrip van het proces en de vertaling in wiskundige formules die worden berekend door de computer. Aan de andere kant worden in *black box* modellen machine learning algoritmen toegepast om het model op basis van historische gegevens te genereren. [2]

### 1.3.2 Ontwikkeling en onderhoud

Een project om een soft-sensor te creëren en te integreren in een productiebeheersysteem of procesbesturingssysteem, volgt een aantal typische stappen zoals weergegeven in Figuur 3.

Omdat een productieproces in de loop der tijd evolueert, kan het zich anders gaan gedragen dan op het moment dat de oorspronkelijke soft-sensor gemaakt is. Daarom zullen continu de prestaties van de soft-sensor en de kenmerken van de input geobserveerd worden. Er zal een configureerbare waarschuwing gegeven worden als er iets lijkt mis te gaan.

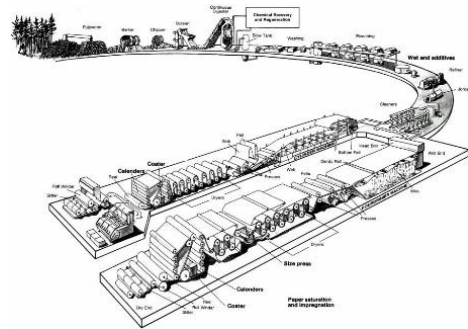


Figuur 3 Soft-sensor process ©Yazzoom

### 1.3.3 Voorbeeld voorspellen van het vochtgehalte van papier

De soft-sensor kan op vele plaatsen ingezet worden zoals bijvoorbeeld bij het voorspellen van het vochtgehalte van papier.

Het vochtgehalte van het papier, geproduceerd in een papiermachine, is een zeer belangrijk productkenmerk dat binnen een gedefinieerde bereik moet liggen.

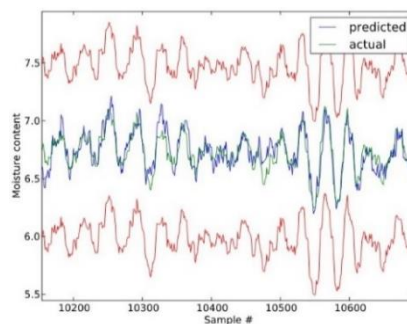


Figuur 4 Papierproductie

Het vochtgehalte wordt gemeten door een meetapparaat (typisch gebaseerd op infraroodmetingen) die heen en weer over het geproduceerde papier scant. Het is een bekend probleem dat in de eerste paar minuten na het starten van de machine, de scanner onbetrouwbare metingen geeft. Een typische reden hiervan is dat water condenseert op het apparaat in stilstand. Ondanks alle inspanningen om dat probleem mechanisch/fysisch op te lossen, is het nog steeds een vaak voorkomend probleem in de pulp- en papierindustrie.

Een oplossing is een soft-sensor te plaatsen die de relatie geleerd heeft tussen tientallen on-line gemeten papier- en machine kenmerken (snelheid, stoomdruk, ...) en het vochtgehalte opmeten op het moment dat het meetapparaat betrouwbaar is. In de eerste paar minuten na het opstarten, wordt de stoom om het papier te drogen geregeld op basis van de vochtinschatting van de soft-sensor. Dit in plaats van gegevens gebaseerd op de fysieke vochtsensor, waardoor minderwaardig kwaliteitspapier gereduceerd is.

Figuur 5 toont aan dat de nauwkeurigheid van het voorspelde vocht op basis van de soft-sensor goed de lijn van de werkelijke vochtmeting door de fysieke scanner benadert. De meting is genomen wanneer de fysieke vochtscanner nauwkeurige metingen produceert.



Figuur 5 Soft-sensor nauwkeurigheid ©Yazzoom

## 2 Onderzoeksvraag

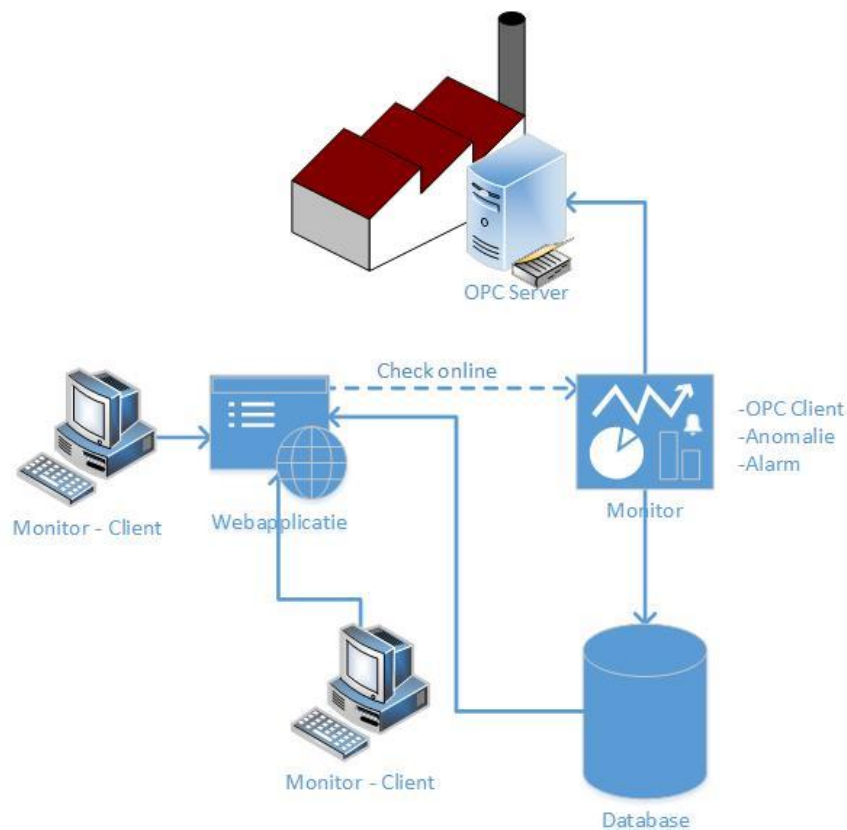
De onderzoeksvraag gaat over de implementatie van een web front-end voor een monitoring- en alarmsysteem voor een virtuele sensor voor industriële processen.

Het onderzoek is opgesplitst in 3 delen zoals weergegeven in Figuur 6.

De communicatie voor industriële processen verloopt via OPC. Er is nood aan een OPC-server (hoofdstuk 4) waar de sensorwaarden live worden opgevraagd. In een ontwikkelomgeving is er meestal geen toegang tot een OPC-server in productie. Alsook geen mogelijkheid om fouten of fictieve data te injecteren. Daarom is er een virtuele OPC-server mock-up geïmplementeerd. Hierdoor kunnen op deterministische wijze de verschillende detectie algoritmen en functionaliteiten getest worden.

Het background monitoring proces (hoofdstuk 5) zal de in en uitgangen van een virtuele sensor controleren op afwijkingen, gebaseerd op verschillende detectie algoritmen. Er zal worden nagegaan wat anomalieën of afwijkingen zijn, welke detectietechnieken er bestaan en hoe deze te implementeren in Python. Alsook onderzoek naar verschillende notificatie platformen waarna detectie van een afwijking een gepaste notificatie wordt verstuurd.

Voor de web front-end (hoofdstuk 6) waar het de gebruikers (bv. procesingenieurs) toelaat om de instellingen te beheren (wie welke alarmen krijgt, drempelwaarden e.d.) zullen verschillende Python web raamwerken worden vergeleken op basis van uitbreidbaarheid, eenvoud, prestatie en beschikbare plug-ins. Waarna de meest geschikte wordt geïmplementeerd. Alsook voor het plotten van de gelogde data zal er een keuze gemaakt worden tussen de vele beschikbare Python visualisatiebibliotheken.



Figuur 6 Schema opdracht

## 3 Python

De masterproef bestaat uit een webapplicatie, anomalie detectiemethoden en machine learning technieken. Er is dus nood aan wetenschappelijke bibliotheken en een webserver. Hiervoor is Python zeer geschikt. Python is open source en volledig gratis, zelfs voor commercieel gebruik. Het omvat standaard een indrukwekkend aantal algemene of meer gespecialiseerde (wetenschappelijke) bibliotheken. Doordat veel van de bibliotheken gratis aan anderen ter beschikking worden gesteld, groeien de mogelijkheden voortdurend. Python wordt zo een programmeertaal die voor snelle ontwikkeling van een nieuwe applicatie kan worden gebruikt, zonder dat de daarbij geproduceerde code onleesbaar wordt. Het is *multiplatform* en kan worden uitgevoerd op Windows, Mac OS en Linux.

Er zijn 2 Python versies beschikbaar. Python 2.x is *legacy* (oudere) en Python 3.x is onder actieve ontwikkeling. Beide versies zijn niet compatibel met elkaar. Oudere plug-ins die niet geüpdatet zijn kunnen niet worden uitgevoerd door de nieuwe versie 3. Hierdoor is Python versie 2.7 gekozen om de tool “Yasense Monitor” te ontwikkelen. [3]

### 3.1 Tools

De gebruikte tools om deze masterproef te verwezenlijken worden hieronder kort beschreven.

#### 3.1.1 Mercurial

Mercurial is een vrij *multiplatform* gedistribueerd versiebeheersysteem. Een alternatief van het git protocol. Het programma zelf heet Hg, een verwijzing naar het scheikundige element kwik (mercury in het Engels). <http://mercurial.selenic.com>

#### 3.1.2 Anaconda

Anaconda is een gratis enterprise-ready Python distributie voor grootschalige dataverwerking, voorspellende analyses en wetenschappelijke berekeningen, dat gericht is op het vereenvoudigen van package management en implementatie. Het package management systeem Conda wordt hiervoor gebruikt.

Conda is een applicatie voor het vinden en installeren van software pakketten. Een conda pakket is een binair tar-bestand met system-level bibliotheken, Python modules, uitvoerbare programma's of andere componenten. Conda houdt afhankelijkheden bij tussen pakketten en platform specifieke kenmerken. Hierdoor is het eenvoudig om werkomgevingen te creëren uit verschillende sets van pakketten.

Het verschil met pip, wat ook een package management systeem is, is dat Conda afhankelijkheden buiten de Python pakketten (*binaries*) kan installeren. <https://store.continuum.io/cshop/anaconda>

#### 3.1.3 Virtualenv

Virtualenv is een hulpmiddel om een geïsoleerde Python omgeving te creëren, omdat verschillende versies van modules kunnen conflicteren met andere Python programma's, die op diezelfde machines draaien. Het creëert een omgeving die zijn eigen installatie directories heeft en daardoor geen bibliotheken deelt met andere virtualenv omgevingen en eventueel geen toegang tot de globale geïnstalleerde bibliotheken heeft. <https://virtualenv.pypa.io>

Virtualenvwrapper is een uitbreiding op de virtualenv tool. De uitbreidingen zijn onder andere *wrappers* voor het maken en verwijderen van virtuele omgevingen en om de ontwikkeling workflow te beheren. Hierdoor wordt het makkelijker om tegelijkertijd te werken aan meer dan één project zonder conflicten te introduceren in de afhankelijkheden. <https://virtualenvwrapper.readthedocs.org>

### 3.1.4 Pylint

Pylint is een statische codechecker, wat betekent dat de code kan analyseren zonder daadwerkelijk uit te voeren. Pylint controleert op fouten, probeert de standaard PEP8 codering (document die de codering conventies geeft voor de Python-code) af te dwingen (zie Figuur 7). <http://docs.pylint.org>

```

PyLintGit
***** Module toolbox.opc
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:160:0: [W0511(fixme), ] TODO Performance?
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:14:0: [C0325(superfluous-parens), ] Unnecessary parens after 'print' keyword
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:119:0: [C0303(trailing-whitespace), ] Trailing whitespace
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:149:0: [C0301(line-too-long), ] Line too long (109/100)
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:179:0: [C0326(bad-whitespace), ] Exactly one space required after comma
def list(self, tags=None):
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:294:0: [W1402(anomalous-unicode-escape-in-string), ] Anomalous Unicode escape
in byte string: '\U'. String constant might be missing an r or u prefix.
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:18:0: [C0103(invalid-name), ] Invalid constant name "OPCErrors"
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:37:0: [C0111(missing-docstring), OPCError] Missing class docstring
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:49:0: [R0903(too-few-public-methods), Command] Too few public methods (1/2)
C:\Users\Seppe\toolbox.hg\toolbox\opc.py:6:0: [W0611(unused-import), ] Unused import pdb
    
```

Figuur 7 Pylint

### 3.2 Ontwikkelomgeving

Er bestaan veel verschillende ontwikkelomgevingen die gebruikt worden met Python. In Tabel 1 een overzicht samen met hun functies.

Tabel 1 Vergelijking ontwikkelomgevingen

	Cross Platform*	Commercial / Free	Auto Code Completion	Multi-Language Support	Integrated Python Debugging	Error Markup	Source Control Integration	Smart Indent	Bracket Matching	Line Numbering	UML Editing / Viewing	Code Folding	Code Templates	Unit Testing
BlackAdder	+	C						+				+		
BlueFish	L													
ConTEXT	W	C												
DABO	+													
DreamPie		F	+					+						
Dr.P+thon		F				+								
Editra	+	F	+	+			+	+	+	+		+		
Emacs	+	F	+	+	+	+	+	+	+	+	+	+	+	+
Eric Ide	+	F	+		+	+		+		+		+		+
E-Texteditor	W													
Geany	+	F	+	+				+	+	+		+		
Gedit	+	F	+	+				+	+	+			+	
Idle	+	F	+											
JEdit	+	F		+					+	+		+		
KDevelop	+	F		+			+	+	+	+		+		
Komodo	+	C/F	+	+	+	+	+	+	+	+		+	+	+
NetBeans	+	F	+	+	+		+	+	+	+	+	+	+	+
NotePad++	W	F		+				+		+				
Pfaiide	W	C	+	+				+	+	+		+	+	
PIDA	LW	F	+	+				+	+	+		+		



PTVS	W	F	+	+	+	+	+	+	+	+	+	+		
PyCharm	+	C	+	+	+		+	+	+	+		+		+
PyDev(Eclipse)	+	F	+	+	+	+	+	+	+	+	+	+	+	+
Pyscripter	W	F	+		+	+		+		+			+	+
PythonWin	W	F	+		+			+	+			+		
SciTE	+	F		+		+			+	+		+	+	
ScriptDev	W	C	+	+	+	+		+	+	+		+	+	
SPE		F	+									+		
Spyder	+	F	+		+	+		+	+	+				
Sublime Text	+	C/F	+	+				+	+	+			+	
TextMate	M			+				+	+	+		+	+	
UliPad	+	F	+	+	+			+	+				+	+
Vim	+	F	+	+	+	+	+	+	+	+		+	+	+
Wing Ide	+	C	+	+	+	+	+	+	+	+		+	+	+
Zeus	W	C					+	+	+	+		+	+	

(L – Linux W – Windows M – Mac) Gebaseerd op <https://wiki.python.org/moin/PythonEditors>

Enkele ontwikkelomgevingen die over een plug-in systeem en unit testen beschikken, onder actieve ontwikkeling en Windows compatibel zijn, worden verder in detail besproken.

### 3.2.1 PyCharm (getest)

PyCharm is een van de meest populaire Python IDE's. Het zit vol met functies zoals code *completion*, code analyse, code navigatie, goede ondersteuning van Django, Javascript, HTML en CSS, goede debugger en nog veel meer functionaliteiten. ([www.jetbrains.com/pycharm](http://www.jetbrains.com/pycharm))

### 3.2.2 Wing IDE

Wing IDE is een solide IDE uit Wingware sinds 1999. Het heeft veel geavanceerde functies, zoals een debugger, code *intelligence* en kan verder worden uitgebreid met Python scripts. Wing IDE's debugger bevat veel functionaliteit en is geoptimaliseerd voor het debuggen van Django templates. Het biedt ondersteuning voor matplotlib waar de plots automatisch worden bijgewerkt. Wing IDE ondersteunt Django, Plone, Pyramid, Google App Engine en vele anderen. ([www.wingware.com](http://www.wingware.com))

### 3.2.3 PyDev (gekozen)

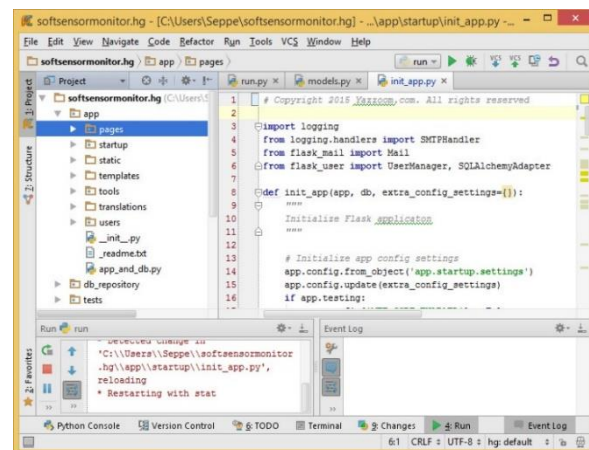
PyDev is een Python IDE voor Eclipse met Django ondersteuning, code *completion*, code analyse, navigatie, remote debuggen, interactieve console en nog veel meer. Soms zijn handelingen in Eclipse onnodig ingewikkeld. Bijvoorbeeld, de manier waarop code kan geïmporteerd worden in een project. ([www.pydev.org](http://www.pydev.org))

### 3.2.4 Komodo IDE (getest)

Komodo is een IDE voor dynamische talen zoals Python, PHP en Ruby. Komodo IDE is aangenaam om mee te werken en de configuraties zijn gemakkelijk te vinden en te veranderen. ([www.komodoide.com](http://www.komodoide.com))

### 3.2.5 Conclusie

PyCharm, Komodo IDE en PyDev werden getest waaruit PyCharm (Figuur 8) de meest geschikte was. Het heeft uitgebreidere Mercurial functionaliteit, code optimalisatie, *refactoring*, snellere code *completion*, toegang tot code documentatie, biedt ondersteuning voor Docutils en er bestaat een editie die gratis is.



Figuur 8 PyDev IDE

## 4 OPC-server

Eerst wordt het begrip OPC toegelicht waarna de mock-up server wordt besproken. Wat zijn de voordelen en waarom werd hiervoor gekozen?

### 4.1 OPC

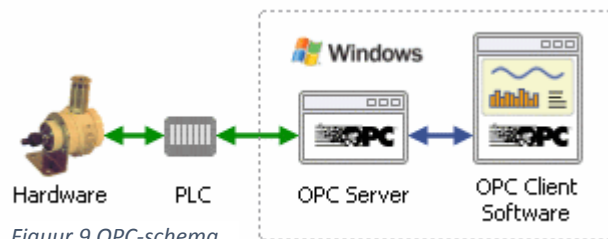
OPC staat voor “OLE voor *Process Control*”, waarin OLE staat voor “*Object Linking and Embedding*”, is een standaard voor gegevensuitwisseling tussen industriële automatiseringssystemen. Een OPC-connectie kan variëren tussen uitwisseling van actuele waarden en statussen van datapunten. Bijvoorbeeld de temperatuur, druk of gewicht, tot complete records met actuele alarmstatussen en historische procesgegevens.

Data wordt uitgewisseld op basis van OLE. De standaard is ontwikkeld door zo'n 270 leveranciers van hardware en software. Deze leveranciers zijn verenigd in de OPC Foundation. OPC is gebaseerd op verschillende Microsoft technologieën zoals DCOM (Distributed Component Object Model), COM en OLE. Alsook op XML en Internet. Het bestaat uit een aantal standaard interfaces, eigenschappen en methoden om data-uitwisseling mogelijk te maken in de procescontrole en fabricageprocessen. [4]

Bijzonder is dat behalve de waarde van zo'n datapunt ook nog de status (is de data goed, twijfelachtig of niet geldig) en de timestamp (moment van waarde meting) wordt uitgewisseld. [5]

Een voorbeeld:

```
opc.read('B_CONCENTR_6.B')
(19169, 'Good', '04/24/15 15:56:11')
```



Een eenvoudig scenario van een OPC-verbinding is een enkele server-cliënt-verbinding op een enkele computer, zoals afgebeeld in Figuur 9. De OPC-server is een softwareprogramma dat het hardware communicatieprotocol gebruikt door een PLC, omzet in het OPC-protocol. De OPC-softwarecliënt is een programma dat moet geconnecteerd worden op de hardware, zoals een HMI<sup>1</sup>. De OPC-cliënt maakt gebruik van de OPC-server om gegevens te verkrijgen of commando's te versturen naar de hardware.

### 4.2 Server mock-up

Aangezien er geen toegang is tot een effectieve OPC-server en er geen mogelijkheid is om fouten of fictieve data te injecteren om de functionaliteiten te testen, wordt een virtuele OPC-server gebruikt. Verscheidene virtuele OPC-servers bestaan zoals MatrikonOPC ([www.matrikonopc.com](http://www.matrikonopc.com)) maar deze zijn betalend, complexer (functionaliteiten die niet vereist zijn voor deze applicatie) en kunnen niet aangepast worden naar onze wensen. Daarom is er in het begin van de stage een model of mock-up ontworpen van een OPC-server.

Bij het instantiëren van de OPC-server mock-up wordt er gelogde data, in de vorm van pickle bestanden (mechanisme voor object serialisatie) of uit gegenereerde lijsten, ingelezen in een Pandas DataFrame structuur.

Pandas is een open source (BSD-licentie) Python bibliotheek voor datamanipulatie en analyse. Het biedt hoge prestaties, eenvoudig te gebruiken datastructuren en data-analyse-instrumenten voor het manipuleren van numerieke tabellen en tijdreeksen. Een DataFrame is een 2-dimensionale gelabelde

<sup>1</sup> Human-Machine Interface: een grafische interface die het mogelijk maakt om een persoon te laten communiceren met een regelsysteem. Het kan trends, alarmen, afbeeldingen of animaties bevatten.

datastructuur met kolommen van mogelijk verschillende types. Deze zijn gelijkaardig aan een spreadsheet of SQL tabel.

De data uit de bestanden komen van een reëel bedrijf dat gedurende enkele dagen gelogd is. Het is zo geïmplementeerd dat de tijd wanneer de OPC-server wordt aangemaakt overeenkomt met het oudste record en live meegaat met de gelogde data. Hierdoor wordt een identiek beeld verkregen als tijdens de logging in het bedrijf. Er is verdere ondersteuning voor de basis OPC functionaliteiten, zoals het lezen en schrijven van één of meerdere tags. Voor betere prestaties is het vaak nodig om tag-namen in een groep te plaatsen. Vervolgens kunnen herhaaldelijk de waarden van deze groep opgevraagd worden en later verwijderd worden om de resources terug vrij te geven. De status van de data kan aangepast worden om specifieke fouten te injecteren en de functionaliteiten te testen. Ook kan een lijst opgevraagd worden van de beschikbare OPC-tags.

Om de correctheid van deze functionaliteiten te garanderen zijn er verschillende testen geschreven. Het Python unit test raamwerk, soms aangeduid als "PyUnit" is een Python-versie van JUnit, door Kent Beck en Erich Gamma. JUnit is, op zijn beurt, een Java-versie van Kent's Smalltalk test raamwerk.

Unit testen is een herhaaldelijke activiteit die controleert of de individuele stukjes code (units) binnen een module of toepassing werken zoals verwacht. Bij unittesten zal voor iedere unit een of meerdere testen ontwikkeld worden. Hierbij worden dan verschillende testcases doorlopen. In het ideale geval zijn alle testcases onafhankelijk van andere tests. Dit maakt het vinden en oplossen van de problemen veel gemakkelijker [6]. De gebruikte IDE PyCharm ondersteunt de meest bekende test raamwerken zoals: Attest, Doctest, Nostest, py.test en de gebruikte Unittest. PyCharm heeft ook de mogelijkheid om bij elke wijziging in de broncode de testen automatisch te laten uitvoeren.

## 5 Monitor

Het background monitoring proces zal de ingangen en output van een virtuele sensor controleren op afwijkingen en hierop gepaste notificaties versturen. In dit hoofdstuk wordt eerst de OPC-cliënt beschreven waarmee de monitor zal communiceren met de OPC-server. Waarna anomaliedetectie wordt besproken en antwoorden worden gegeven op vragen als: Wat zijn juist anomalieën of afwijkingen? Welke detectietechnieken bestaan er? Hoe anomaliedetectie implementeren? Tenslotte worden verschillende notificatieplatformen besproken.

### 5.1 OPC-cliënt

Voor de OPC-cliënt is er gebruik gemaakt van OpenOPC, het is een gratis open source OPC bibliotheek ontworpen voor gebruik met Python. Hierover is een extra laag gemaakt met onze eigen functionaliteiten. Er is ondersteuning om te communiceren via opdrachtregel, Python bibliotheek en de mock-up test server.

Er werd eerst gedacht om de OPC-cliënt en mock-up server te laten communiceren over COM (Component Object Model). Dit is een software architectuur om de ontwikkeling van component-gebaseerde toepassingen te ondersteunen. Het bleek niet goed ondersteund te worden door Python en kon zodoende voor problemen zorgen. Na enkele dagen onderzoek is overwogen om dit weg te laten en rechtstreeks, via Python, met de mock-up bibliotheek te communiceren. [7]

### 5.2 Anomaliedetectie

Bij het inlezen van de fysische sensorgegevens waarop de soft-sensor zijn output baseert, is er een kans dat er foutieve waarden worden ingelezen. Deze fouten moeten opgevangen worden en weergegeven, zodat er gepast op kan gereageerd worden. Er zal dus een anomaliedetectie systeem geïmplementeerd worden. Hierdoor kunnen gegevens geïdentificeerd worden die niet voldoen aan het verwachte patroon of andere items in een dataset. Deze gegevens worden afwijkingen, uitschieters, uitzonderingen of anomalieën genoemd.

#### 5.2.1 Theorie

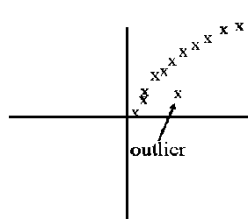
Het doel van anomaliedetectie is het vinden van alle objecten die verschillend zijn van andere objecten. Alsook een techniek om de analyse te verbeteren van typische gegevensobjecten.

Deze afwijkende objecten zijn in zeker zin uitzonderlijk zoals bv.: Ze liggen ver uit de buurt van andere datapunten (uitschieters). Ze hebben attribuutwaarden die significant afwijken van de verwachte of typische attribuutwaarden. Ze geven fouten in de data.

Anomaliedetectie is om verschillende redenen belangrijk. Ten eerste om problemen op te sporen, ten tweede om nieuwe fenomenen te detecteren en ten derde om ongewoon gedrag in data te ontdekken.

Voorbeelden van interessante toepassing voor anomaliedetectie:

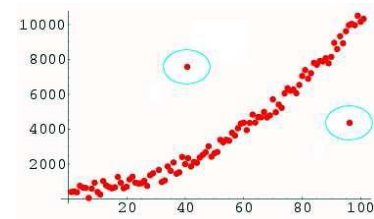
- Opsporing van fraude: op zoek naar patronen die afwijken van het typische koopgedrag
- *Intrusion Detection*: een geautomatiseerd systeem dat hackpogingen en voorkomens van ongeautoriseerde toegang tot een informatiesysteem of netwerk detecteert.
- Verstoring van het ecosysteem: proberen om gebeurtenissen zoals orkanen en overstromingen te voorspellen
- Volksgezondheid: gebruik van medische rapporten voor het stellen van diagnoses
- Geneeskunde: ongewone symptomen of testresultaten aan mogelijke gezondheidsproblemen te koppelen
- Soft-sensor: detecteren van afwijkende input data



Figuur 10 Uitschieter

Een anomalie is een afwijkend object (punt) dat merkkelijk verschilt van andere objecten (punten). In de statistiek wordt een uitschieter aanzien als data die relatief ver van de overige data verwijderd ligt, zie Figuur 10.

Hawkins' definitie van een uitschieter: "An outlier is an observation that differs so much from other observation as to arouse suspicion that it was generated by a different mechanism"

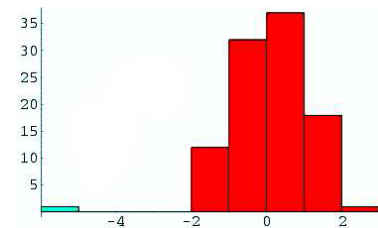


Figuur 11 Grafische weergave

Moore and McCabe: "An outlier is an observation that lies outside the overall pattern of a distribution".

#### 5.2.1.1 Oorzaken van anomalie

Veel voorkomende oorzaken van anomalie zijn ten eerste gegevens uit verschillende klassen: objecten verschillen omdat ze van een ander type klasse zijn. Ten tweede natuurlijke variatie: datasets gemodelleerd door statistische verdelingen (Figuur 12) zijn toegelaten variaties in de gegevens. Ten derde meetgegevens: fouten bij het verzamelen van gegevens of tijdens het meetproces.



Figuur 12 Statistische weergave anomalie

#### 5.2.1.2 Gegevens classificatie

De labels die behoren bij de gegevens duiden aan of deze specifieke data normaal of abnormaal is. Er moet wel worden opgemerkt dat het verkrijgen van gelabelde gegevens, die accuraat alsook representatief zijn voor alle soorten gedragingen, vaak onbetaalbaar is. Labeling wordt vaak handmatig gedaan door een menselijke expert waardoor er dus een aanzienlijke inspanning vereist is om de gelabelde training dataset te verkrijgen. Typisch, het krijgen van een label-set van afwijkende gegevens waarin alle mogelijke soorten afwijkend gedrag zit, is moeilijker dan het krijgen van labels voor normaal gedrag. Bovendien is het abnormaal gedrag vaak dynamisch van aard. Zo kunnen bijvoorbeeld nieuwe types afwijkingen ontstaan waarvoor geen gelabelde trainingsgegevens bestaan.

Op basis van de mate waarin de labels beschikbaar zijn, kunnen anomalie detectietechnieken in één van de volgende drie manieren functioneren:

#### 1. Supervised

Er is nood aan traininggegevens met zowel abnormale als normale objecten.

#### 2. Unsupervised

Detecteren van afwijkingen op niet-gelabelde testgegevens in de veronderstelling dat de meerderheid van de gevallen in de dataset normaal zijn door te zoeken naar gevallen die het minst bij de rest van de dataset passen. Er wordt een score toegekend aan elk exemplaar (mate van afwijking).

Probleem met vele soortgelijke afwijkingen is dat het als normaal kan worden geclassificeerd of een lage uitschieterscore hebben.

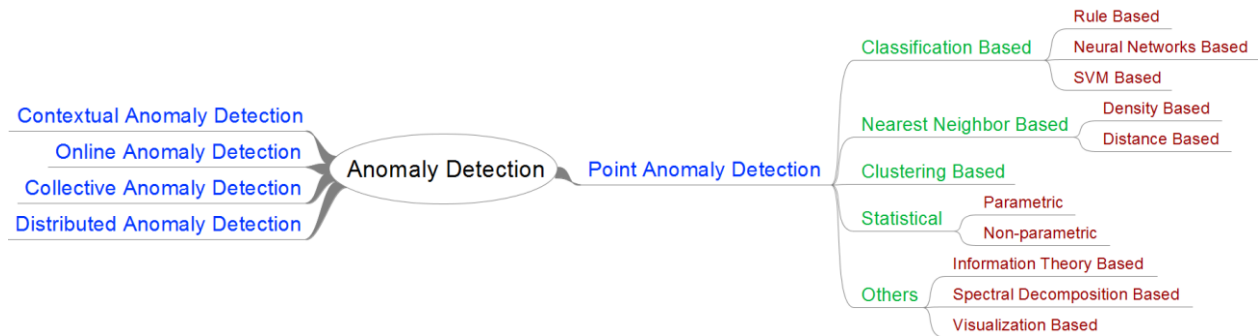
#### 3. Semi-supervised

Training gegevens bevatten gelabelde normale gegevens, maar hebben geen informatie over abnormale objecten.

Er wordt geprobeerd om een label of score te vinden voor het abnormale object met behulp van de informatie uit het label van de normale objecten.

5.2.1.3 Type anomalie

Een belangrijk aspect van een anomaliedetectietechniek is de aard van de gewenste anomalie. Een overzicht op Figuur 13 van verschillende anomaliedetectietechnieken, waarin point anomalie detectie in detail wordt besproken.

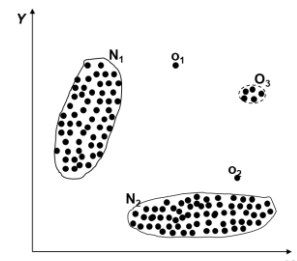


Figuur 13 Overzicht Anomalie Detectie  
Outlier Detection – A Survey [8]

Afwijkingen kunnen worden ingedeeld in drie categorieën:

1. Point Anomalies

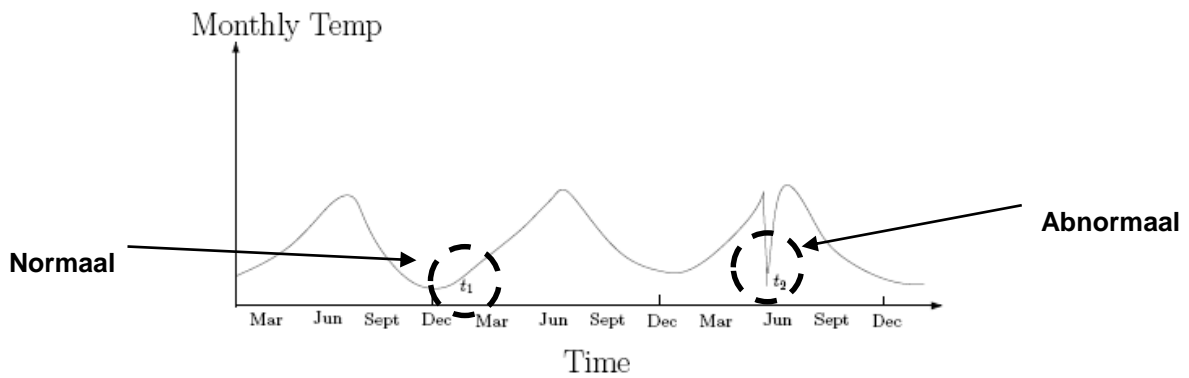
Een individueel instantiegegeven is afwijkend ten opzichte van de gegevens (Figuur 14).



Figuur 14 Point Anomalies

2. Contextual Anomalies

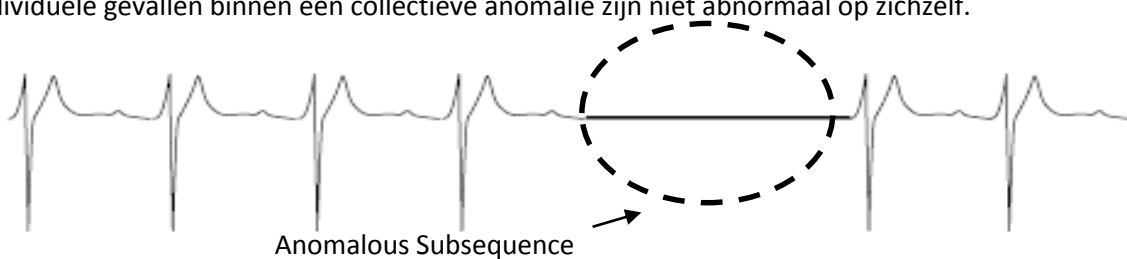
Een individuele gegeven instantie is een anomalie binnen een context. Ook wel aangeduid als voorwaardelijke anomalieën (Figuur 15).



Figuur 15 Contextual Anomalies

3. Collective Anomalies

Een verzameling van gerelateerde instantiegegevens is afwijkend (Figuur 16). Hiervoor is een relatie tussen data-instanties vereist: Sequentiële gegevens, ruimtelijke gegevens en grafiekgegevens. De individuele gevallen binnen een collectieve anomalie zijn niet abnormaal op zichzelf.



Figuur 16 Collective Anomalies

#### 5.2.1.4 Output van anomalie detectie

Een belangrijk aspect voor anomaliedetectietechniek is de wijze waarop de afwijkingen worden gerapporteerd. Typisch, de uitgangen verkregen door anomaliedetectietechnieken zijn van één van de volgende twee typen:

##### 1. Label

Elke testinstantie krijgt een normaal of abnormaal label. Dit geldt vooral bij gebruik van classificatie benaderingen.

##### 2. Score

Elke testinstantie krijgt een anomaliescore toegewezen. Hiermee kan de uitgang worden gerangschikt. Een analist kan ervoor kiezen om ofwel enkele topuitschieters te analyseren of gebruik te maken van een *cutoff threshold* (drempel) om de uitschieters te selecteren.

#### 5.2.1.5 Classificatie gebaseerde technieken

Classificatie wordt gebruikt om een classificatiemodel voor normale en afwijkende (zeldzame) gebeurtenissen te leren, gebaseerd op een set van gelabelde data instanties (*training*). Het is de bedoeling om deze te gebruiken om elke nieuwe ongeziene gebeurtenis te classificeren (*testing*). Classificatie anomaliedetectietechnieken werken via een twee-fasen mode. De trainingsfase leert een *classifier* met behulp van de beschikbare gelabelde trainingsgegevens. De testfase classificeert een test bijvoorbeeld als normaal of abnormaal door gebruik te maken van de *classifier*.

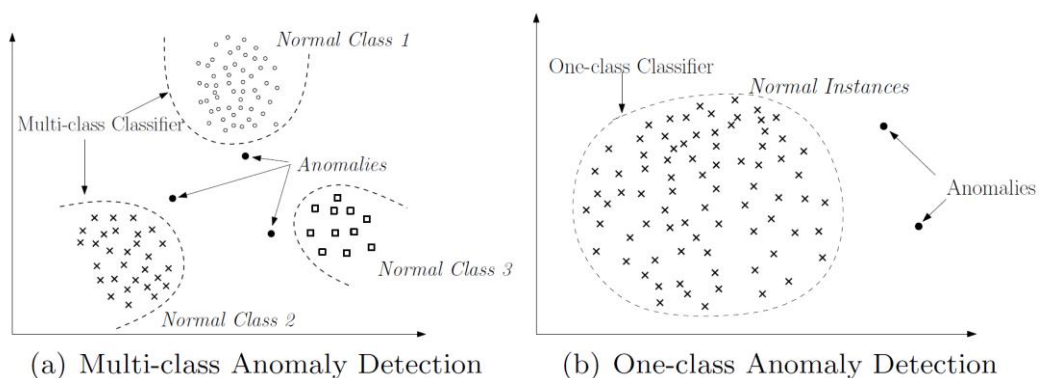
Gebaseerd op de *labels* beschikbaar voor de trainingsfase, kunnen classificatie anomaliedetectietechnieken worden ingedeeld in twee grote categorieën: multi-class en one-class anomaliedetectietechnieken.

##### 1. Multi-class classificatie

Multi-class classificatie anomaliedetectietechnieken veronderstellen dat de training data gelabelde instanties bevat die behoren tot meerdere normale klassen. Dergelijke anomaliedetectietechnieken leren een *classifier* om onderscheid te maken tussen elke normale klas tegen de rest van de klassen. Zie Figuur 17 ter illustratie. Een testinstantie wordt verwacht een anomalie te zijn als het niet geclassificeerd is als normaal door een van de *classifiers*.

##### 2. One-class classificatie

One-class classificatie anomaliedetectietechnieken veronderstellen dat alle trainingsdata slechts één klas label hebben. Zoals getoond in Figuur 17 wordt elke testinstantie dat niet binnen de aangeleerde grens valt verklaart als een anomalie.



Figuur 17 Classificatie anomalie detectie

Voorbeelden van classificatie-gebaseerde technieken zijn Neural Networks Based, Bayesian Networks Based, Support Vector Machines Based en Rule Based. Deze technieken worden uitvoerig besproken in het paper "Outlier Detection - A Survey" (Chandola, V., Banerjee, A. and Kumar, V.) [5].

#### 5.2.1.6 *Nearest Neighbor Based Techniques*

Het concept van de *nearest neighbor analysis* is gebruikt in verscheidene anomaliedetectietechnieken. Dergelijke technieken zijn gebaseerd op de volgende belangrijke veronderstelling: Normale gegevensinstanties hebben naaste burens, terwijl afwijkingen ver van hun naaste burens liggen.

Algemene aanpak in twee stappen: In de eerste stap wordt de nabijheid voor elk datarecord berekend. In de tweede stap wordt de nabijheid geanalyseerd om te bepalen of het datarecord een anomalie is.

Deze techniek kan onderverdeeld worden in twee verschillende categorieën:

##### 1. Proximity-Based Technique

Allereerst wordt de nabijheid tussen de verschillende objecten gedefinieerd. Anomalieën zijn objecten die zich relatief ver van de meeste andere objecten bevinden.

Eenvoudigste manier om de afstand te meten, is de afstand tot de k-dichtste burens (*k-Nearest Neighbors*). De waarde k is een door de gebruiker gedefinieerde constante om een niet gelabeld punt te classificeren. Dit, door het toewijzen van het label dat het meeste voorkomt onder de k training data die het dichtst bij dat punt liggen. Een uitschieterscore wordt gegeven door de afstand tot de k-dichtste burens.

Het is makkelijker om een "goed" object te bepalen dan het bepalen van de statistische verdeling. Maar heeft als nadeel dat het te duur is voor grote datasets en de uitschieterscore zeer gevoelig is aan de waarde van k. Indien k te klein gekozen is zullen enkele nabijgelegen uitschieters een kleine uitschieterscore hebben. Indien k te groot is gekozen zullen clusters met minder dan k objecten aanzien worden als uitschieters.

Deze techniek kan niet gebruikt worden voor datasets met zeer uiteenlopende dichtheden.

##### 2. Density-Based Technique

We maken een schatting van de dichtheid van de objecten. Objecten die in gebieden met een lagere dichtheid liggen, worden aanzien als afwijkingen. Wanneer er een lage dichtheid is, liggen de burens relatief verder van elkaar. Wanneer beschikt wordt over een aantal regio's met sterk uiteenlopende dichtheden, is er nood aan geavanceerdere methodes.

De uitschieterscore van een object is het omgekeerde van de dichtheid rond een object. Een object is een anomalie als het zich in een regio met een lage dichtheid bevindt. Is nauw verwant met *Proximity-Based*. Dichtheid wordt meestal gedefinieerd in termen van *Proximity*. Dichtheid is het omgekeerde van de gemiddelde afstand tot de k-naaste buur (kleine afstand= hoge dichtheid)

#### 5.2.1.7 *Clustering Based Techniques*

Clustering wordt gebruikt om groepen te vinden van (sterk) gerelateerde objecten. Dit wordt gebruikt om objecten te vinden die niet verwant zijn tot andere objecten. Alsook om kleine clusters die ver van andere clusters liggen weg te gooien.

Het voordeel is dat het kan gebruikt worden met elke techniek. Nadelig zijn de vereiste drempels voor minimale grootte en afstand. Ze zijn ook zeer gevoelig aan de gekozen clustergrootte. De techniek kan ook gebruikt worden voor clustering en het beoordelen van de mate waarin een object behoort tot elke cluster.



### 5.2.1.8 *Statistics Based Techniques*

Deze techniek is gebaseerd op het bouwen van een kansverdelingsmodel. Hierop kan overwogen worden hoe waarschijnlijk objecten onder dat model vallen. Een uitschieter is een object dat een lage waarschijnlijkheid heeft met betrekking op een kansverdelingsmodel van de gegevens.

### 5.2.2 *Toepassing*

Op ingestelde tijd (5s) wordt de gelogde data, al dan niet gedeeltelijk, overlopen door de geactiveerde anomaliedetectiesystemen. De monitor vertrouwt op de consensus van de detectiesystemen. Als de meerderheid van de detectiealgoritmen het erover eens zijn dat een bepaalde metriek abnormaal is, zal de metriek worden geclassificeerd als abnormaal en wordt er een notificatie aangemaakt. Via de instellingen van de monitor kan het percentage dat nodig is om een datapunt als abnormaal te classificeren (standaard 50%) aangepast worden.

Momenteel zijn de ontworpen algoritmen gebaseerd op 3-sigma, die afgeleid zijn van Shewhart's statistische procesbeheersing. Hiervoor wordt er gebruik gemaakt van de Numpy Python bibliotheek. Dit is een gratis alternatief voor MATLAB dat ondersteuning biedt voor grote multidimensionale arrays en matrices samen met een indrukwekkende bibliotheek van wiskundige functies.

Later kunnen de algoritmen worden uitgebreid om de monitor zo robuust mogelijk te maken. Hiertoe zijn de algoritmes heel makkelijk uit te breiden en aan te passen. Dit zolang het detectiesysteem een *boolean* teruggeeft.

Hieronder een korte beschrijving van de geïmplementeerde anomaliedetectiesystemen.

#### 5.2.2.1 *Median absolute deviation*

Median Absolute Deviation (MAD) of absolute afwijking rond de mediaan is een robuuste maatregel van een centrummaat. Met een centrummaat wordt een indruk gegeven van het centrum van een hoeveelheid gegevens of een verdeling. Robuuste statistiek zijn statistieken met goede prestaties voor data, afkomstig uit een breed scala van niet-normaal verdeelde kansverdelingen. In tegenstelling tot het standaard gemiddelde en standaardafwijking, is MAD niet gevoelig voor de aanwezigheid van uitschieters. De interkwartielafstand<sup>2</sup> is ook bestand tegen de invloed van uitschieters, hoewel het gemiddelde en MAD beter zijn doordat ze kunnen worden omgezet in waarden die de standaarddeviatie benaderen.

Hoofdzakelijk is het breekpunt voor een parameter (mediaan, gemiddelde, variantie, ...), de verhouding of het aantal willekeurig kleine of grote extreme waarden die moeten worden geïntroduceerd in een sample. Dit om een schatting te veroorzaken die een willekeurig slecht resultaat opleveren. De mediaan breekpunt is 0,5 of de helft (het gemiddelde is 0). Dit betekent dat de mediaan pas "slecht" wordt wanneer meer dan 50% van de waarnemingen uitzonderlijk groot zijn.

Een datapunt is een afwijking wanneer het een ingesteld aantal (standaard 6) keer groter is als de MAD.

#### 5.2.2.2 *Grubbs*

Grubbs' test vernoemd naar Frank E. Grubbs, ook bekend als de maximale genormeerde residuele test, is een statistische test gebruikt om uitschieters te detecteren in een univariante dataset, verondersteld afkomstig uit een normaal verdeelde populatie. Grubbs' test is gebaseerd op de veronderstelling van normaliteit. Wat wil zeggen, dat eerst moet worden gecontroleerd of de gegevens redelijkerwijs kunnen worden benaderd door een normale verdeling voor het aanbrengen van de Grubbs' test.

---

<sup>2</sup> De interkwartielafstand is het verschil tussen de getalswaarde die de laagste 25% van de getalswaarden onderscheidt van de hogere waarde en de getalswaarde die de 25% hoogste getalswaarden onderscheidt.

Grubbs' test detecteert één uitschieter per keer. Deze uitschieter wordt geschrapt uit de dataset en de test wordt herhaald tot er geen uitschieters meer worden gedetecteerd.

De eerste stap is om te kwantificeren hoe ver de uitschieter van de anderen verwijderd is. Hiervoor wordt de verhouding  $Z$  berekend (1) als het verschil tussen de uitschieter en het gemiddelde, gedeeld wordt door de standaardafwijking. Wanneer  $Z$  groot is zal de waarde ver van de anderen liggen.

$$Z = \frac{|Y_i - \bar{Y}|}{s} \quad (1)$$

Bij het analyseren van experimentele gegevens, is de standaardafwijking van de populatie niet gekend. In plaats daarvan wordt de standaardafwijking uit de data berekend. De aanwezigheid van een uitschieter verhoogt de berekende standaardafwijking. Aangezien de aanwezigheid van een uitschieter zowel de teller (verschil tussen de waarde en het gemiddelde) als de noemer (standaardafwijking van alle waarden) verhoogt, kan  $Z$  niet zo groot worden. Wanneer bijvoorbeeld  $N = 3$ , kan  $Z$  niet groter zijn dan 1,155 voor de verzameling waarden. Meer in het algemeen, met een sample van  $N$  observaties, kan  $Z$  nooit groter zijn dan (2):

$$\frac{N - 1}{\sqrt{N}} \quad (2)$$

De Grubbs-test statistiek is gedefinieerd als (3):

Met  $t_{\alpha/(2N), N-2}$  aanduiding van de kritische waarde van de t-verdeling met  $(N-2)$  vrijheidsgraden en een significantieniveau van  $\alpha/(2N)$ .

$$Z > \frac{N - 1}{\sqrt{N}} \sqrt{\frac{t_{\frac{\alpha}{2N}, N-2}^2}{N - 2 + t_{\frac{\alpha}{2N}, N-2}^2}} \quad (3)$$

Een uitschieter wordt gedetecteerd wanneer de  $Z$  Score groter is dan de Grubb's score.

#### 5.2.2.3 Gemiddelde over tijd

Bereken het gemiddelde over een ingestelde tijdsduur (1 uur) van een ingestelde tijd geleden (1 dag). Een tijdreeks is een anomalie als het gemiddelde van de laatste drie datapunten buiten de drie standaarddeviaties van bovenstaande waarde ligt.

#### 5.2.2.4 Standaardafwijking van de gemiddelde

Het voortschrijdend gemiddelde (*moving average*) is het gemiddelde van een vast aantal opeenvolgende elementen in een tijdreeks. Bepaalde periodieke verschijnselen in een tijdreeks kunnen door een geschikte keuze van de periode uitgemiddeld worden, zodat het voortschrijdend gemiddelde het verloop op de langere termijn toont.

Een tijdreeks is een anomalie als de absolute waarde van het gemiddelde van de laatste drie datapunten verminderd met het voortschrijdend gemiddelde (*moving average*), meer is dan het ingestelde (drie) standaarddeviaties van het gemiddelde.

#### 5.2.2.5 Standaardafwijking van het exponentiële voortschrijdend gemiddelde

Een exponentieel gewogen voortschrijdend gemiddelde (EWMA), is een soort oneindige *impuls respons* filters dat weegfactoren toepast die exponentieel afnemen. De weging voor elke oudere datum neemt exponentieel af maar bereikt nooit nul.

Een tijdreeks is een anomalie als de absolute waarde van het gemiddelde van de laatste drie datapunten, verminderd met het exponentiële voortschrijdend gemiddelde, meer is dan drie exponentiële standaarddeviaties van het voortschrijdend gemiddelde. Dit is beter voor het vinden van anomalieën met betrekking tot de korte termijn trends.

#### 5.2.2.6 Gemiddelde verminderd met de cumulatie

Een tijdreeks is een anomalie wanneer de waarde van het volgende gegevenspunt in de reeks, verder is dan drie standaarddeviaties in cumulatieve termen, na aftrekken van het exponentieel gewogen voortschrijdend gemiddelde van elk gegevenspunt.

#### 5.2.2.7 Kleinste-kwadratenmethode

De kleinste-kwadratenmethode (*least squares*) is een rekenmethode om bij een gegeven verzameling puntenparen in het vlak uit een verzameling curven de "best passende" te bepalen. De methode dankt zijn naam "kleinste kwadraten" aan het daarbij gehanteerde criterium voor "best passen", waarbij de mate van passen wordt afgemeten aan het totaal van de kwadratische afwijkingen (meestal in verticale zin) van de curve.

De kleinste-kwadratenmethode in z'n eenvoudigste, oorspronkelijke vorm is een methode om bij een gegeven verzameling punten in het xy-vlak, die verondersteld worden (min of meer) op een rechte lijn te liggen, de "best passende" lijn te bepalen. "Best passend" betekent dat het totaal van de gekwadrateerde afwijkingen in verticale zin van de punten t.o.v. de lijn zo klein mogelijk is.

Een tijdreeks is abnormaal wanneer het gemiddelde van de laatste drie gegevenspunten op een geprojecteerde kleinste kwadratenmodel groter is dan drie keer de standaardafwijking.

#### 5.2.2.8 Histogramklassen

Een tijdreeks is een anomalie als het gemiddelde van de laatste drie datapunten samenvalt in een histogramklas met minder dan de (20) ingestelde andere datapunten. Kleiner in omvang betekent meer afwijkend.

#### 5.2.2.9 Kolmogorov-Smirnovtoets

De Kolmogorov-Smirnovtoets is een statistische toets gebaseerd op een maat voor het verschil in twee verdelingen. In de vorm voor één steekproef, is het een aanpassingstoets, waarmee onderzocht wordt of de verdeling waaruit de steekproef getrokken is, afwijkt van een bekende verdeling zoals de normale verdeling, de uniforme verdeling, de Poisson-verdeling, de exponentiële verdeling, .... In de vorm voor twee steekproeven wordt nagegaan of de verdelingen waaruit de steekproeven afkomstig zijn van elkaar verschillen.

De vorm van twee steekproeven is een zeer geschikte parameter vrije toets om na te gaan of twee steekproeven uit dezelfde verdeling afkomstig zijn, aangezien de toets gevoelig is voor zowel verschillen in plaats als in vorm van de verdelingen. De toetsingsgrootte is de grootste afstand tussen de beide empirische verdelingsfuncties.

Een tijdreeks is een anomalie als in de vorm voor twee steekproeven Kolmogorov-Smirnovtoets aangeeft dat de distributie van gegevens voor de laatste ingestelde (10) minuten iets anders is dan de laatste ingestelde tijd (1uur). Het produceert op een niet-stationaire reeks valse positieven, zo wordt Augmented Dickey-Fuller-test toegepast om te controleren op stationariteit.

### 5.3 Notificatie/Alarm

Wanneer een detectiealgoritme een afwijking in een input detecteert wordt een notificatie aangemaakt. Hierin wordt de informatie zoals de tag-naam, uitschieterwaarde, tijdstempel en het algoritme dat de afwijking heeft gedetecteerd opgeslagen in de databank. Daarna worden notificaties verstuurd naar de gebruikers die voor dit specifiek algoritme één of meerdere meldingen hebben ingesteld.

Er is ondersteuning voor e-mail, SMS via Twilio (in testfase) en HipChat (instant messaging tool voor bedrijfswereld).

Voorbeeld van een notificatie die de gebruiker ontvangt:

*Anomalous value: 550.7328*  
*From M6OPMM01.SNELHEID: grubbs*  
*Time: 2015-04-09 09:35:30*

Bij de opstart van de Monitor worden alle Notificatieklassen opgehaald. Hierdoor kunnen nieuwe notificatie manieren eenvoudig worden toegevoegd door het erven van de Notificatieklasse. Er is verder geen extra configuratie nodig.

## 6 Webapplicatie

Een web front-end die de gebruikers (bv. procesingenieurs) toelaat om de instellingen te beheren (wie welke alarmen krijgt, drempelwaarden e.d.) en een grafische weergave om de gelogde data te bekijken. In dit hoofdstuk worden eerst verschillende raamwerken vergeleken. In deel twee een layout raamwerk en de voordelen hiervan. In deel drie wordt de ondersteuning voor verschillende talen beschreven om zo de webapplicatie toegankelijk te maken voor buitenlandse klanten. Deel vier geeft een vergelijking van verschillende visualisatietools voor het plotten van de gelogde data. Laatste deel geeft een overzicht van de geïmplementeerde functies.

### 6.1 Raamwerk

De mogelijkheid bestaat om een eigen raamwerk te maken, maar zoals Guido van Rossum, de maker van Python, zegt: *"a framework written to serve the needs of a single target application wouldn't necessarily be better than some of the web frameworks that already exist"* ("Een raamwerk geschreven om aan de behoeften van een alleenstaande applicatie te voldoen, is niet noodzakelijk beter dan sommige web-raamwerken die al bestaan").

Door een raamwerk te gebruiken worden veelgebruikte taken geautomatiseerd en handige functies op voorhand gedefinieerd (zoals bv. de toegang tot de databank, template en sessie management, *low-level* handelingen zoals protocollen, sockets en processen). Dit om te vermijden dat code steeds opnieuw moet geschreven worden. Hierdoor kan er sneller gericht worden naar de onderdelen specifiek voor deze masterproef.

Sommige web raamwerken bieden standaard al vele functies aan, zoals een ingebouwde Administrator-interface, terwijl anderen een meer minimale benadering hanteren door alleen de essentie te implementeren om zo een eenvoudige web-toepassing te creëren. Het kiezen van het juiste raamwerk voor de masterproef kan overweldigend zijn door de veelheid aan keuze. Elk raamwerk heeft zijn verschillende voordelen en nadelen. Zoals Titus Brown, een bekende Python ontwikkelaar, zegt: *"There are more Python frameworks than reserved Python keywords."* ("Er zijn meer Python raamwerken dan gereserveerde Python *keywords*." (<https://wiki.python.org/moin/WebFrameworks>))

Web raamwerken kunnen onderverdeeld worden in micro en *full stack* raamwerken. Microramwerken zijn minimalistisch, eenvoudig in te stellen en gemaakt om verschillende *use cases* te behandelen. Full stack raamwerken bevatten meer functies, maar kunnen ingewikkelder zijn om te configureren.

Verschillende Python raamwerken worden met elkaar vergeleken. Waarna het gekozen raamwerk Flask meer in detail wordt besproken. Vanwege het grote aantal beschikbare raamwerken en hun opties, worden enkel de verschillen besproken.

#### 6.1.1 Django

Django is ontworpen voor complexe, databank-gestuurde websites en applicaties. Het is een populaire optie die zichzelf verkoopt als "het web framework voor perfectionisten met deadlines." Django wordt geleverd met ORM (Object relationele *mapper*), een *template engine*, *regex* gebaseerd URL *dispatcher* en beschikt ook over een minimalistische ontwikkel- en testserver. Zo is er geen behoefte aan een externe applicatieserver. Django bevat ook veel gebundelde applicaties zoals een authenticatiesysteem, beheerder-interface, tools voor het genereren van RSS-feeds, .... Django benadrukt herbruikbaarheid van code. Echter kan Django te uitgebreid zijn voor eenvoudige projecten die al deze functies niet vereisen. Het wordt onder andere gebruikt bij The Washington Post, The New York Times, The Onion, Instagram, Pinterest, Mozilla.

De grootste voordelen zijn het sterk benadrukken van herbruikbaarheid van code, het voorzien van vele functies, de grootschaligheid ervan, het fantastische ecosysteem, beschikbare plug-ins en een gebruiksvriendelijke administratorinterface. De nadelen zijn het ingewikkelde bestandssysteem, strikte directory structuur, minderwaardige ORM en enigszins beperkte template taal. ([www.djangoproject.com](http://www.djangoproject.com))

### 6.1.2 Flask

Omschrijft zichzelf als een "*microframework*". Flask is een licht raamwerk die een eenvoudig sjabloon biedt voor webontwikkeling. Door zijn eenvoud, is Flask ideaal voor beginners en mensen met eenvoudige projecten en single-page-toepassingen. Het is echter ook zeer schaalbaar voor grotere projecten. Flask mist een data-abstractielaag, formulier validatie en componenten die veel andere raamwerken bevatten. Toch kunnen deze componenten worden toegevoegd met externe bibliotheken en plug-ins. Het is nu ook vrij goed uitgerust met veel 3<sup>rd</sup> party plug-ins en vooral geschikt voor kleinere websites en eenvoudige webapplicaties tot middelgrote sites.

Het voordeel is de eenvoud, wat tevens ook als nadeel kan zijn doordat er niet veel functies standaard beschikbaar zijn en dus geen "*full stack*" raamwerk is. Wel heeft het een groot gebruiksgemak en flexibiliteit. Van grootte zit het tussen het raamwerk Django en Bottle. ([flask.pocoo.org](http://flask.pocoo.org))

### 6.1.3 Pyramid

Pyramid is een minimalistisch Python web raamwerk geïnspireerd door Zope, Pylons en Django. In termen van standaardfuncties, valt het ergens tussen "*full stack*" raamwerk zoals Django en een microframework als Flask. Pyramid biedt een grote mate van flexibiliteit. Ontwikkelaars kunnen hun eigen templatetaal, formulier bibliotheken en databanklaag kiezen. Wordt gebruikt o.a. door Reddit, Dropbox, BitTorrent, Opera en SourceForge.

Het voordeel is de grotere flexibiliteit dan de microramwerken zoals Flask. Het is minimalistischer dan de andere *full stack* raamwerken. Het heeft wel een kleinere gemeenschap dan Django en klantvriendelijke documentatie ontbreekt. Tevens heeft het een steile leercurve. ([www.pylonsproject.org](http://www.pylonsproject.org))

### 6.1.4 Bottle

Bottle is een *microframework* dat eenvoud benadrukt. Het wordt verspreid als één bestand en heeft geen afhankelijkheden buiten de Python standaardbibliotheek. Het is gemakkelijk te leren en te implementeren en krachtig genoeg om een aanzienlijke website of webapplicatie te bouwen. Door zijn eenvoud is Bottle een uitstekende oplossing voor kleinere projecten. Voor grotere projecten is dit niet aan te raden. Bottle wordt gebruikt door Netflix en tal van kleine websites en applicaties.

Voordelen zijn vooral de snelheid, eenvoud, gebruiksgemak en flexibiliteit. De nadelen zijn de schaalbaarheid en het ontbreken van functies (*microframework*).

Het biedt ondersteuning voor applicaties zoals de Google App Engine en bevat diverse sjablonen zoals Cheetah en Mako. (<http://www.bottlepy.org>)

### 6.1.5 CherryPy

CherryPy is een zeer minimalistisch raamwerk dat de pythonic<sup>3</sup> conventies benadrukt, waardoor ontwikkelaars webapplicaties of websites op dezelfde manier kunnen creëren zoals elk ander object-georiënteerde Python programma. Dit resulteert in compactere broncode en ontwikkelt in minder tijd. Verschillende *full stack* raamwerken, waaronder web2py en TurboGears, gebruiken CherryPy als

---

<sup>3</sup> Python functies benutten om code te produceren die duidelijk, beknopt en onderhoudbaar is.

basistemplate. CherryPy gaat standaard niet om met taken zoals templates of toegangsrechten. In plaats daarvan kan het raamwerk uitgebreid worden met plug-ins. Hierdoor is CherryPy nog eenvoudiger dan *microframeworks*, zoals Flask. De server is ook apart te gebruiken en te combineren met andere raamwerken zoals Bottle, Flask en Django. Het wordt o.a. gebruikt door Hulu, Netflix, SickBeard, TurboGears en web2py.

De voordelen zijn vooral de eenvoud, flexibiliteit, vele beschikbare plug-ins en de onafhankelijkheid van andere pakketten, maar heeft gebrek aan functies door zijn minimalistische aanpak. ([www.cherrypy.org](http://www.cherrypy.org))

#### 6.1.6 web.py

Dit minimalistisch raamwerk beweert het meest "*pythonic*" te zijn met een templatetaal die streeft om Python dichterbij HTML te brengen, zodat het lijkt dat er rechtstreeks met HTTP wordt gepraat. In plaats van andere manieren te bedenken om HTML te beschrijven. De meeste raamwerken worden geleverd met allerlei functies en complexiteiten maar web.py houdt het graag simpel en efficiënt. ("*Django lets you write web apps in Django. TurboGears lets you write web apps in TurboGears. Web.py lets you write web apps in Python.*" Adam Atlas) ([www.webpy.org](http://www.webpy.org))

#### 6.1.7 Tornado

Tornado is een Python web framework en een asynchrone netwerk bibliotheek, oorspronkelijk ontwikkeld door FriendFeed. Dankzij het gebruik van *non-blocking* netwerk I/O, kan Tornado uitbreiden naar tienduizenden open verbindingen. Het is ideaal voor langdurige *polling*, *websockets* en andere toepassingen die een langlevende verbinding naar elke gebruiker vereisen. In principe is het de huidige concurrent van Node.js in Python. De API is niet fantastisch, maar niet te moeilijk en de prestaties zijn goed. ([www.tornadoweb.org](http://www.tornadoweb.org))

#### 6.1.8 TurboGears

TurboGears begint als een *microframework* en kan uitbreiden tot een *full stack* oplossing. Dit is selectief tot stand gebracht, rekening houdend met de beperkingen van Django en Rails raamwerk. TurboGears heeft een zeer krachtige en flexibele ORM (*Object Relational Mapper*), die multi-databank ondersteunt. In het moderne tijdperk van Web applicaties is AJAX een must en TurboGears wordt geleverd met een *widget* systeem om AJAX-gebaseerde applicaties gemakkelijker te maken. Het heeft een gebruiksvriendelijk template engine en ondersteunt ook meerdere data-uitwisselingsformaten. ([www.turbogears.org](http://www.turbogears.org))

#### 6.1.9 Web2py

Web2py is een *full stack* raamwerk geschikt voor snelle ontwikkeling van "veilige" (beschermen tegen XSS-aanval, Schadelijke File Execution en nog veel meer aanvallen), schaalbare en snelle databank-gestuurde webapplicaties. Werkt met de meeste populaire SQL databanken zoals MySQL, PostgreSQL, MSSQL, SQLite, Oracle, .... Het benadrukt backward compatibiliteit en werkt met een aantal 3<sup>de</sup> party tools, waarbij 3<sup>de</sup> party afhankelijkheden worden vermeden. Er zijn geen configuratie-bestanden, er is geen installatie nodig en het kan worden uitgevoerd vanaf een USB-stick. Van grootte zit het tussen Django en Bottle, maar biedt een andere filosofie en bevat veel grafische tools. ([www.web2py.com](http://www.web2py.com))

#### 6.1.10 Bobo

Bobo is een minimalistisch raamwerk voor het creëren van WSGI (Web Server Gateway Interface) webapplicaties. Heeft als doel om gemakkelijk te gebruiken en te onderhouden. Bobo heeft geen templatetaal, databank-integratielaag of andere functies die beter door WSGI *middleware* of applicatie-specifieke bibliotheken worden uitgevoerd. Bobo bouwt voort op andere raamwerken, met name WSGI en WebOb. ([bobo.digicool.com](http://bobo.digicool.com))

### 6.1.11 Cyclone

Cyclone is een webserverraamwerk voor Python dat de Tornado API implementeert als een Twisted protocol (zie 6.1.15). Deze combinatie is de basis voor het opbouwen van hybrideservers die in staat zijn om HTTP zeer efficiënt te behandelen en tegelijkertijd e-mail, ssh, sip, irc, ... aan te bieden of te gebruiken. Het geeft gelijkaardige prestaties als Twisted maar is eenvoudiger om conventionele webapplicaties te schrijven. Als nadeel heeft het een relatief kleine gemeenschap. ([www.cyclone.io](http://www.cyclone.io))

### 6.1.12 Itty-Bitty

Itty.py is een klein experiment, beïnvloed door het Sinatra micro-raamwerk geschreven in Ruby, dat precies genoeg doet om bruikbaar te zijn en niets meer. Niet aan te raden in een Enterprise omgeving. ([www.github.com/toastdriven/itty](http://www.github.com/toastdriven/itty))

### 6.1.13 Klein

Klein is een *microframework* voor de ontwikkeling van web services met Python. Het is gebouwd op de veel gebruikte en goed geteste componenten zoals Werkzeug en Twisted en heeft bijna volledige testdekking. ([www.github.com/twisted/klein](http://www.github.com/twisted/klein))

### 6.1.14 Morepath

Morepath is een Python WSGI microramwerk en zeer krachtig. Het gebruikt routing, maar de routering wordt behandeld door de modellen. Morepath is model-gedreven en flexibel. ([morepath.readthedocs.org](http://morepath.readthedocs.org))

### 6.1.15 Twisted

Twisted is een *event*-gedreven raamwerk geschreven in Python, wat betekent dat gebruikers korte *callbacks* schrijven die worden opgeroepen door het raamwerk. Twisted ondersteunt een groot aantal protocollen. Zeer krachtig raamwerk, ongelooflijke goede prestaties maar een slechte API. ([www.twistedmatrix.com](http://www.twistedmatrix.com))

### 6.1.16 Pecan

Pecan werd opgericht om het gat in de Python web-raamwerken te vullen. Is een zeer minimalistisch raamwerk dat *object-dispatch* routing aanbiedt. Pecan heeft niet tot doel om een "*full stack*" raamwerk te zijn en omvat dus geen *out of the box* ondersteuning voor dingen zoals sessies of databanken. Pecan plaats richt zich op HTTP zelf. ([www.pecanpy.org](http://www.pecanpy.org))

### 6.1.17 Conclusie

De keuze is gereduceerd tot Django en Flask, deze beschikken beiden over een zeer uitgebreide documentatie, grote gemeenschap, goede prestatie en veel plug-ins die steeds bijgewerkt worden. Beide kunnen worden gebruikt voor kleine tot grote projecten. Doordat Flask een micro-web raamwerk is en standaard over minder functies beschikt, is het eenvoudiger voor kleinere projecten. Hierdoor wordt Flask boven Django verkozen.

Flask is gebaseerd op de Werkzeug *toolkit*, een *Web Server Gateway Interface* modules en Jinja2 template engine. Het valt onder de BSD licentie en het heeft tot doel de kern eenvoudige maar uitbreidbaar te houden. Flask mist een data-abstractielaag, formulervalidatie en andere componenten die veel andere raamwerken bevatten. Toch kunnen deze componenten worden toegevoegd met externe bibliotheken en plug-ins. Zoals bv. er voor formulervalidatie gebruik gemaakt is van de bibliotheek WTForms. Dit heeft als voordeel dat er veel meer zelf bepaald kan worden en dit in tegenstelling tot het Django raamwerk.



## 6.2 Lay-out raamwerk

Voor de lay-out van de webapplicatie is er gebruik gemaakt van Bootstrap versie 3. Het bevat HTML en CSS-gebaseerde design templates voor typografie<sup>4</sup>, vormen, knoppen, grafieken, navigatie en andere interface-onderdelen. Bootstrap werkt nauw samen met jQuery (een vrij JavaScript-framework voor dynamische en interactieve pagina's). Er zijn nog tal van extra plug-ins gratis beschikbaar. Eén van de zaken die dit framework zo modern maakt ten opzichte van de rest, is de mogelijkheid tot een responsief ontwerp.

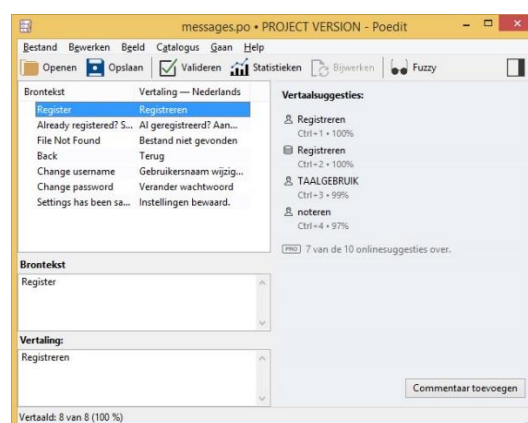
Door gebruik te maken van Jinja2 macro's (de template engine van Flask) kan op eenvoudige manieren volledige formulieren gegenereerd worden geschikt voor Bootstrap.

## 6.3 I18n en L10n

De term I18n staat voor *internationalization*, 18 staat voor het aantal letters tussen de eerste i en de laatste n in het woord. L10n staat voor *localization*, door de lengte van het woord. Om de webapplicatie toegankelijk te maken voor buitenlandse klanten worden verschillende talen ondersteunt. Momenteel is er ondersteuning van Engels en het Nederlands in testfase. Om met vertalingen te werken wordt er gebruik gemaakt van Babel. Samen met de uitbreiding Flask-Babel die I18n en L10n ondersteuning toevoegt aan Flask.

Door gebruik te maken van pybabel, onderdeel van de Babel uitbreiding, worden de gemarkeerde teksten voor vertaling geëxtraheerd in een apart "\*.pot"-bestand. Het "\*.pot"-bestand is een template-bestand dat alle teksten bevat die vertaald worden. Dit bestand wordt gebruikt als een model om taalbestanden te genereren.

Om een vertaling van een nieuwe taal te maken, wordt er terug gebruik gemaakt van pybabel. Deze zal een "\*.po"-bestand aanmaken in een standaardformaat voor vertalingen. Dit is een formaat dat o.a. gebruikt wordt door de *gettext* tool ([www.gnu.org/software/gettext](http://www.gnu.org/software/gettext)). Er zijn vele vertaalapplicaties die werken met "\*.po"-bestanden. Er is gekozen voor Poedit omdat het een van de meest populaire is en ondersteuning biedt voor alle belangrijke besturingssystemen. Hieronder een schermafbeelding (Figuur 18) waarin enkele teksten zijn vertaald naar het Nederlands. Het linkse deel van het figuur toont de teksten in hun originele en vertaalde taal. Rechts worden enkele vertaalsuggesties gegeven om zo snel mogelijk en efficiënter te kunnen vertalen.



Figuur 18 Poedit

<sup>4</sup> Typografie is de kunst van het vormgeven, zetten en drukken van tekst, zowel voor functionele als esthetische doeleinden.

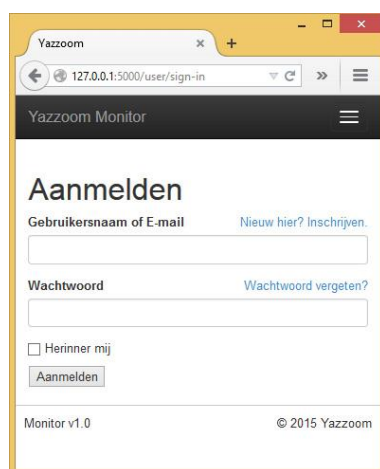
Zodra de teksten vertaald zijn en terug naar het “\*.po”-bestand zijn opgeslagen, wordt via pybabel de (leesbare) inhoud gecompileerd naar een “\*.mo” bestand. Dit bestand bevat de vertaalde teksten in een geoptimaliseerde vorm die efficiënt in de applicatie kan gebruikt worden.

Voor al de voorgaande stappen zijn Python scripts geschreven zodat in de toekomst de vertalingen eenvoudig kunnen worden bijgewerkt.

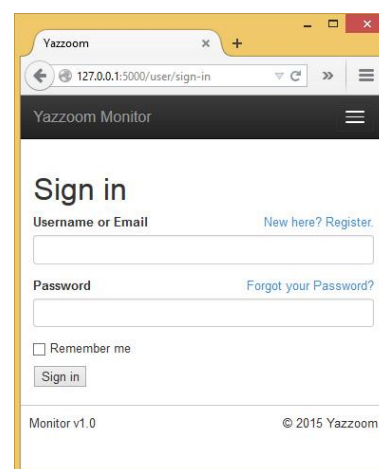
Aan de hand van de *Accept-Languages header* verzonden door de browser in het HTTP-verzoek wordt de best passende taal uit de lijst van ondersteunende talen bepaald.

De header is in de meeste browsers standaard geconfigureerd met de gekozen taal van het besturingssysteem. Alle browsers geven gebruikers de kans om andere talen te selecteren. Gebruikers kunnen een lijst met talen, elk met een gewicht voorzien. Hierbij een voorbeeld:

*Accept-Language: nl, en-gb; q = 0,8, en; q = 0,7*



*Figuur 20 Webapplicatie Nederlands*



*Figuur 19 Webapplicatie Engels*

## 6.4 Plot

Om de gelogde data grafisch weer te geven, wordt een visualisatietool gebruikt. Voor Python zijn er veel opties beschikbaar voor het visualiseren van de gegevens. Hierdoor is het een uitdaging om te achterhalen welke wanneer te gebruiken. Onderstaand worden de meeste voorkomende bibliotheken, die geschikt zijn, kort besproken. [9]

### 6.4.1 Matplotlib

Matplotlib is de grootvader van de python visualisatiepakketten. Het is zeer krachtig, maar hierdoor ook complex. Door de ruime mogelijkheden die matplotlib biedt, is het niet eenvoudig om ermee te werken. Veel tools (bv Seaborn) zijn *wrappers* rond matplotlib. Een nadeel is dat het veel werk vraagt om complexe grafieken overzichtelijk weer te geven. <http://matplotlib.org>

### 6.4.2 Seaborn

Seaborn is een visualisatiebibliotheek op basis van matplotlib en heeft als doel om standaard datavisualisaties veel aanschouwelijker te maken en om een ingewikkelde plot eenvoudiger te creëren. Het integreert goed met de Pandasbibliotheek. <http://stanford.edu/~mwaskom/software/seaborn>

### 6.4.3 Ggplot

Ggplot is vergelijkbaar met Seaborn omdat dit voortgebouwd is op matplotlib en heeft tot doel de visualisaties van matplotlib op een eenvoudige manier te verbeteren. Het wijkt van Seaborn af doordat

het een geport is van ggplot2 voor de programmeertaal R. Gezien dit doel zijn sommige API niet-pythonic, maar wel zeer krachtig. <http://ggplot2.org>

#### 6.4.4 MPLd3

De mpld3 project brengt matplotlib en D3.js (de populaire “low-level” Javascript bibliotheek voor het maken van interactieve visualisaties voor het web) bij elkaar. Het resultaat is een eenvoudige API voor het exporteren van matplotlibgrafieken naar interactieve grafieken in de browser. <http://mpld3.github.io>

#### 6.4.5 Bokeh

Bokeh verschilt van de voorafgaande bibliotheken omdat het niet afhankelijk is van matplotlib en gericht is op het genereren van visualisaties in moderne webbrowsers. Het doel is om stijlvolle, eenvoudige nieuwe grafieken te genereren in de stijl van D3.js, maar ook de mogelijkheid te leveren voor hoge performante interactiviteit over zeer grote of *streaming* datasets. Het is bedoeld om interactieve webvisualisaties te maken. <http://bokeh.pydata.org>

#### 6.4.6 Pygal

Pygal wordt gebruikt voor het maken van dynamische SVG-grafieken. De SVG-bestanden zijn vrij nuttig om interactieve grafieken eenvoudig te maken. Het is vrij gemakkelijk om visuele aantrekkelijke grafieken te maken. <http://pygal.org>

#### 6.4.7 Plotly

Plotly is anders doordat het een online tool is voor analyse en visualisatie. Het beschikt over een python API en uitstekende documentatie, hierdoor is het creëren van grafieken relatief eenvoudig. Plotly integreert naadloos met Pandas. <https://plot.ly>

#### 6.4.8 Rickshaw

Rickshaw is een JavaScript-toolkit voor het creëren van interactieve tijdreeksgrafieken. Het is gebaseerd op D3, zodat grafieken worden getekend met standaard SVG en gestyled met CSS. Het is gratis en open source, beschikbaar onder de MIT-licentie en ontwikkeld bij Shutterstock. <http://code.shutterstock.com/rickshaw>

#### 6.4.9 Bearcart

Bearcart is een kleine bibliotheek voor het creëren van Rickshaw visualisaties met Pandastijdreeksen. Het heeft een eenvoudige API, een aantal plotsoorten, overzichtelijke legendes en tooltips. Bearcart gebruikt Jinja2 templates om de output te creëren en ze zijn eenvoudig aan te passen voor de toepassing. <https://github.com/wrobstory/bearcart>

#### 6.4.10 Vincent

Trifacta heeft het mogelijk gemaakt om visualisaties te bouwen op D3 met Vega. Vega is een visualisatie modelleertaal, in een declaratief formaat voor het maken, opslaan en delen van visualisatie ontwerpen. Vincent maakt het gemakkelijk om Vega te gebruiken met Python. Vincent is compatibel met Pandas en is speciaal gebouwd om te zorgen voor het snel plotten van Pandas DataFrames en Series. <http://vincent.readthedocs.org>

#### 6.4.11 Mcflyin

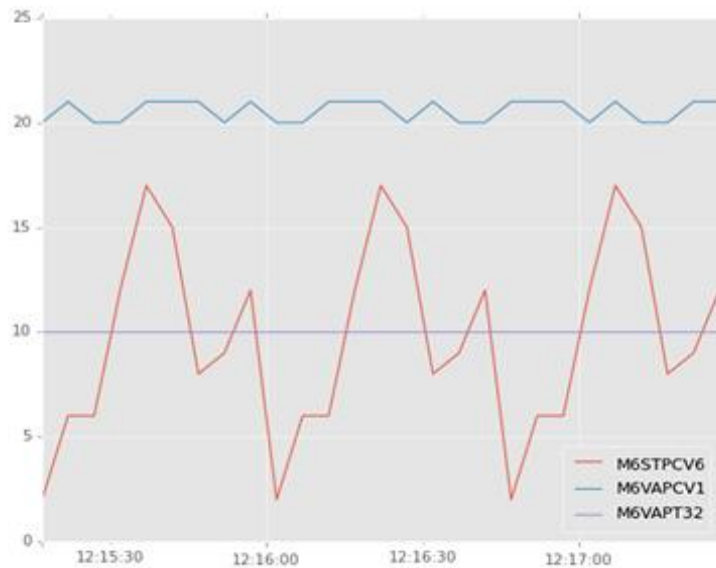
Past transformaties toe op tijdreeksen en heeft gebruik gemaakt van Pandas, het raamwerk Flask, Vincent of Bearcart. <https://github.com/wrobstory/mcflyin>

#### 6.4.12 Python NVD3

NVD3 heeft als doel om herbruikbare grafieken en grafiekcomponenten voor D3.js te creëren, zonder het wegnemen van de flexibiliteit die D3.js biedt. Python NVD3 is een wrapper voor NVD3. <https://github.com/areski/python-nvd3>

#### 6.4.13 Conclusie

Elk van bovenstaande bibliotheken heeft zijn voor- en nadelen. Pandas heeft een zeer goede integratie met Matplotlib om te plotten. Door tijdsgebrek is hiervoor ook gekozen. In een later stadium kan er gekozen worden voor interactieve grafieken. Zoals bv Bearcart die zeer geschikt lijkt voor het plotten van tijdseries, samen met Mcflyin om de data te converteren.



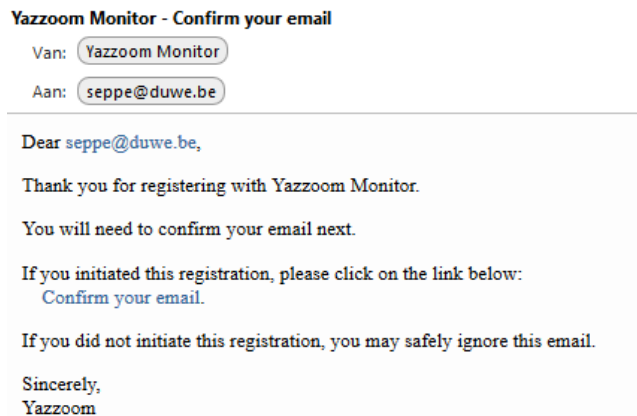
Figuur 21 Voorbeeld Matplotlib

## 6.5 Functies

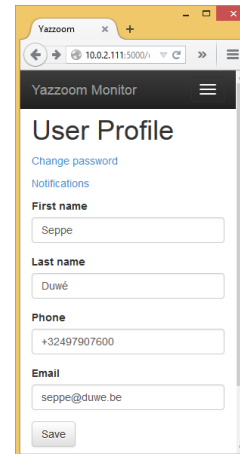
Onderstaand de verschillende functies die aanwezig zijn in de webapplicatie. Afbeeldingen kunnen afwijken van nieuwere versies.

### 6.5.1 Gebruikers profiel

Mogelijkheid om te registreren (al dan niet verifiëren via email, zie Figuur 23), profiel wijzigen (zie Figuur 22) en wachtwoord resetten.



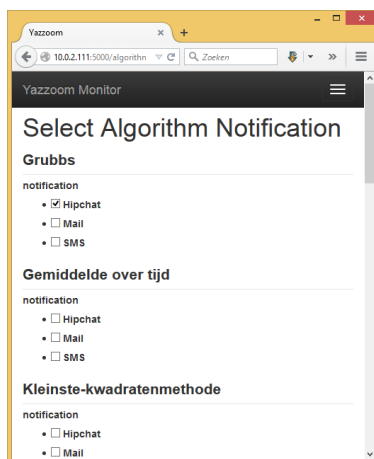
Figuur 23 Bevestig e-mail



Figuur 22 Profile

### 6.5.2 Notificaties

Elke gebruiker kan per anomaliedetectie algorithm instellen welke notificaties er worden verstuurd, zie Figuur 26. Momenteel is er ondersteuning voor Mail (Figuur 24), SMS en Hipchat (Figuur 25).



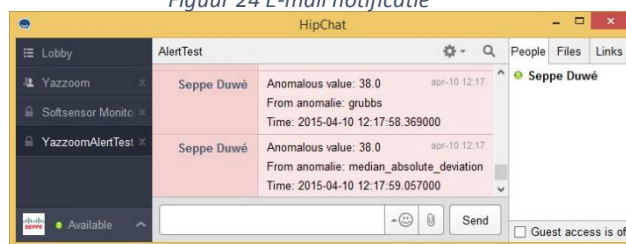
Figuur 26 Algorithm notificatie

#### Yazzoom Toolbox Alert: stddev\_from\_average

Van: info yazzoom  
Aan: admin@duwe.be

Anomalous value: 1.0  
From M6VDV08.TM: stddev\_from\_average  
Time: 2015-03-20 10:12:57.110000

Figuur 24 E-mail notificatie



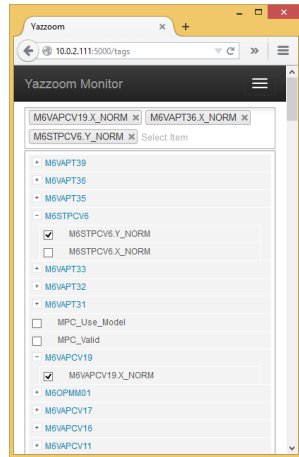
Figuur 25 Hipchat notificatie

### 6.5.3 Tags

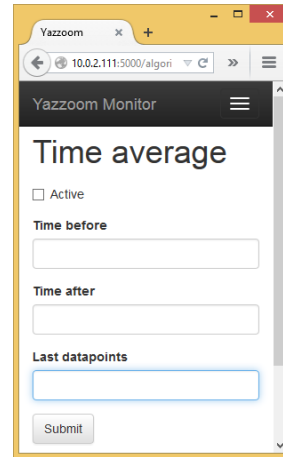
De gebruikers die in de groep "Admins" zitten, krijgen een extra item in het menu. Zij krijgen de bevoegdheden om de tagnamen te selecteren, afkomstig van de OPC-server, die gelogd en geanalyseerd worden (zie Figuur 28). Per anomaliedetectie algoritme kunnen specifieke instellingen worden aangepast (zie Figuur 27).

Om de beschikbare tagnamen op te halen wordt er gebruik gemaakt van AJAX. In plaats van alle tagnamen in één keer op te halen, wat de OPC-server mogelijks kan belasten, worden de gegevens van 1 niveau diep opgevraagd. Door gebruik te maken van het "."-karakter worden verschillende branches gecreëerd, om zo een logische structuur op te bouwen die het beheer ervan vereenvoudigt.

Asynchronous JavaScript And XML (AJAX) is een term voor het ontwerp van interactieve webpagina's waarin asynchroon gevraagde gegevens worden opgehaald van de webserver. Daardoor hoeven dergelijke pagina's niet in hun geheel vernieuwd te worden. Voor het transport van gegevens wordt in dit geval JSON gebruikt. JSON staat voor JavaScript Object Notation en is een deelverzameling van de programmeertaal JavaScript.



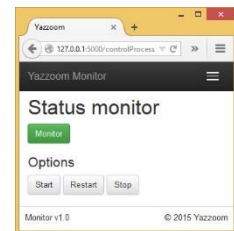
Figuur 28 Select tags



Figuur 27 Settings algorithm

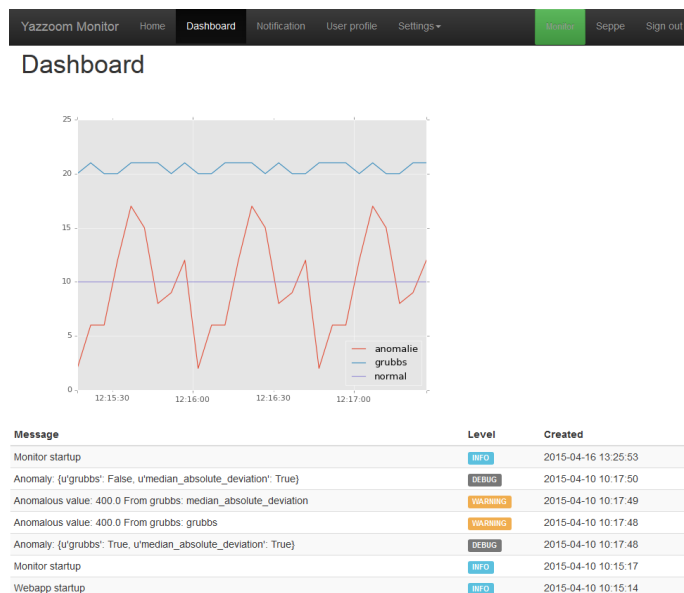
### 6.5.4 Status monitor

De webapplicatie en de monitortool zijn gescheiden applicaties. Hierdoor is het mogelijk om enkel de monitor, die de tagswaarden logt en bij afwijkingen notificaties verstuurd, uit te voeren zonder de webapplicatie. De status van de monitortool wordt rechts bovenaan in het menu weergegeven (Figuur 30). De kleuren groen of rood geven aan dat het proces in uitvoering is of niet. De monitor kan via de webapplicatie gestart of gestopt worden (Figuur 29).



Figuur 29 Statusmonitor

### 6.5.5 Dashboard



Figuur 30 Dashboard

Op het dashboard wordt de gelogde data in grafiek, samen met de loginformatie van de applicatie weergegeven (zie Figuur 30). De loginformatie bevat verschillende niveaus (debug, info, warning, error, critical) en dit elk aangeduid met een kleur al naargelang de ernst.

## 7 Databank

Voor het opslaan van de gebruikersprofielen, instellingen en gelogde data wordt de SQLite-databank gebruikt. Dit is een databanksysteem dat gebruik maakt van Structured Query Language (SQL) en geen nood heeft aan een aparte server. Vooreerst wordt het begrip ORM en de gekozen ORM (SQLAlchemy) uitgelegd. In het tweede deel wordt een overzicht van de databankstructuur weergegeven, waarna elke tabel wordt verduidelijkt samen met een voorbeeld.

### 7.1 ORM

Object-Relational *Mapping*, gewoonlijk aangeduid als ORM, is een techniek die objecten van de applicaties koppelt aan tabellen in een relationeel databankbeheersysteem. De bewerkingen uitgevoerd op de objecten worden vertaald in de databankcommando's, transparant door de ORM. Met behulp van ORM, kunnen de eigenschappen en relaties van de objecten in een toepassing gemakkelijk worden opgeslagen en opgehaald uit een databank zonder het rechtstreeks schrijven te van SQL-statements.

SQLAlchemy biedt een volledig pakket van bekende enterprise-level persistentie patronen, ontworpen voor een efficiënte en performante toegang tot de databank.

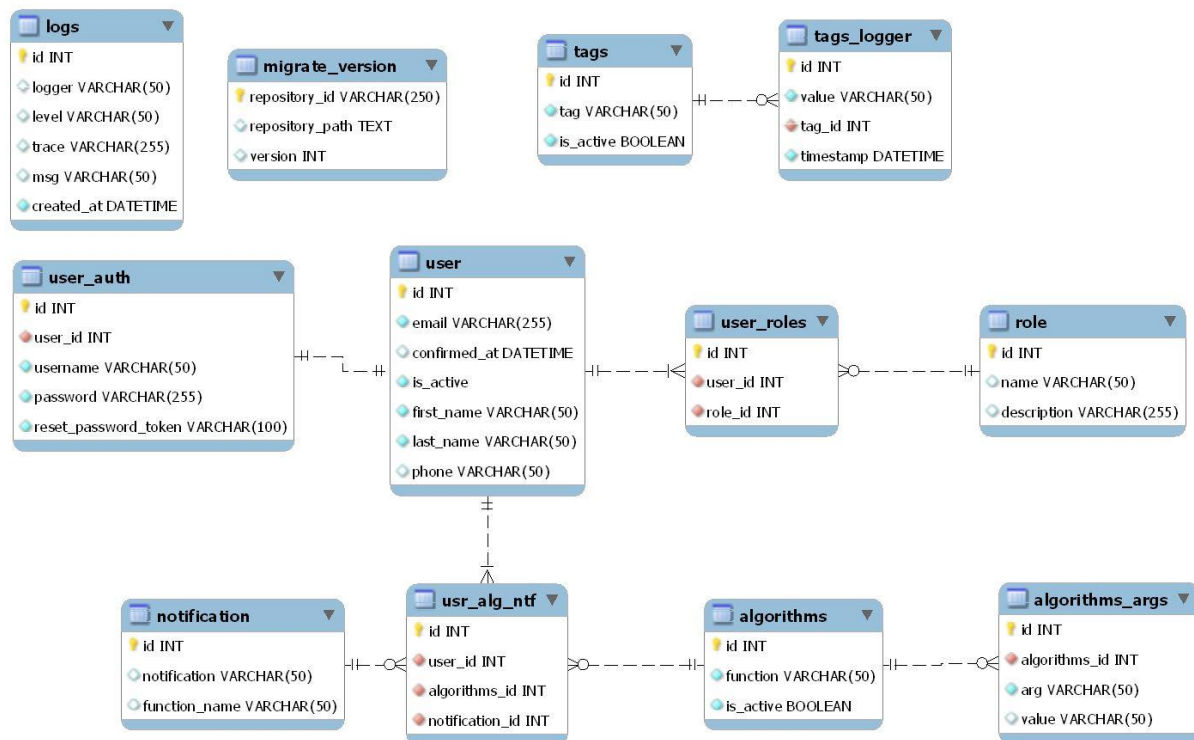
SQL-databanken gedragen zich minder als objectcollecties en daarom worden grootte en prestaties belangrijker. Objectcollecties gedragen zich minder als tabellen en rijen, daardoor is abstractie belangrijker. SQLAlchemy streeft ernaar om aan beide principes tegemoet te komen.

SQLAlchemy bestaat uit twee afzonderlijke componenten, bekend als de Core en de ORM. De Core zelf is een volledig functionele SQL-abstractietool, die een laag van abstractie voorziet over diverse DBAPI-implementaties evenals een SQL *Expression Language* welke expressies van de SQL-taal toelaat via generatieve Pythonuitdrukkingen.

SQLAlchemy is het meest bekend om zijn ORM, een optioneel onderdeel dat het data *mapper* patroon voorziet (in plaats van het actieve recordpatroon dat door een aantal andere object-relatieve *mappers* wordt gebruikt). Dit laat toe om elk Python-type te converteren naar een databanktype.

Buiten SQLite is er ook ondersteuning voor tal van andere databanken zoals Firebird, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, Sybase die standaard worden meegeleverd met de Core en dit kan nog verder uitgebreid worden door externe projecten.

## 7.2 Structuur



Figuur 31 Structuur databank

### 7.2.1 Migrate version

Wanneer er structurele wijzigingen aangebracht worden op de databank kan SQLAlchemy-migration gebruikt worden voor het bijhouden van de databank updates. In plaats van de oude databank te verwijderen en een nieuwe te maken waarbij alle gegevens verloren gaan. Wanneer de gegevens moeilijk opnieuw kunnen worden aangemaakt, moet er noodgedwongen export en importscripits geschreven worden. Dit is nu geautomatiseerd, dus geen zorgen meer over handmatige databank-migraties.

Repository_id	Repository_path	Version
1	C:\db_repository	2

### 7.2.2 Tags

In de tagkolom worden de tagnamen, afkomstig van de OPC-server, opgeslagen. Via de “active” boolean (een datatype met 2 mogelijke waarden) wordt bepaald of de tag gelogd en geanalyseerd wordt op afwijkingen. De beheerders kunnen deze waarden wijzigen via de webapplicatie.

id	tag	active
1	B_CONCENTR_6.B	1
2	M6BZMM01.Reserve6	0
3	M6BZMM01.SNELHEID	1

### 7.2.3 Tags logger

De waarden van de geactiveerde tags zullen op ingestelde tijd worden gelogd. Om te kijken hoe de applicatie zou werken na lange tijd werden een maximaal realistisch (3000000 samples per tag \* 100 tags) aantal records toegevoegd. Hieruit bleek dat informatie opvragen van een bepaalde tag meer als 3 minuten in beslag nam. Dit was onaanvaardbaar, maar door het toevoegen van een index op kolom Tag\_id is dit gereduceerd naar ongeveer 400ms, wat een aanvaardbare wachttijd is.



id	value	Tag_id	timestamp
1	20.0	1	2015-04-10 12:15:17
2	2.0	2	2015-04-10 12:15:17

### 7.2.4 User auth

Er wordt gebruik gemaakt van de Flask-Login plug-in voor het afhandelen van het inloggen, uitloggen en het onthouden van de gebruiker sessie. Hiervoor was het aangeraden om een aparte tabel te creëren. De username van de gebruiker waarmee wordt aangemeld alsook het gehasht wachtwoord worden bijgehouden. Wachtwoorden worden gehasht met bcrypt. Als het wachtwoord moet worden gereset, wordt de reset\_password\_token kolom gebruikt.

id	User_id	username	Password	Reset_password_token
1	1	seppe	\$2a\$12\$EoFORiWOvEMeWDFdvosVt.tlqOU7NFT8fwJTS4nvzJXrHE65kT0W	
2	2	yazzoom	\$2a\$12\$EoFORiWOvEMeWDFdvosVt.tlqOU7NFT8fwJTS4nvzJXrHE65kT0W	

### 7.2.5 Users

Alvorens de gebruiker de gelogde data samen met de te ontvangen notificaties kan instellen moet er geregistreerd worden. Deze gegevens worden opgeslagen in de tabel "Users". Er wordt onderscheid gemaakt tussen de voornaam, achternaam, telefoonnummer (voor sms). Een boolean "is\_active" geeft aan dat het profiel geactiveerd is

id	email	Confirmed_at	Is_active	First_name	Last_name	phone
1	seppe@duwe.be	2015-05-03 16:53:07	1	Seppe	Duwe	+32497907600
2	info@yazzoom.com	2015-05-03 16:55:07	1	Info	Yazzoom	+32486316809

### 7.2.6 User roles

Bepaalt de relatie tussen de gebruikers en de verschillende rollen. Hierdoor is het mogelijk om een gebruiker aan meerdere rollen te linken.

id	User_id	Role_id
1	1	1
2	2	1

### 7.2.7 Role

Bij Role worden de verschillende soorten groepen van gebruikers opgeslagen, elk met bepaalde rechten. Nu zijn er 2 groepen: Admins en Users.

id	name	discription
1	Admins	Administrator
2	Users	Registered Users

### 7.2.8 Notification

Bij het starten van de Monitor worden de verschillende geïmplementeerde notificatieklassen opgehaald en de nodige informatie opgeslagen in de databank. Hierdoor is er geen extra configuratie nodig om nieuwe notificaties toe te voegen. Erven van de Notificatie-klasse is voldoende.

id	Notification	Function_name
1	Mail	NotificationMail
2	SMS	NotificationSMS
3	Hipchat	NotificationHipchat

### 7.2.9 Usr al ntf

De applicatie heeft nood aan 3 meer-naar-meer relaties. De verschillende notificaties, de geactiveerde algoritmen om anomalieën te detecteren en de gebruikers.

id	User_id	Algorithms_id	Notifaction_id
1	1	1	1
2	1	3	1
3	1	3	3

### 7.2.10 Algorithms

Alsook wordt hier bij de eerste opstart van de Monitor gekeken welke detectie algoritmen geïmplementeerd zijn en deze worden opgeslagen in de databank.

id	Function	Is_active
1	Grubbs	1
2	mean_subtraction_cumulation	1
3	histogram_bins	0

### 7.2.11 Algorithm args

Bij verschillende algoritmen zijn er extra configuratie instellingen. Om zo de detectie te optimaliseren aan de specifieke data.

id	Algorithms_id	Arg	Value
1	1	significance_level	5
2	3	last_datapoints	3
3	3	bin_max	20

### 7.2.12 Logs

Registreert de opgetreden gebeurtenissen, vb. het opstarten van het programma, detectie van een uitschieter, een onverwacht resultaat, ... samen met de belangrijkheidsgraad.

id	Logger	Level	Trace	Msg	Created_at
1	App.app_and_db	INFO		Webapp startup	2015-04-10 09:40:17
2	Toolbox.applicatie	INFO		Monitor startup	2015-04-10 09:40:18
3	Toolbox.applicatie	WARNING		Anomalous value: 38.0 From anomalie: grubbs	2015-04-10 10:12:59

## Besluit

De belangrijkste elementen bij deze masterproef zijn het filteren van anomalieën (afwijkingen) binnen een productieproces en het genereren van de hierop gepaste meldingen zodoende een geoptimaliseerd productieproces te bekomen.

De grote uitdaging was, het uitlezen van de ingangswaarden afkomstig van de productiesensoren en deze constant vergelijken met de vooraf gestelde normale en verwachte waarden voor het bekomen van een feilloos productieproces.

Bij mijn onderzoek ben ik gestart met het uitlezen en verwerken van datasensoren van een bepaald productieproces (meer bepaald papier, pulp) gedurende een bepaalde tijd (24 uur). Indien er uitzonderlijke sensorwaarden aanwezig zijn, dienen deze te worden bepaald en zal het soft-sensor model onjuiste uitgangswaarden genereren.

Mijn masterproef is een gebruiksvriendelijke toepassing die geïmplementeerd is in de programmeertaal Python. Deze laat gebruikers toe om aan de hand van verschillende instellingen een verwittiging te sturen (SMS, e-mail, ...) en een melding te geven wanneer één van de ingangen voor de soft-sensor afwijken. Hierdoor kan er tijdig worden ingegrepen, aangepast of bijgestuurd al dan niet machinaal of via menselijke handelingen. Het resultaat is dat een geoptimaliseerd productieproces verkregen wordt die de anomalieën reduceert, beter nog uitsluit.

De geleverde prestatie zal als basis dienen voor de Yasense Monitor tool.

De stageopdracht met daaraan gekoppeld de masterproef heeft me zaken bijgebracht, zoals het belang van de keuze van een juiste competitieve stage. Het onderzoeken, het formuleren van voorstellen, het uitwerken van ideeën en vragen van mij over vooraf totaal onbekende problemen.

Verdere uitwerking van de masterproef is het uitbreiden van de anomaliedetectiesystemen om de monitor zo robuust mogelijk te maken. Het anomaliedetectiesysteem kan verder uitgebreid worden naar een systeem die de kans voorspelt en waardoor het aantal mogelijke variëteiten vergroot. De beschikbare info op het dashboard van de webapplicatie kan nog uitgebreid worden, alsook de grafieken van de gelogde data dynamisch genereren. Hierdoor zou het mogelijk worden om te zoomen of de weergave aan te passen. Tevens meer instellingen via de webapplicatie beschikbaar stellen, zodanig dat er minder op de server dient ingesteld te worden.

Bij toekomstige mogelijkheden en uitbreiding kijk ik uit naar een gebruiksvriendelijk, betaalbaar software programma dat bij tal van productieprocessen kan worden ingezet met een 100% feilloos resultaat als streefdoel.

Dit wordt de uitdaging naar de toekomst toe. Moeilijk, ja - doenbaar, ja - maar vooral mogelijk!

Bijlagen



**MEASURE THE UNMEASURABLE**

Soft sensors, also known as surrogate sensors or virtual sensors, are computer models that calculate estimates of physical sensor measurements or lab measurements, based on other variables that can be more easily and/or more frequently and reliably measured.

Soft sensors are a valuable solution for the following issues with classical measurement instruments:

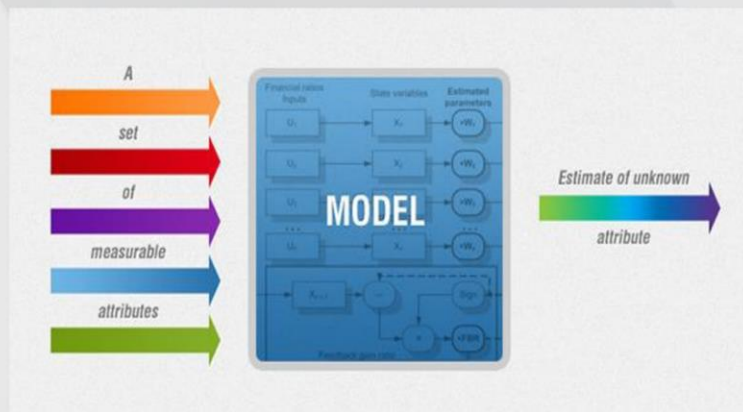
- Lack of appropriate, reliable, affordable on-line instrumentation.
- Laboratory measurements are often done infrequently and with long analysis delays, using expensive equipment and with a high operational cost.
- Physical sensors and process analyzers are sometimes put off-line for maintenance service, and fail from time to time.

**To MEASURE is to KNOW**

# YASENSE

## Soft Sensor Tool Suite

A family of software tools to develop, integrate and monitor soft sensors



### Benefits of using YASENSE

- Improves your **plant's profitability**
- Faster availability of product and/or process quality information allows you to **react faster**
- Better control and as a result **savings in energy and material cost and increased safety**
- Better quality control and as a result **more consistent production**
- Realize new **innovating applications**
- Produce **less off-quality product**

## Bibliografie

- [1] „Soft sensor development,” Yazzoom, [Online]. Available: <http://en.apc.yazzoom.com/soft-sensor-development.html>. [Geopend 05 11 2014].
- [2] L. Fortuna, S. Graziani, A. Rizzo en M. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*, Springer, 2007.
- [3] A. Martelli, A. Ravenscroft en D. Ascher, *Python cookbook*, " O'Reilly Media, Inc.", 2005.
- [4] OPC, „What is OPC?,” OPC Foundation, [Online]. Available: [www.opcfoundation.org/about/what-is-opc/](http://www.opcfoundation.org/about/what-is-opc/). [Geopend 13 Feb 2015].
- [5] „About OpenOPC,” OpenOPC, [Online]. Available: <http://openopc.sourceforge.net/>. [Geopend 5 Mei 2015].
- [6] D. Graham, E. Van Veenendaal, I. Evans en R. Black, *Foundations of software testing: ISTQB certification*, Course Technology Cengage Learning, 2008.
- [7] M. Hammond en A. Robinson, *Python Programming on Win32: Help for Windows Programmers*, " O'Reilly Media, Inc.", 2000.
- [8] V. Chandola, A. Banerjee en V. Kumar, „Anomaly detection: A survey,” *ACM Computing Surveys (CSUR)*, vol. 41, nr. 3, p. 15, 2007.
- [9] I. Milovanovi, *Python Data Visualization Cookbook*, Packt Publishing Ltd, 2013.
- [10] A. Verikas, „Soft Sensors for Monitoring,” in *Soft Sensors for Monitoring*, 2009.
- [11] M. Serpas, Y. Chu en J. Hahn, „Fault detection approach for systems involving soft sensors,” *Journal of Loss Prevention in the Process Industries*, vol. 26, nr. 3, pp. 443-452, 2013.
- [12] M. Lutz, *Learning python*, " O'Reilly Media, Inc.", 2013.
- [13] K. D. Lee, *Python programming fundamentals*, Springer, 2010.
- [14] S. Kabadayi, A. Pridgen en C. Julien, „Virtual sensors: Abstracting data from physical sensors,” in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, 2006.
- [15] H. J. Galicia, Q. P. He en J. Wang, „Adaptive outlier detection and classification for online soft sensor update,” in *International Symposium on Advanced Control of Chemical Processes ADCHEM*, 2012.
- [16] T. Dunning en E. Friedman, *Practical Machine Learning: A New Look at Anomaly Detection*, " O'Reilly Media, Inc.", 2014.
- [17] J. Verhasselt, „Soft sensor development and application,” in *Soft sensor development and application*, Yazzoom, 2014.